



Doc2.0 Connecting Party Integration

Jan 05, 2026

CONTENTS

1	Payment Gateway overview	2
2	Connecting Party integration scenario	3
3	Connecting Party integration example	5
4	Payment Gateway Transaction Types	6
5	API Use-Cases	8
5.1	Server-to-Server Sale	8
5.2	Server-to-Server Sale by SBP	13
5.3	Server-to-Server Preauth, Capture and Cancel	13
5.4	Server-to-Server Transfer	21
5.5	Server-to-Server Payout	26
5.6	Server-to-Server Account Verification	28
5.7	Server-to-Server Account Verification by SBP	34
5.8	Sale Form	34
5.9	SBP Sale Form	36
5.10	Preauth Form, Capture and Cancel	36
5.11	Transfer Form	42
5.12	Payout Form	43
5.13	Payout By Reference	50
5.14	Payout with bank confirmation	52
5.15	Account Verification Form	52
5.16	SBP Account Verification Form	54
5.17	Withdrawal by SBP	57
5.18	Deposit To Card Transfer	57
5.19	Recurring Sale	59
5.20	Recurring Preauth, Capture and Cancel	64
5.21	Recurring Payments Service	70
5.22	Return Transactions	76
5.23	Payment Cashier	78
5.24	Google Pay™ App	87
5.25	Google Pay™ Web	96
5.26	Apple Pay Web	104
5.27	eCheck Sale	115
5.28	Buy Now Button	115
5.29	Direct Bank transfer	120
5.30	Bank transfer with Invoice	122
5.31	Mobile Device Sale	125

5.32	Mobile Device Transfer	129
5.33	Mobile Device Card Verification	134
6	General API Usage	136
6.1	Async-Response	136
6.2	Doc2.0 test scenarios	137
6.3	Request Authentication Methods	140
7	API Commands, Callbacks & Redirects	171
7.1	/api/v2/account-verification	171
7.2	/api/v2/account-verification-form	182
7.3	/api/v2/auth-cash	191
7.4	/api/v2/documents/add-document	199
7.5	/api/v2/bank-wire-transfer	203
7.6	/api/v2/capture	208
7.7	/api/v2/create-card-ref	212
7.8	/api/v2/close-day	215
7.9	/api/v2/get-balance	220
7.10	/api/v2/get-balance/manager	223
7.11	/api/v2/get-balance/merchant	226
7.12	/api/v2/get-exchange	229
7.13	/api/v2/get-card-info	232
7.14	/api/v2/make-rebill-preauth	236
7.15	/api/v2/make-rebill-sale	245
7.16	/api/v2/pan-eligibility	255
7.17	/api/v2/pan-eligibility/form	266
7.18	/api/v2/preauth	273
7.19	/api/v2/preauth-form	288
7.20	/api/v2/return	302
7.21	/api/v2/sale	306
7.22	/api/v2/sale-form	325
7.23	/api/v2/payout-form	339
7.24	/api/v2/sale-echeck	349
7.25	/api/v2/status	356
7.26	/api/v2/void	364
7.27	/api/v2/card-insurance-document	368
7.28	/api/v4/sale	370
7.29	/api/v4/create-card-ref	391
7.30	/api/v4/create-recurring-payment	395
7.31	/api/v4/process-recurring-payment	404
7.32	/api/v4/update-recurring-payment	407
7.33	/api/v4/create-recurring-payments	415
7.34	/api/v4/process-recurring-payments	424
7.35	/api/v4/update-recurring-payments	427
7.36	/api/v4/status	435
7.37	/api/v4/payout	444
7.38	/api/v4/payout-form	456
7.39	/api/v4/payout-by-ref/	468
7.40	/api/v4/payout-check	479
7.41	/api/v4/payout-pay	491
7.42	/api/v4/withdrawal-sbp	495
7.43	/api/v4/transfer	502
7.44	/api/v4/transfer-form	539
7.45	/api/v4/transfer-by-ref	589

7.46	/api/mfo/scoring	602
7.47	/api/mfo/v2/scoring-form	609
7.48	/api/mfo/v2/scoring-result	611
7.49	/api/scoring/v1/score	616
7.50	/api/scoring/v1/status	619
7.51	/mapi/v1/sale	621
7.52	/mapi/v1/preauth	629
7.53	/registry/upload	636
7.54	/services/transaction-report	637
7.55	/services/v2/rpt-engine	655
7.56	/view-invoice	658
7.57	/view-invoice-xml	660
7.58	/api/3ds/v1/upload-cres-result	662
7.59	/api/3ds/v1/upload-method-url-result	665
7.60	/api/3ds/v1/upload-pares-result	669
7.61	CRes Redirect	673
7.62	PaRes Redirect	674
7.63	Final Redirect	676
7.64	Mobile Device Connecting Party Server API	678
7.65	Mobile Device Doc2.0 API	722
7.66	Connecting Party Callbacks	765
8	mPOS Integration	773
8.1	mPOS API	773
8.2	mPOS SDK for iOS	774
8.3	mPOS SDK for Android and Java	780
8.4	mPOS FAQ	786
8.5	mPOS Test	793
8.6	mPOS API Ru	794
8.7	mPOS Terminals	817
8.8	mPOS Test	838
9	Common Utilities	840
9.1	Add Document	840
9.2	Account balance	841
9.3	Get Exchange Rate	843
9.4	PAN Eligibility	844
9.5	Receipt and message templates	849
9.6	Reports for Integration	855
9.7	Fiscalization service	857
9.8	Card Scoring	860
9.9	Card Scoring by NSPK	862
9.10	Data migration	863
9.11	Insurance	864
10	Reference	871
10.1	FAQ	871
10.2	3DS Overview	875
10.3	Forms Customization	896
10.4	Country and State Codes	911
10.5	Currency Codes	1023
10.6	Payment Gateway Internal Errors	1026
10.7	Language Codes	1030
10.8	Order Stage	1034

10.9 Statuses 1050

10.10Transaction Filters 1051

10.11Glossary 1148

- [Payment Gateway overview](#)
- [Connecting Party integration scenario](#)
- [Connecting Party integration example](#)
- [Payment Gateway Transaction Types](#)

PAYMENT GATEWAY OVERVIEW

“Doc2.0 Payment Gateway” (here and hereafter System, Payment Gateway or Doc2.0) is a PCI DSS certified platform, which provides accepting, processing, storage and transmitting of payment data between participants of payment processes.

Main participants:

Payers or Receivers	end customers of merchants.
Connecting Party	merchants themselves, PSP/Payment institutions which represent merchants, or third-party systems for data exchange (CRM, BI, monitoring).
Processors	integrated external payment institutions and payment providers.

Payment Gateway provides the following methods of accepting payment data:

1. API available on the Internet
2. POS-terminals
3. Virtual terminal for manual entry of payment data, received by e-mail and phone

Payment Gateway provides access to user accounts. It has the following user roles. Each root entity, can have its own Employees who can get access to the data from the root entity, but with certain restrictions:

Merchant	provided to the merchant’s representatives (Connecting Party).
Reseller	provided to the agent, which engage merchants for Payment Gateway.
Manager	provided to the representatives of Payment Gateway.

Note: See all terms definitions in [Glossary](#).

CONNECTING PARTY INTEGRATION SCENARIO

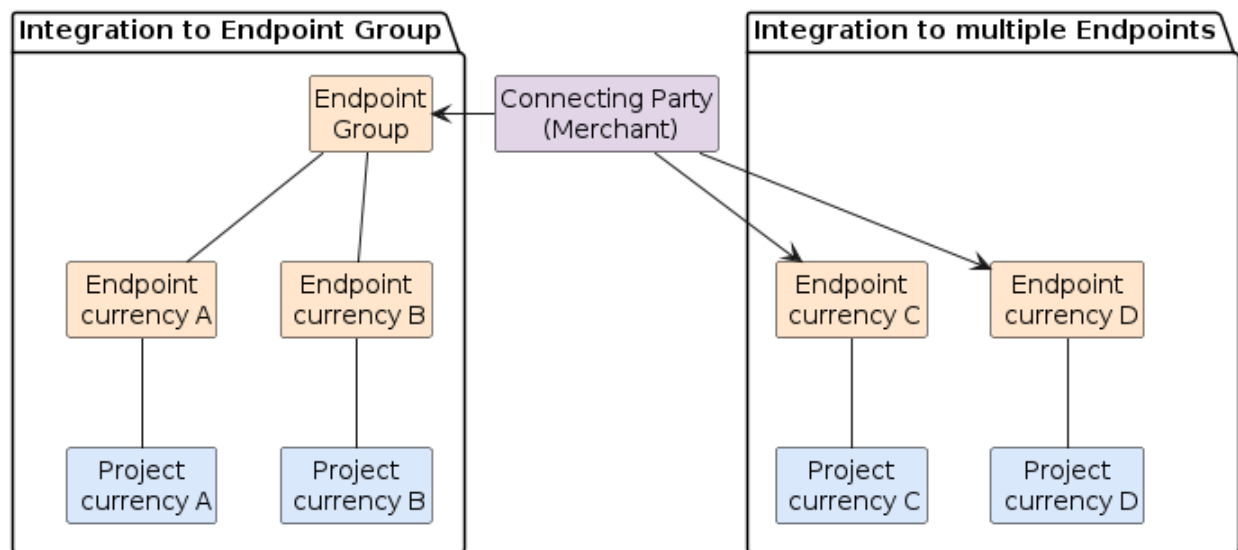
Depending on the PCI compliance and business requirements, Connecting party integrates to Payment Gateway via Server-to-Server APIs, Hosted payment form APIs or a combination of them. Each integration option is provided in [API Use-Cases](#) section, and each API Use-Case provides clear instructions which [API commands](#) to call on each stage of the payment flow and how to handle their results. All APIs are [asynchronous](#). [Common utilities](#) section provides added value services which might be enabled by request. [FAQ](#) page helps to resolve common issues during Connecting Party integration.

Doc2.0 support manager configures Projects with connected Endpoints and Endpoint Groups (if needed) for one or multiple merchant accounts of Connecting Party in Payment Gateway. For API integration, Connecting party receives the following data from Doc2.0. This data can be provided independently for sandbox (test) and production environment. Any additional required credentials are mentioned in relevant API Use-Cases.

Main required credentials are:

1. Endpoint IDs per currency or Endpoint Group IDs for multicurrency integration (see reference schema below).
2. Merchant login.
3. Merchant control key.
4. Integration scenario documentation.

Options for multi-currency processing integration



CONNECTING PARTY INTEGRATION EXAMPLE

1. [Sale Form](#) - integrate hosted payment form processing of E-commerce sale transactions.
2. [Return transactions](#) - integrate processing of refunds.
3. [Connecting Party callbacks](#) - receive transaction data to CRM, BI and other systems.
4. [Forms customization](#) - brand hosted payment forms and provide them to Doc2.0 support manager for installation.
5. Test the solution on sandbox with Doc2.0 [test scenarios](#).
6. Inform Doc2.0 support manager about successful finish of testing and request production credentials to start processing payments.

PAYMENT GATEWAY TRANSACTION TYPES

Transaction is the operation of money transfer between accounts. Doc2.0 payment solution supports all common types of transactions associated with bank card and alternative payments. Transactions can be initiated via API-commands mentioned in corresponding Use-Cases, Virtual terminal, and other methods.

Payments:

sale - Sale is a type of transaction, in which Payer receives goods or services from Connecting Party in exchange for money or other assets. Sale combines the pre-authorization and capture process in one transaction (final authorization). Credit card associations require that Connecting Party submits a sale transaction request only when the order is fulfilled immediately. For example, when selling an item over the counter in a retail store.

preauth - Preauthorization (also preauth) is a transaction type in which bank blocks the specified amount in the Payer's card account and does not allow the cardholder to use this blocked money. The block remains for a definite period of time. The Preauth transaction is requested when the Payer makes a purchase. Successful Preauth confirms the cardholder's ability to pay, ensuring that the Payer's credit card account is in good standing with sufficient funds to complete the purchase with capture request later.

cancel - Unlocks the funds blocked by Preauth transaction.

capture - After providing a service/product to the Payer, Connecting Party uses the existing pre-authorization and submits a capture request to initiate a transfer of previously held funds between the Payer's credit card account and Connecting Party checking account.

reversal - Returns the specified amount to the cardholder's account.

void - A credit card purchase that a seller cancels after it has been authorized but before it has been settled.

transfer - MasterCard MoneySend, Visa Money Transfer, and Deposit-to-Card transactions.

payout - Payout is a type of transaction which results in funds transfer from Connecting Party banking account to customer (receiver) banking account or digital wallet. Payout transaction in most cases is used for bank account funding.

Dispute and Service transactions:

fraud - Marks fraudulent transaction.

retrieval - The card issuer asks the merchant for a copy of the actual ticket of a transaction.

chargeback - A chargeback occurs when a cardholder contacts their credit card issuing bank to initiate a refund for a purchase made on their credit card.

chargeback_reversal - When applicable, the acquirer may process a second presentment for a chargebacked transaction.

prearbitration - The issuer may initiate an arbitration chargeback after the second presentment (2nd chargeback).

arbitration - If the acquirer does not accept financial responsibility for the Prearbitration transaction, he may pursue Arbitration (2nd chargeback reversal).

account_verification - Zero Dollar Value Authorization Request (CVV check).

mfo_scoring - Additional risk scoring to mitigate fraud operations.

pan_eligibility - Verification request to ensure that a cardholder's account is eligible to receive MasterCard MoneySend transactions. Provides additional information for sending and receiving cards: the Interbank Card Association (ICA) code of the issuer, currency code (alpha and numeric), country code (alpha and numeric), brand acceptance code, and brand product code.

API USE-CASES

5.1 Server-to-Server Sale

- [Introduction](#)
- [Sale Flow](#)
- [3DS Decision Making Schema](#)
 - [Non3D Flow](#)
 - [Simplified Authentication Flow](#)

5.1.1 Introduction

Sale is a type of transaction, in which Payer receives goods or services from Connecting Party in exchange for money or other assets. In this use case Server-to-server Sale, card data is transferred directly in initiating request.

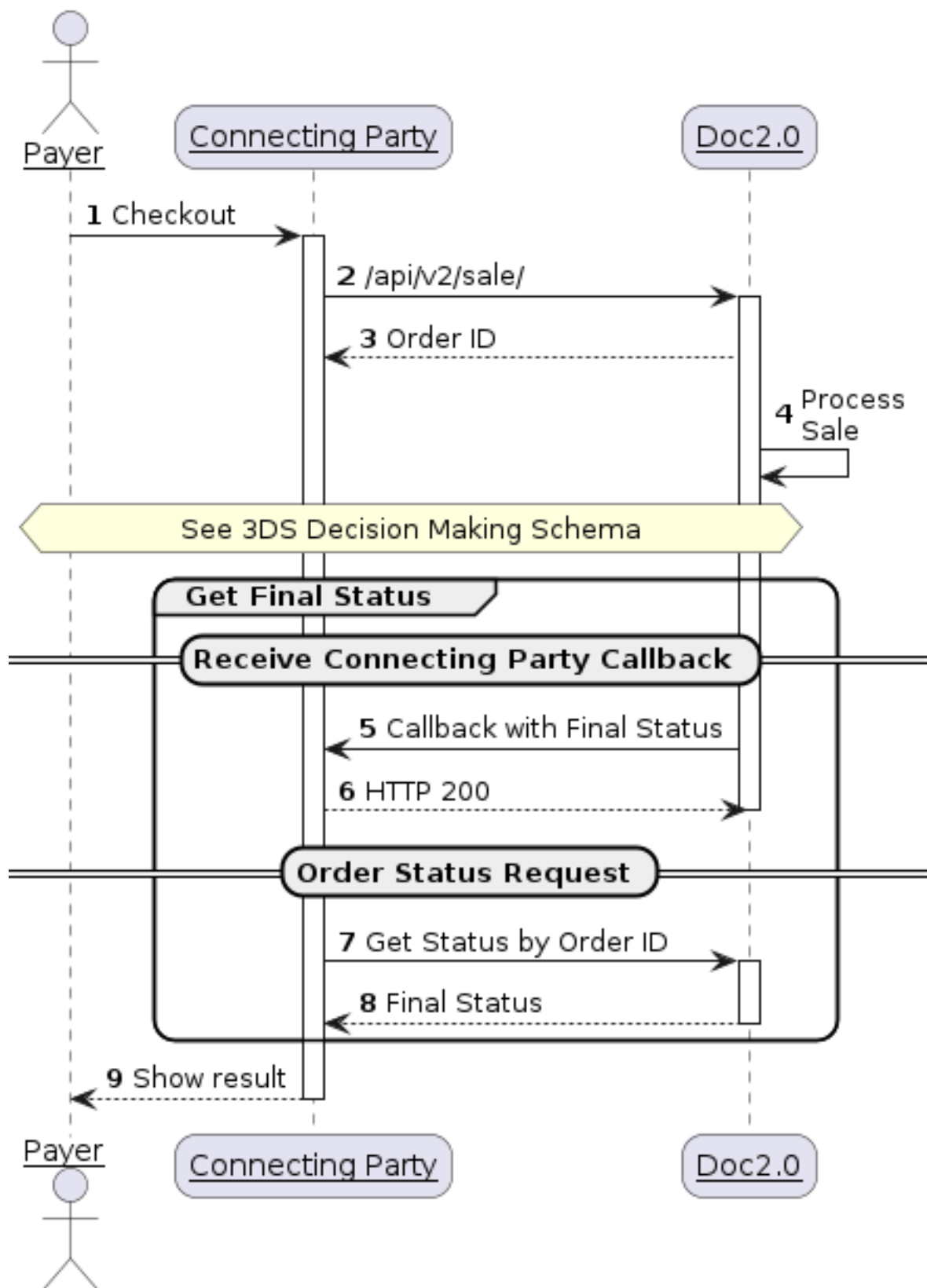
See terms definitions (Connecting Party, 3DS Method, etc) in [Glossary](#).

5.1.2 Sale Flow

(2) To implement sale request see </api/v2/sale/>. See [3DS Overview](#) to get more information about 3DS flow. See [3DS Decision Making Schema](#) and [3DS Implementation Scenarios](#) to correctly implement 3DS flow for this Use-Case.

(5) To implement callback with final status handling see [Connecting Party Callback](#).

(7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.



5.1.3 3DS Decision Making Schema

Connecting party has to implement all steps marked in green and purple. Below are the description for steps which reference specific API commands according to the step ID:

- (1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (4) If html and redirect-to fields are present, see [Simplified authentication flow with html page](#).
- (5) The same as point (1).

Note: The [3DS decision making schema](#) is showcasing 3DS being initiated and performed by Payment Gateway. 3DS is initiated by Payment Gateway and performed on Connecting Party side., see [3DS Overview](#)

Non3D Flow

Sale transaction should be considered as non3D (no 3DS authentication) if all conditions are met:

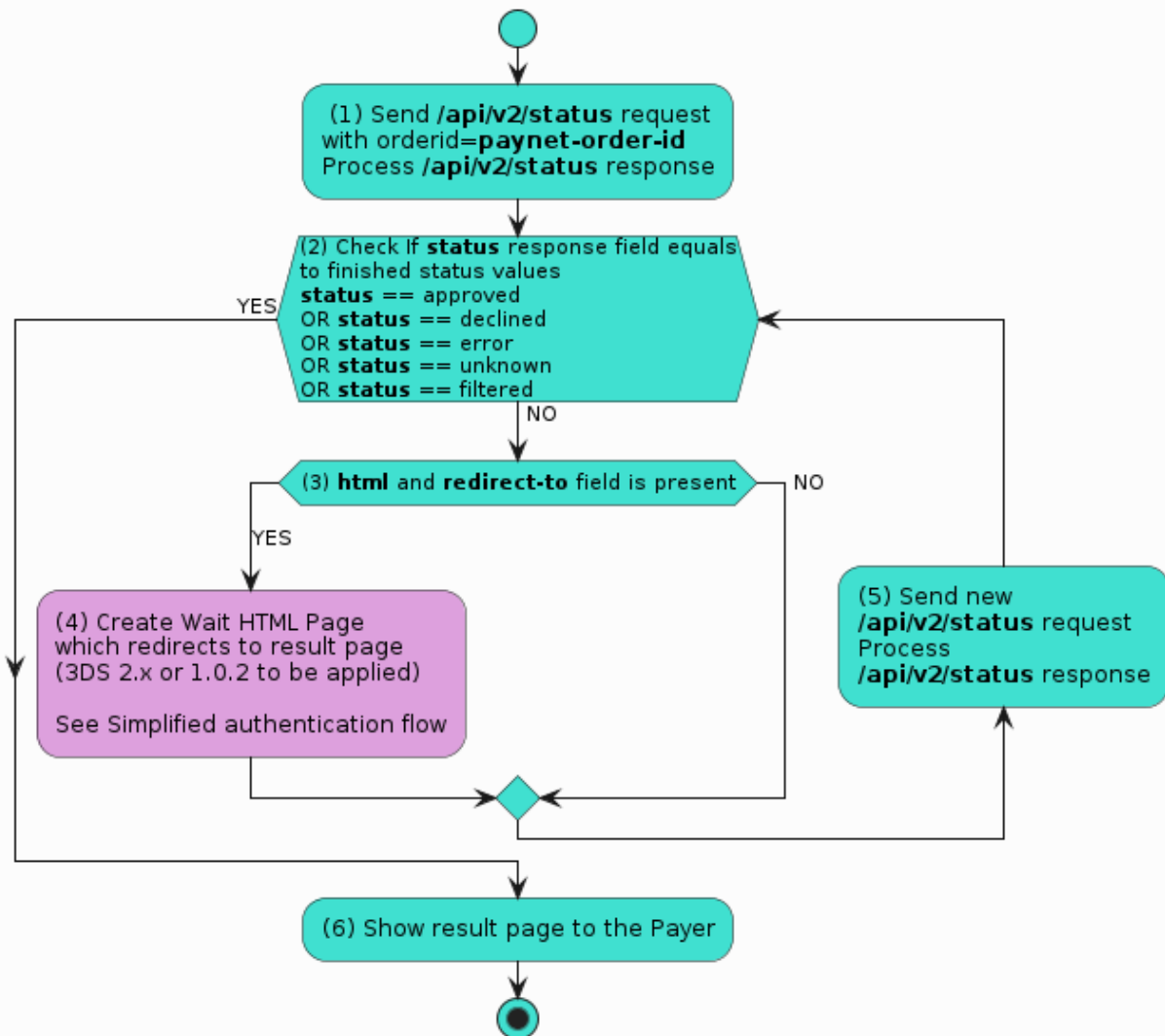
- 1. Steps 1-2-(5)-6 of [3DS decision making schema](#) were followed.
- 2. tds_status, html and redirect-to parameters were not present.
- 3. Transaction received final status (approved, declined, error, filtered).

Note: Please note that transaction status “unknown” might appear for both 3DS and non3D transactions. See details in [Statuses](#).

Simplified Authentication Flow

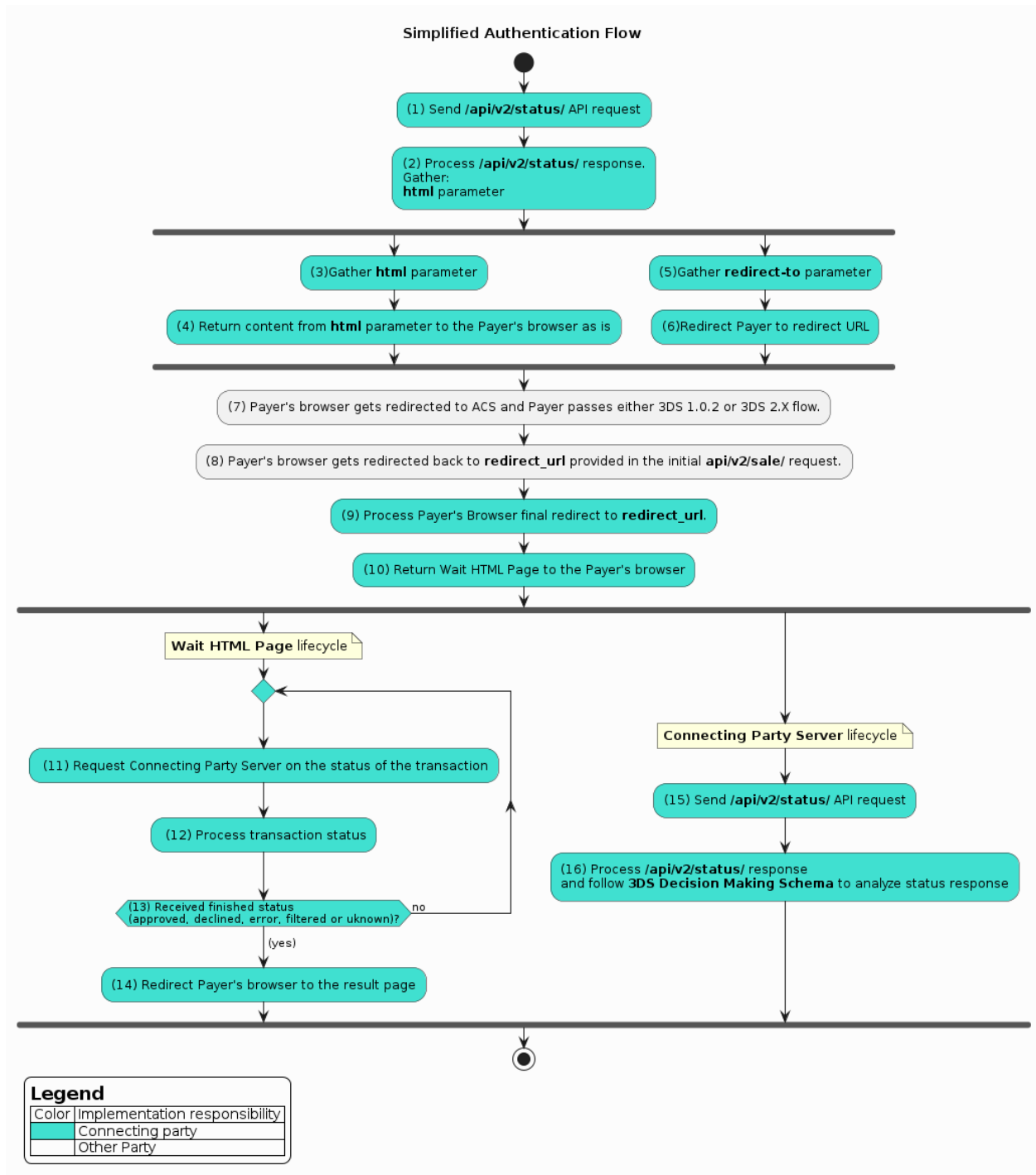
- (1) and (2) To implement order status request see </api/v2/status/>.
- (9) To implement final redirect see [Final redirect](#).
- (10) The HTML wait page on Connecting Party side can have custom design and should communicate with Connecting Party server as described on the diagram.
- (15) and (16) The same as point (1) and (2).

3DS Decision Making Schema



Legend

Color	Implementation responsibility
	Connecting party
	Connecting and other party
	Other Party



5.2 Server-to-Server Sale by SBP

- [Introduction](#)
- [SBP Sale Flow](#)

5.2.1 Introduction

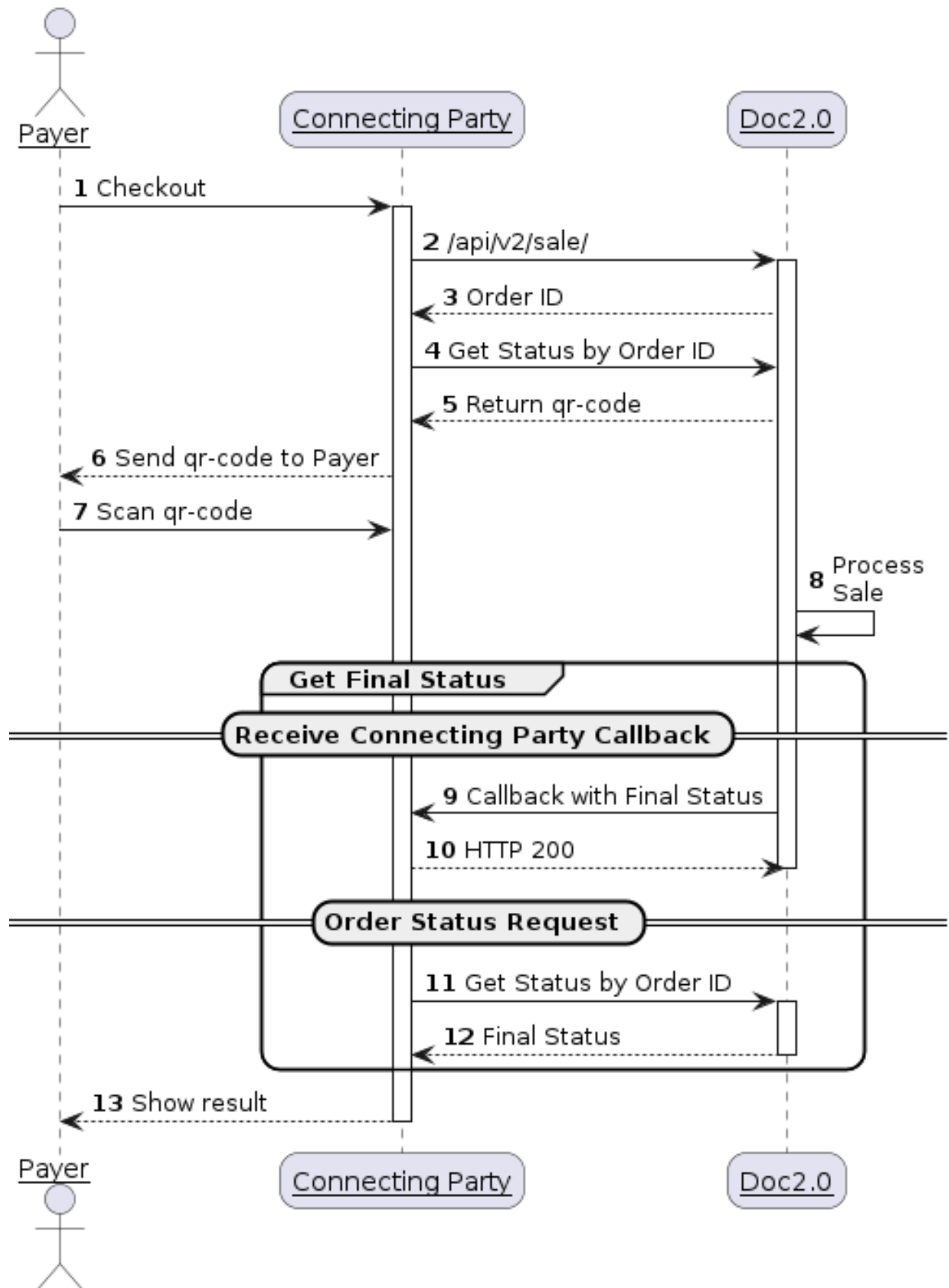
Sale is a type of transaction, in which Payer receives goods or services from Connecting Party in exchange for money or other assets. Server-to-Server integration allows Connecting Party to offer a Payer the option to pay via SBP QR Code, directly received and displayed to Payer by Connecting Party.

5.2.2 SBP Sale Flow

- (2) To implement sale request see [/api/v2/sale/](#).
- (4,11) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until QR code or final status is received.
- (6) Connecting party sends received QR code to Payer.
- (7) Payer scans the QR code to pay.
- (9) To implement callback with final status handling see [Connecting Party Callback](#).

5.3 Server-to-Server Preauth, Capture and Cancel

- [Introduction](#)
- [Preauth Flow](#)
- [3DS Decision Making Schema](#)
 - [Non3D Flow](#)
 - [Simplified Authentication Flow](#)
- [Capture Flow](#)
- [Cancel Flow](#)



5.3.1 Introduction

Preauthorization is a transaction type in which bank blocks the specified amount in the Payer's card account and does not allow the cardholder to use this blocked money. It is important to know that the block remains for a definite period of time depending on whether this is a debit or a credit card (usually the maximum block period is 7 days for debit cards and 28 days for credit cards). In this use case Server-to-server Preauth, card data is transferred directly in initiating request.

See terms definitions (Connecting Party, 3DS Method, etc) in [Glossary](#).

[Capture](#) is a transaction followed after preauth which deducts the locked amount from Payer's card.

[Cancel](#) is opposite of Capture which cancels the deduction and returns locked amount back to Payer's card.

5.3.2 Preauth Flow

(2) To implement preauth request see </api/v2/preauth/>. See [3DS Overview](#) to get more information about 3DS flow. See [3DS Decision Making Schema](#) and [3DS Implementation Scenarios](#) to correctly implement 3DS flow for this Use-Case.

(5) To implement callback with final status handling see [Connecting Party Callback](#).

(7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

5.3.3 3DS Decision Making Schema

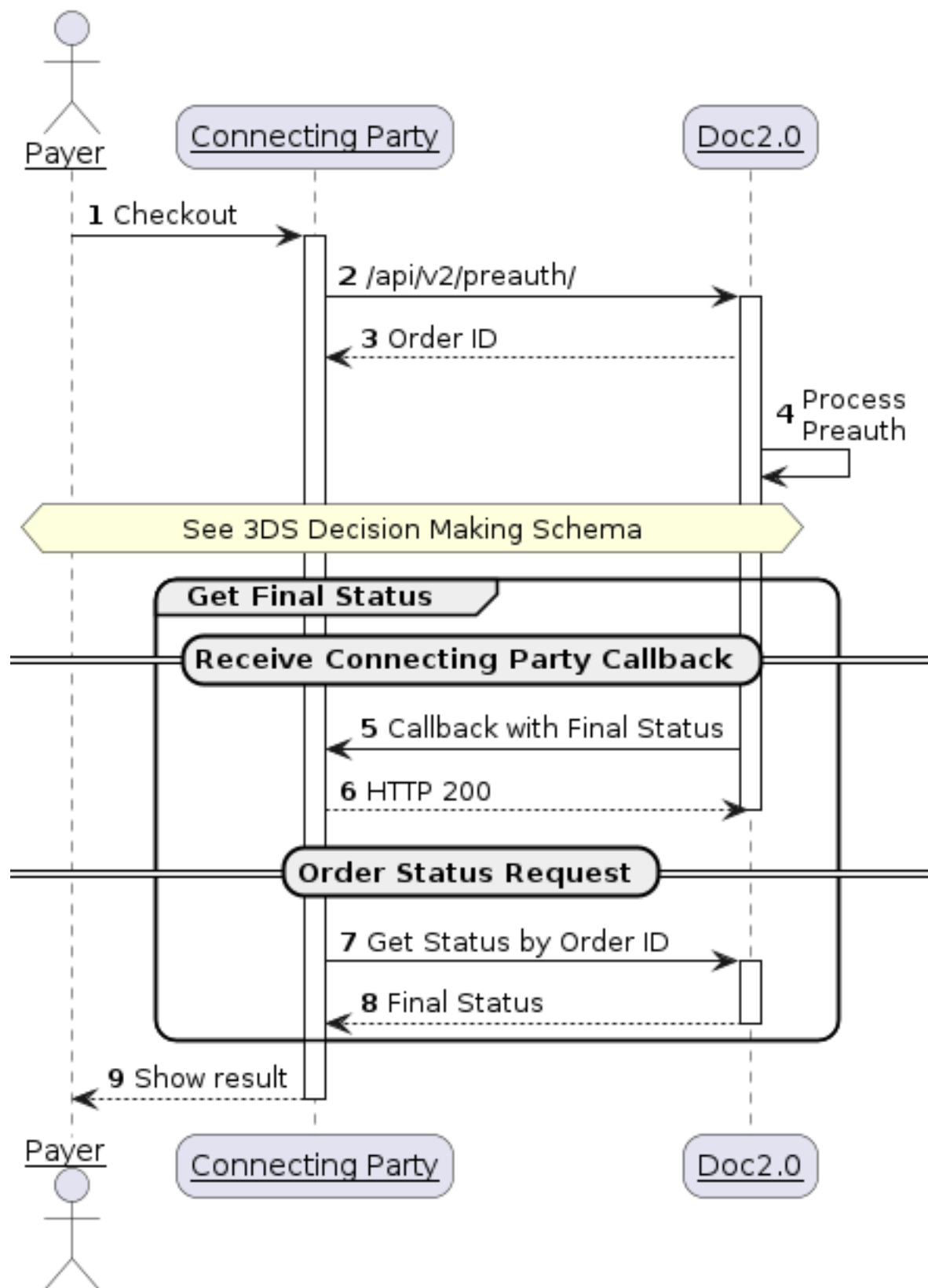
Connecting party has to implement all steps marked in green and purple. Below are the description for steps which reference specific API commands according to the step ID:

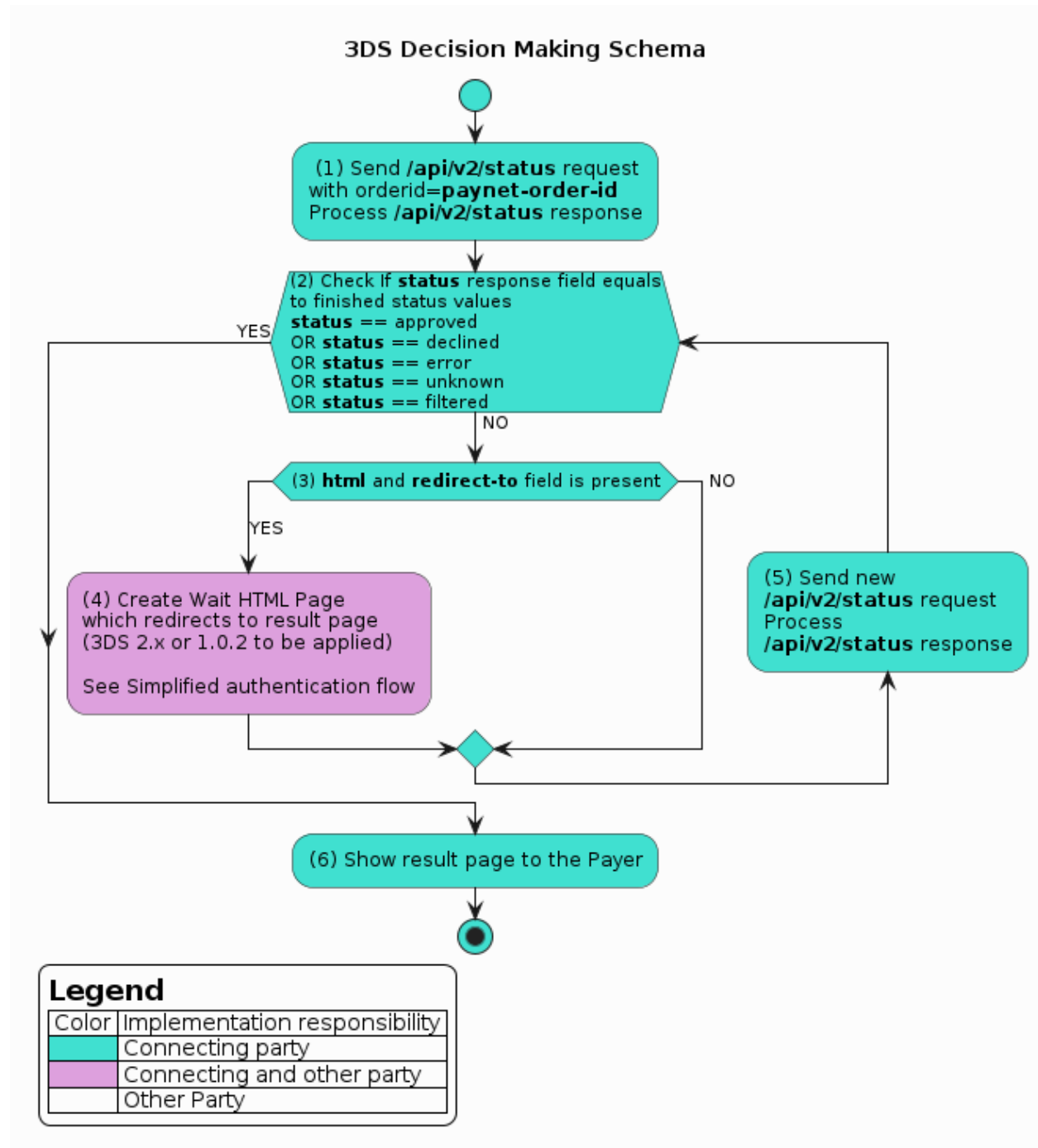
(1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

(4) If html and redirect-to fields are present, see [Simplified authentication flow with html page](#).

(5) The same as point (1).

Note: The [3DS decision making schema](#) is showcasing 3DS being initiated and performed by Payment Gateway. 3DS is initiated by Payment Gateway and performed on Connecting Party side., see [3DS Overview](#)





Non3D Flow

Sale transaction should be considered as non3D (no 3DS authentication) if all conditions are met:

1. steps 1-2-(5)-6 of [3DS decision making schema](#) were followed.
2. tds_status, html and redirect-to parameters were not present.
3. transaction received final status (approved, declined, error, filtered).

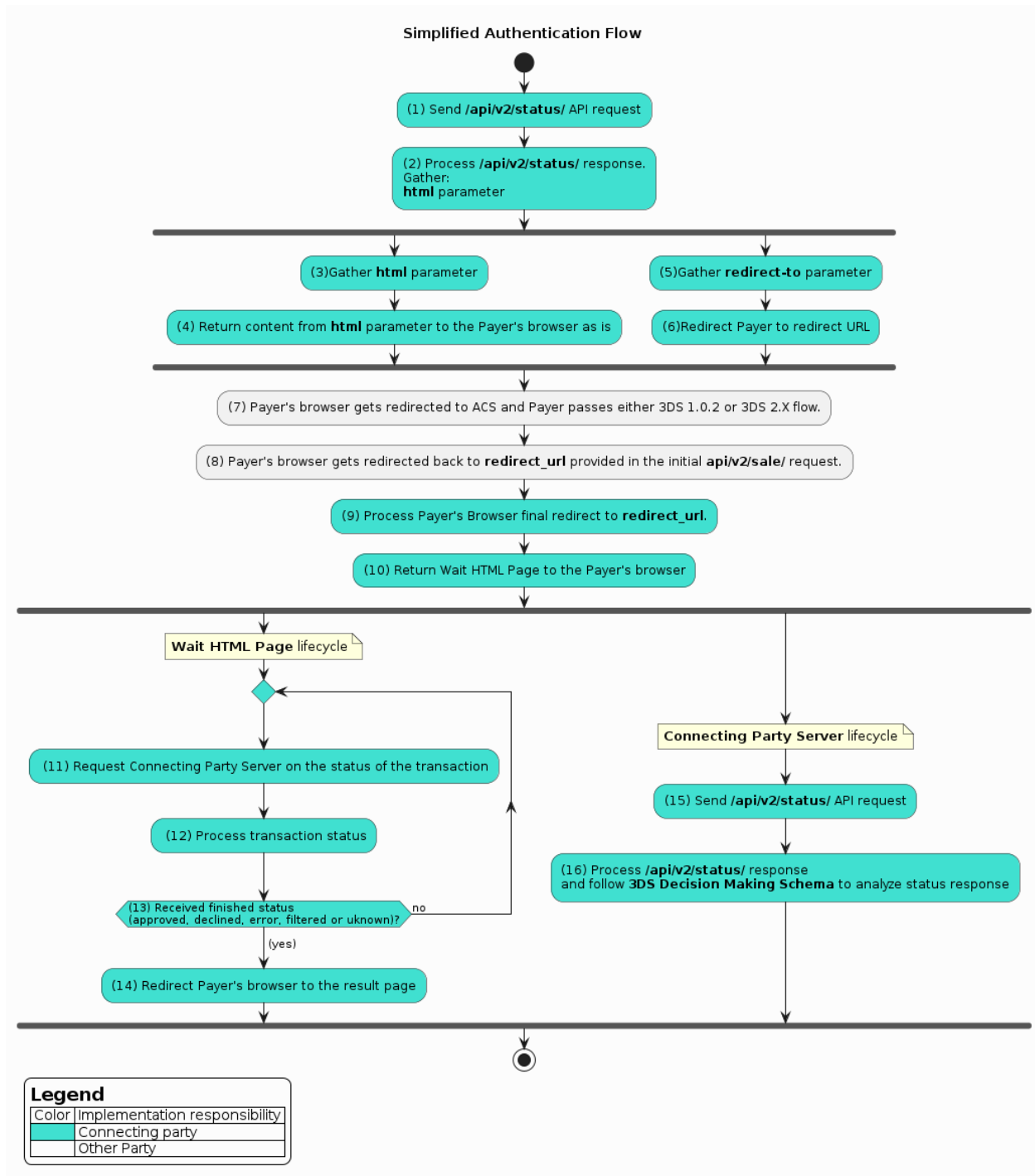
Note: Please note that transaction status “unknown” might appear for both 3DS and non3D transactions. See details in [Statuses](#).

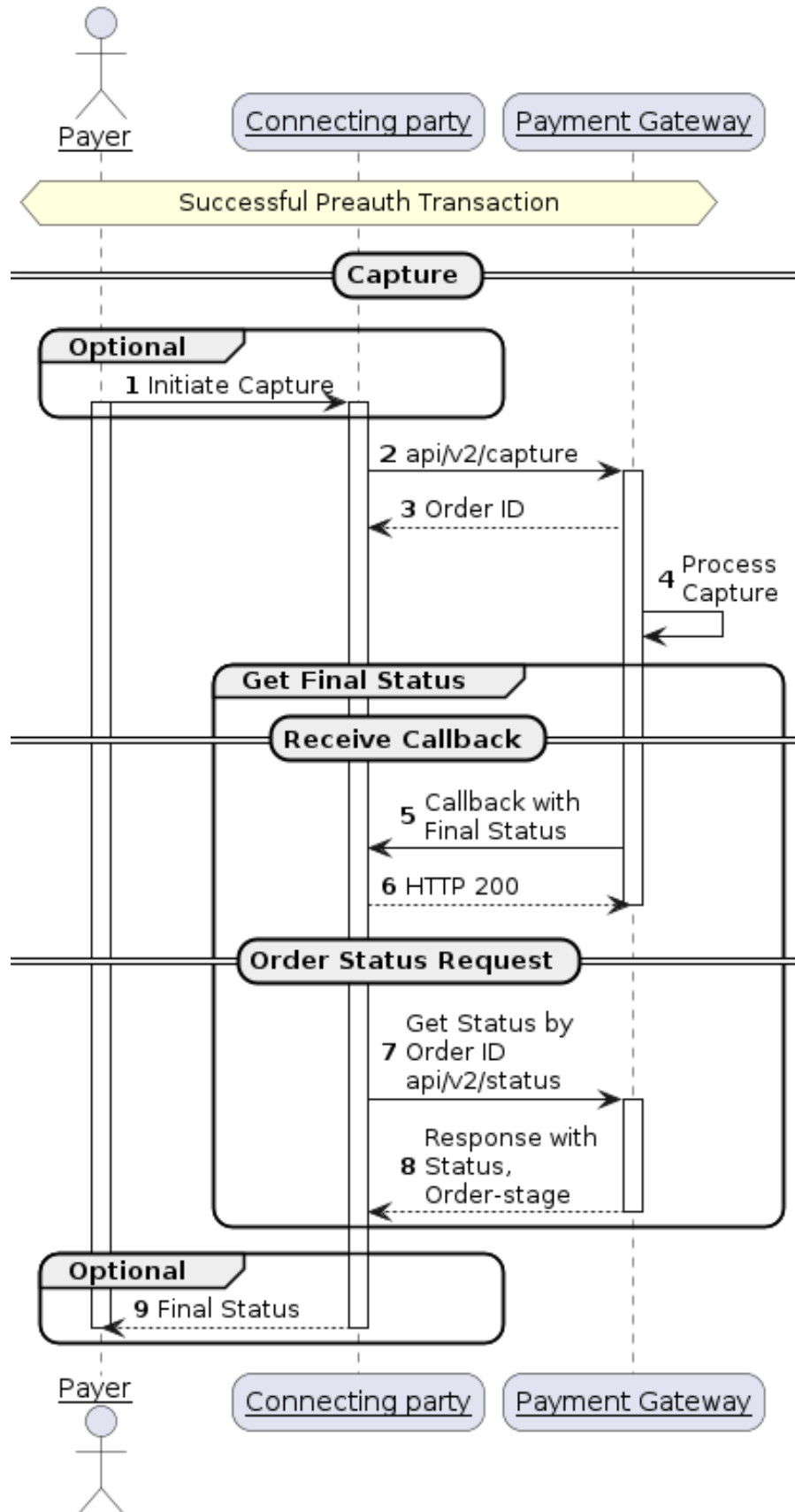
Simplified Authentication Flow

- (1) and (2) To implement order status request see </api/v2/status/>.
- (9) To implement final redirect see [Final redirect](#).
- (10) The HTML wait page on Connecting Party side can have custom design and should communicate with Connecting Party server as described on the diagram.
- (15) and (16) The same as point (1) and (2).

5.3.4 Capture Flow

- (1) Capture can be initiated by Connecting Party based on internal business process or Payer's request.
- (2) To implement capture request see </api/v2/capture/>.
- (5) Callback for Capture will be sent only if notify_url was provided in initial transaction request or additional callback URL for Capture transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Capture will not be sent. To implement callback with final status handling see [Connecting Party Callback](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.





5.3.5 Cancel Flow

- (1) Cancel can be initiated by Connecting Party based on internal business process or Payer's request.
- (2) To implement cancel request see [/api/v2/return/](#).
- (5) Callback for Cancel will be sent only if notify_url was provided in initial transaction request or additional callback URL for Cancel transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Cancel will not be sent. To implement callback with final status handling see [Connecting Party Callback](#).
- (7) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.

5.4 Server-to-Server Transfer

- [Introduction](#)
- [Transfer Flow](#)
- [3DS Decision Making Schema](#)
 - [Non3D Flow](#)
 - [Simplified Authentication Flow](#)

5.4.1 Introduction

Transfer is a type of complex transaction that allows to move funds between bank cards ([PAN](#)), card tokens ([RPI](#)) and Connecting Party bank account ([deposit](#)).

See terms definitions (Connecting Party, 3DS Method, etc) in [Glossary](#).

Possible transfer Use-Cases:

PAN to PAN	Transfer from PAN to PAN occurs when transferring funds from one bank card to another, when specifying for each card their card numbers, and not the recurring payment id.
PAN to RPI	Transfer from PAN to RPI occurs when funds are transferred from one bank card for which its number is indicated to another bank card for which its recurring payment id is indicated.
RPI to PAN	Transfer from RPI to PAN occurs when funds are transferred from one bank card for which its recurring payment id is indicated to another bank card for which its number is indicated.

continues on next page

Table 1 – continued from previous page

RPI to RPI	Transfer from RPI to RPI occurs when transferring funds from one bank card to another, when specifying for each card their recurring payment id, and not the card number.
deposit to PAN	Transfer from deposit to PAN occurs when funds are transferred from Connecting Party account to a bank card by its number.
deposit to PRI	Transfer from deposit to RPI occurs when funds are transferred from Connecting Party account to a bank card by recurring payment id.

5.4.2 Transfer Flow

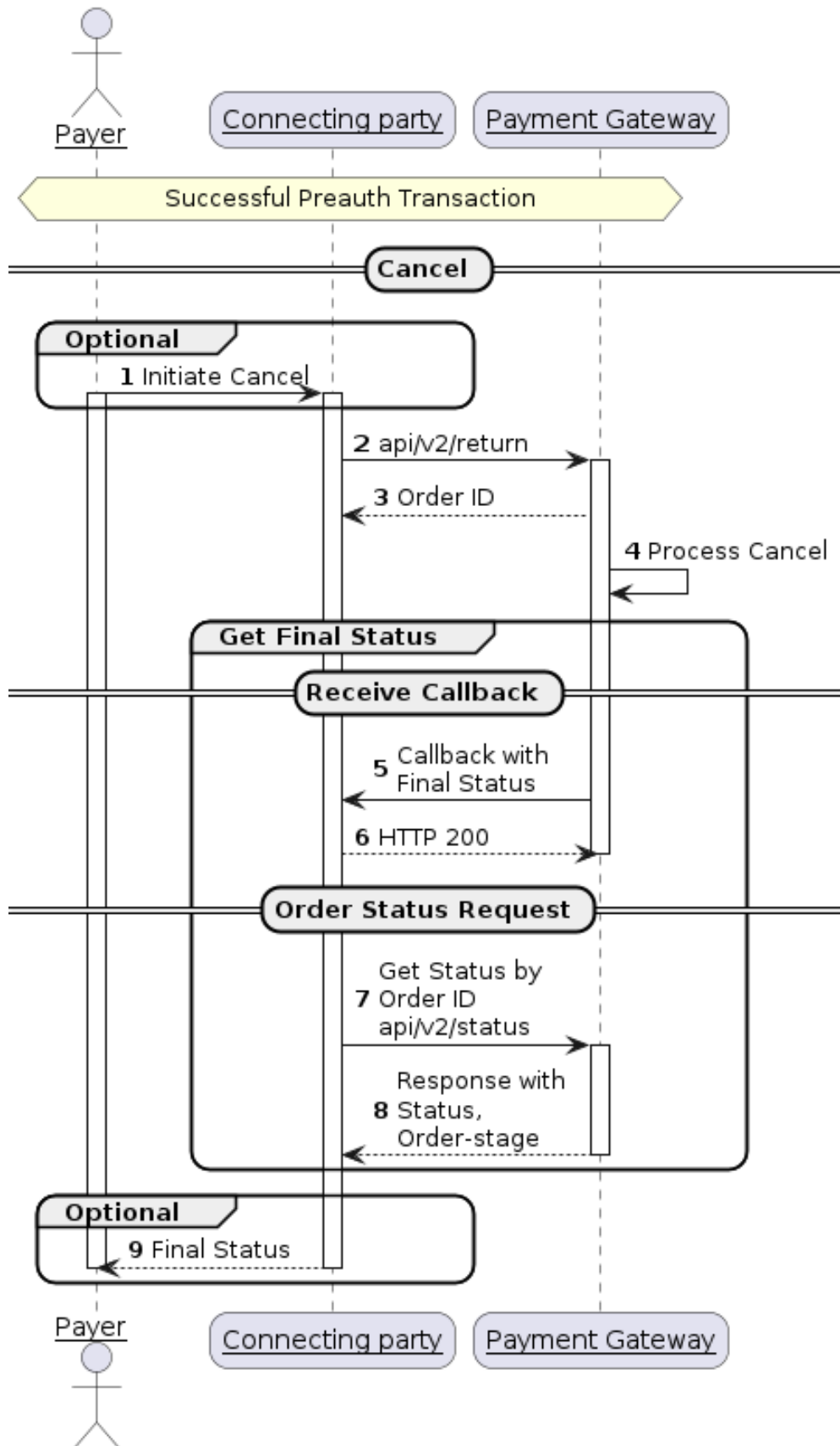
- (1) Transfers between cards are initiated by Sender. Transfer from Connecting Party (deposits) are initiated by Receiver, with the same transaction flow.
- (2) To implement transfer request see </api/v4/transfer/>. See [3DS Overview](#) to get more information about 3DS flow. See [3DS Decision Making Schema](#) and [3DS Implementation Scenarios](#) to correctly implement 3DS flow for this Use-Case. For deposit to PAN and deposit to RPI cases 3DS is not initiated (flow is [non3D](#)).
- (5) To implement callback with final status handling see [Connecting Party Callbacks](#).
- (7) To implement order status see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

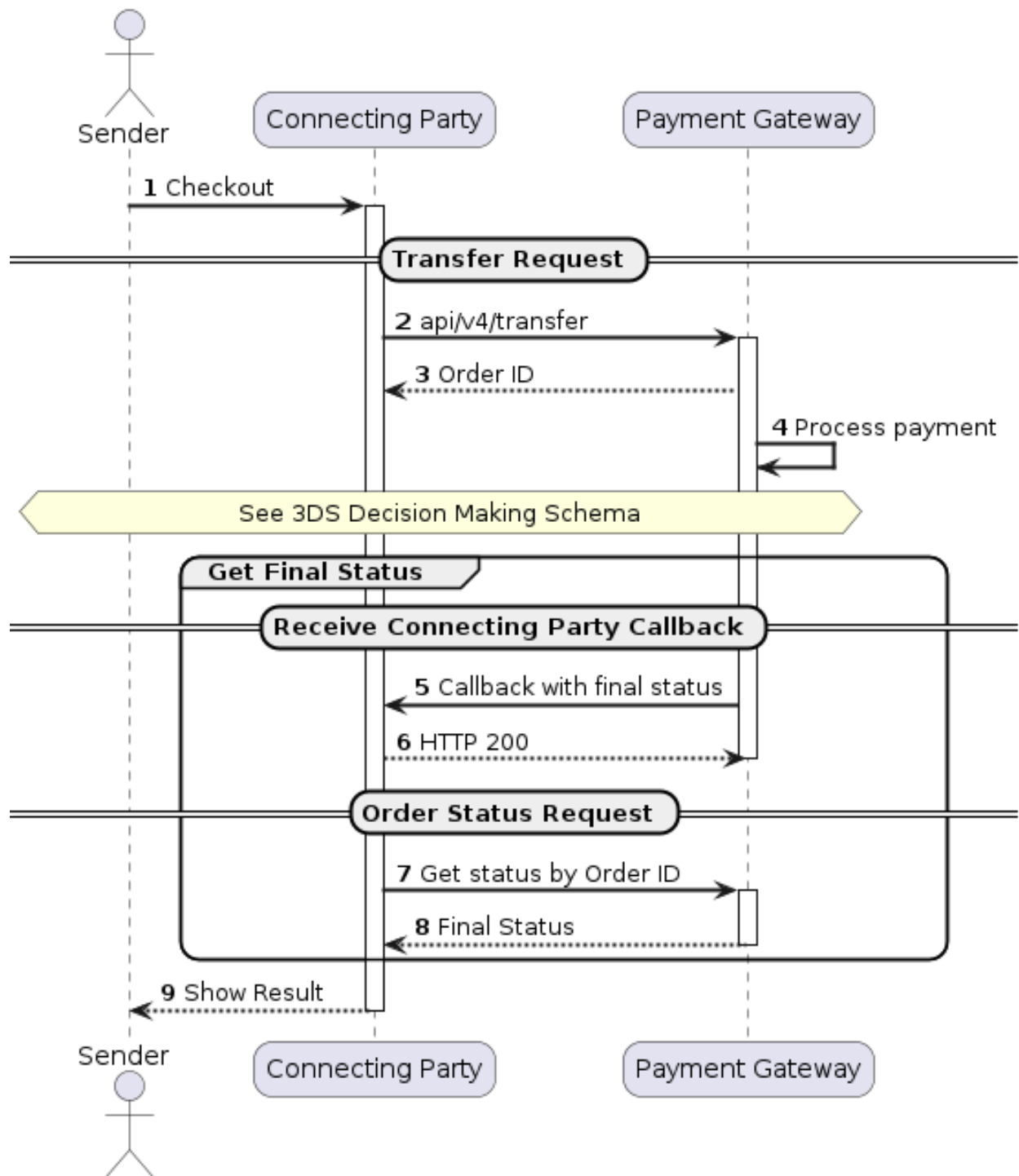
5.4.3 3DS Decision Making Schema

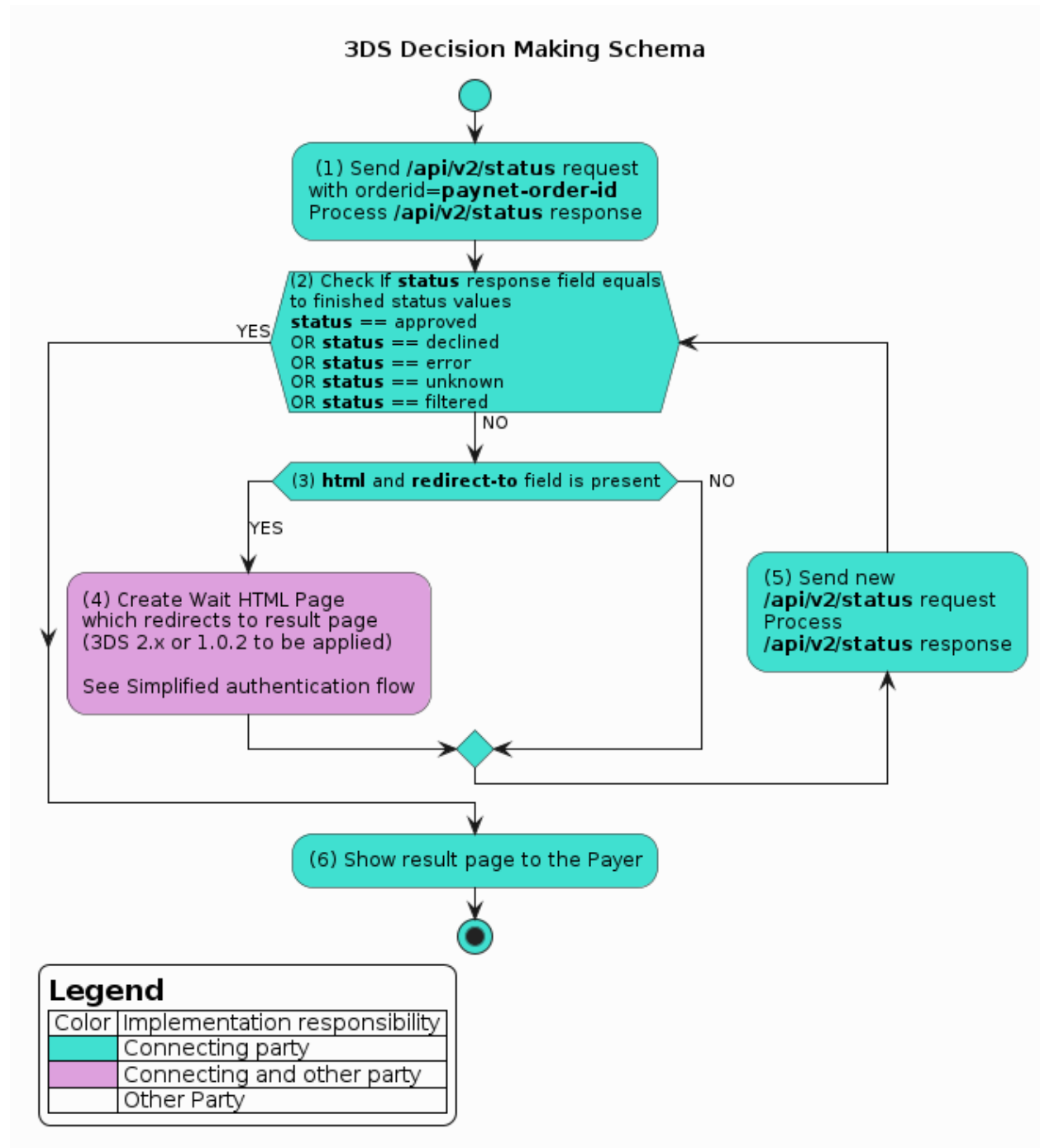
Connecting party has to implement all steps marked in green and purple. Below are the description for steps which reference specific API commands according to the step ID:

- (1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (4) If html and redirect-to fields are present, see [Simplified authentication flow with html page](#).
- (5) The same as point (1).

Note: The [3DS decision making schema](#) is showcasing 3DS being initiated and performed by Payment Gateway. 3DS is initiated by Payment Gateway and performed on Connecting Party side., see [3DS Overview](#)







Non3D Flow

Sale transaction should be considered as non3D (no 3DS authentication) if all conditions are met:

1. Steps 1-2-(5)-6 of [3DS decision making schema](#) were followed.
2. tds_status, html and redirect-to parameters were not present.
3. Transaction received final status (approved, declined, error, filtered).

Note: Please note that transaction status “unknown” might appear for both 3DS and non3D transactions. See details in [Statuses](#).

Simplified Authentication Flow

- (1) and (2) To implement order status request see </api/v2/status/>.
(9) To implement final redirect see [Final redirect](#).
(10) The HTML wait page on Connecting Party side can have custom design and should communicate with Connecting Party server as described on the diagram.
(15) and (16) The same as point (1) and (2).

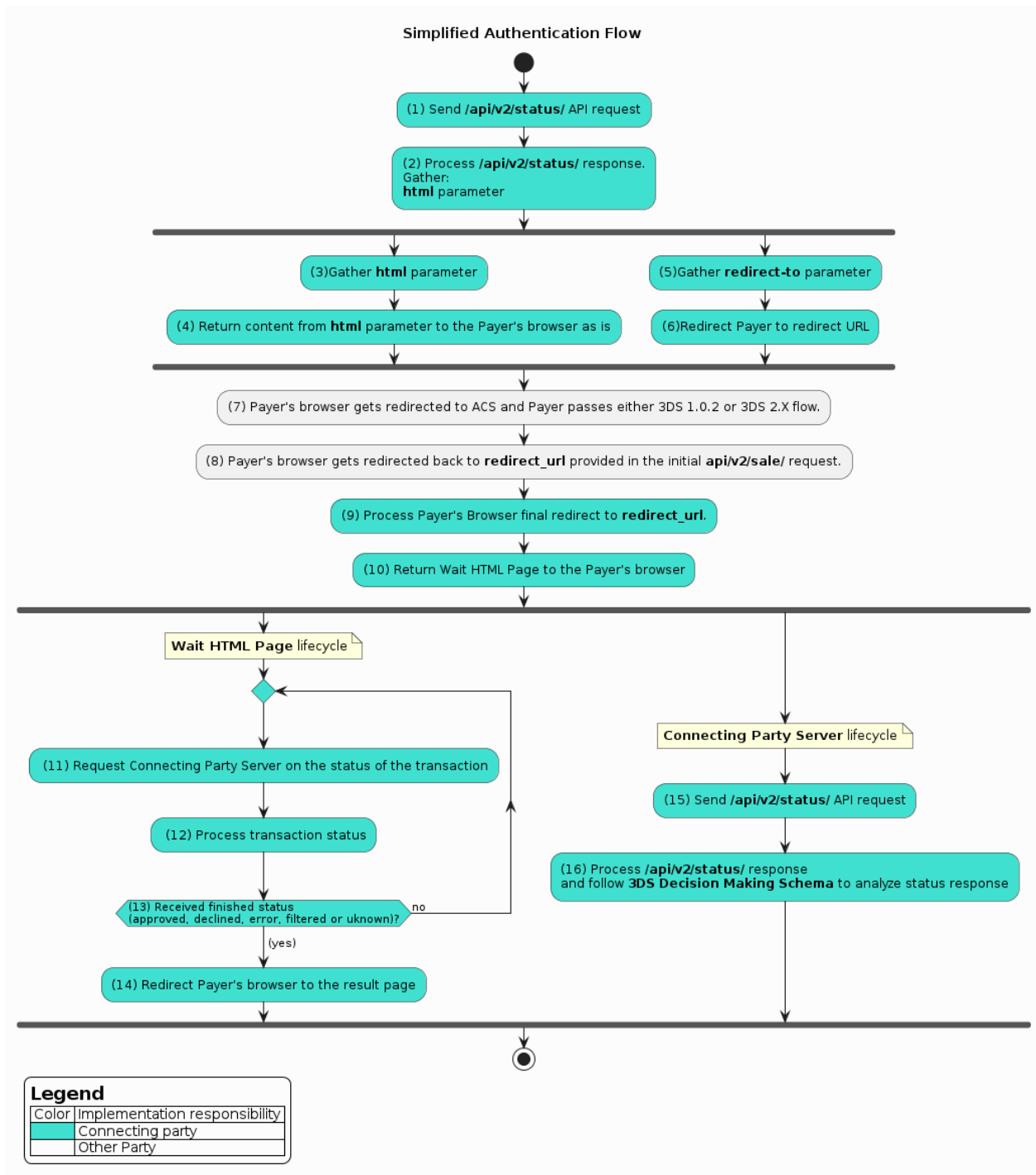
5.5 Server-to-Server Payout

- [Introduction](#)
 - [Payout Flow](#)

5.5.1 Introduction

Payout is a type of transaction which results in funds transfer from Connecting Party banking account to customer (receiver) banking account or digital wallet. Payout transaction in most cases is used for bank account funding.

See terms definitions in [Glossary](#).



5.5.2 Payout Flow

- (1) Payout can be initiated by Connecting Party based on internal business model or Receiver's request.
- (2) To implement payout transaction see [/api/v4/payout](#).
- (5) Some payout methods require the Receiver to fill the additional data on the form. The form to redirect the customer will return in status response in redirect-to parameter.
- (8) The Receiver submits the payout form.
- (9) The Receiver gets redirected back to Connecting Party. See [Final redirect](#).
- (11) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (13) To implement callback with final status handling see [Connecting Party Callback](#).
- (15) Final Status can be sent by Connecting Party based on internal business model or by Receiver's request.

5.6 Server-to-Server Account Verification

- [Introduction](#)
- [Account Verification Flow](#)
- [3DS Decision Making Schema](#)
 - [Non3D Flow](#)
 - [Simplified Authentication Flow](#)

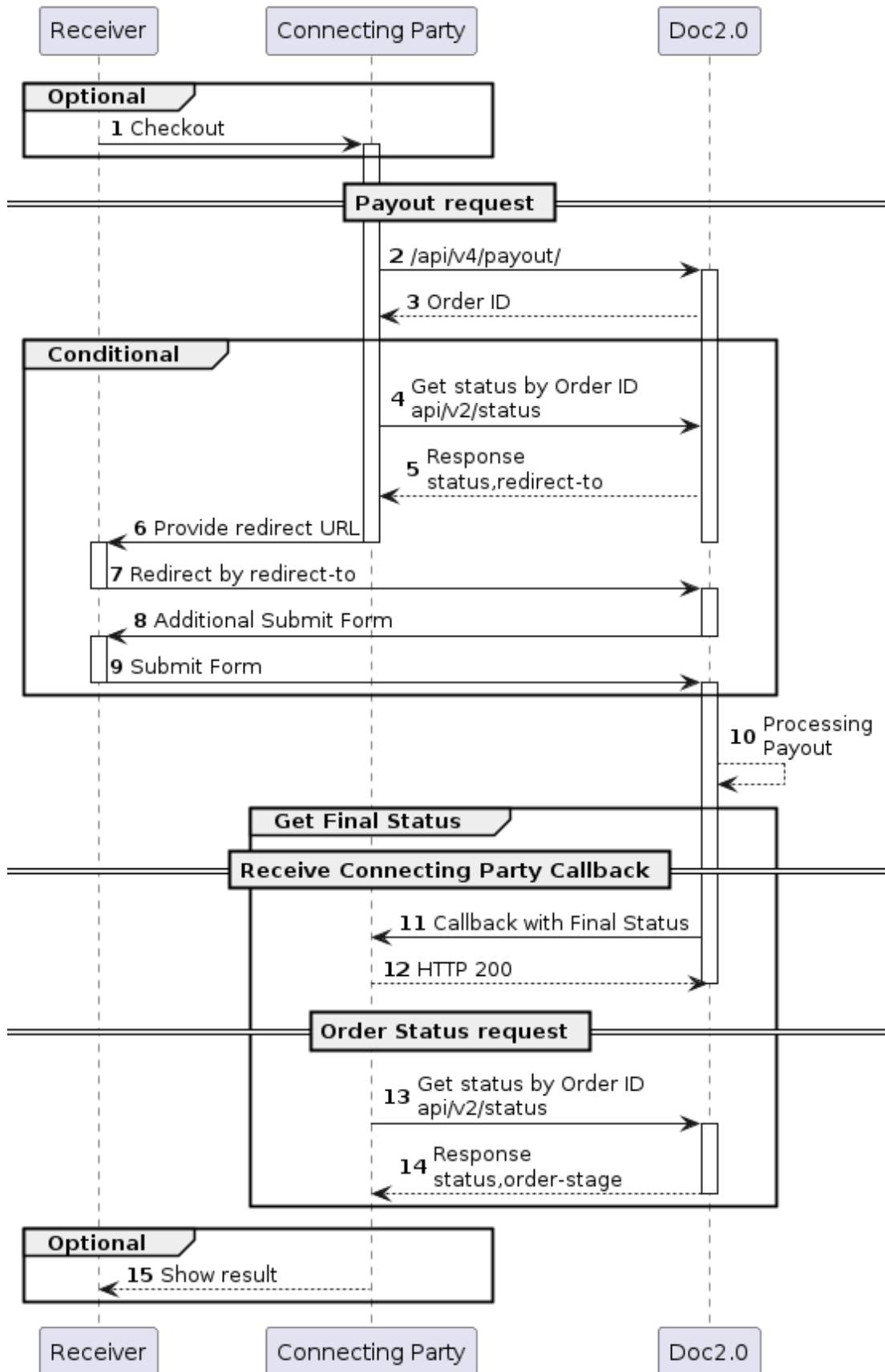
5.6.1 Introduction

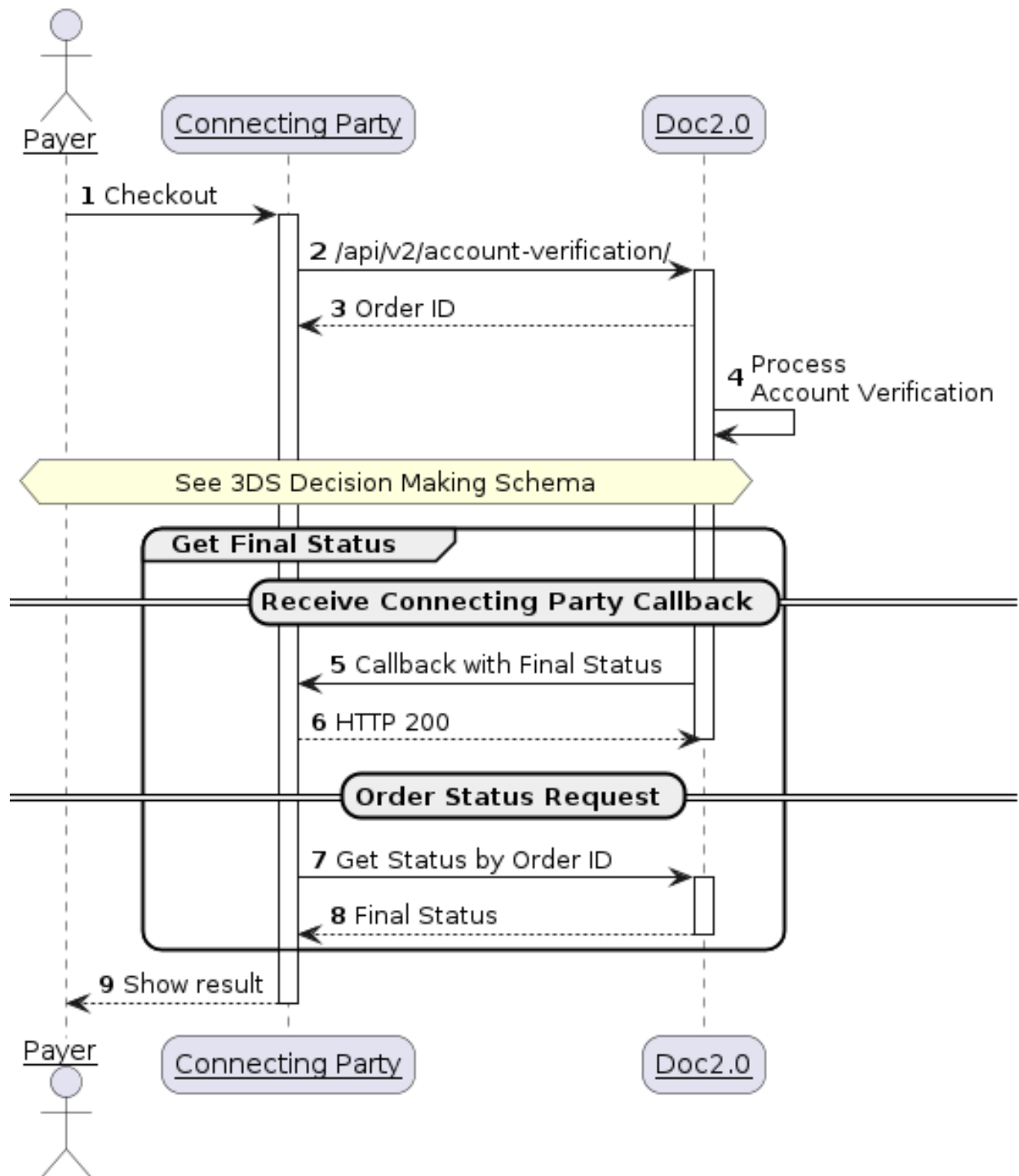
Account verification is a non-financial transaction, which allows to validate Payer's card account information. This verification procedure helps to prevent potential fraudulent activity and meet industry-specific compliance obligations. In this Use-Case, Payer's cardholder data is transferred directly in initial request from Connecting Party to Payment Gateway.

See terms definitions (Connecting Party, 3DS Method, etc) in [Glossary](#).

5.6.2 Account Verification Flow

- (2) To implement account verification request see [/api/v2/account-verification/](#). See [3DS Overview](#) to get more information about 3DS flow. See [3DS Decision Making Schema](#) and [3DS Implementation Scenarios](#) to correctly implement 3DS flow for this Use-Case.
- (5) To implement callback with final status handling see [Connecting Party Callback](#).
- (7) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.





5.6.3 3DS Decision Making Schema

Connecting party has to implement all steps marked in green and purple. Below are the description for steps which reference specific API commands according to the step ID:

- (1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (4) If html and redirect-to fields are present, see [Simplified authentication flow with html page](#).
- (5) The same as point (1).

Note: The [3DS decision making schema](#) is showcasing 3DS being initiated and performed by Payment Gateway. 3DS is initiated by Payment Gateway and performed on Connecting Party side., see [3DS Overview](#)

Non3D Flow

Sale transaction should be considered as non3D (no 3DS authentication) if all conditions are met:

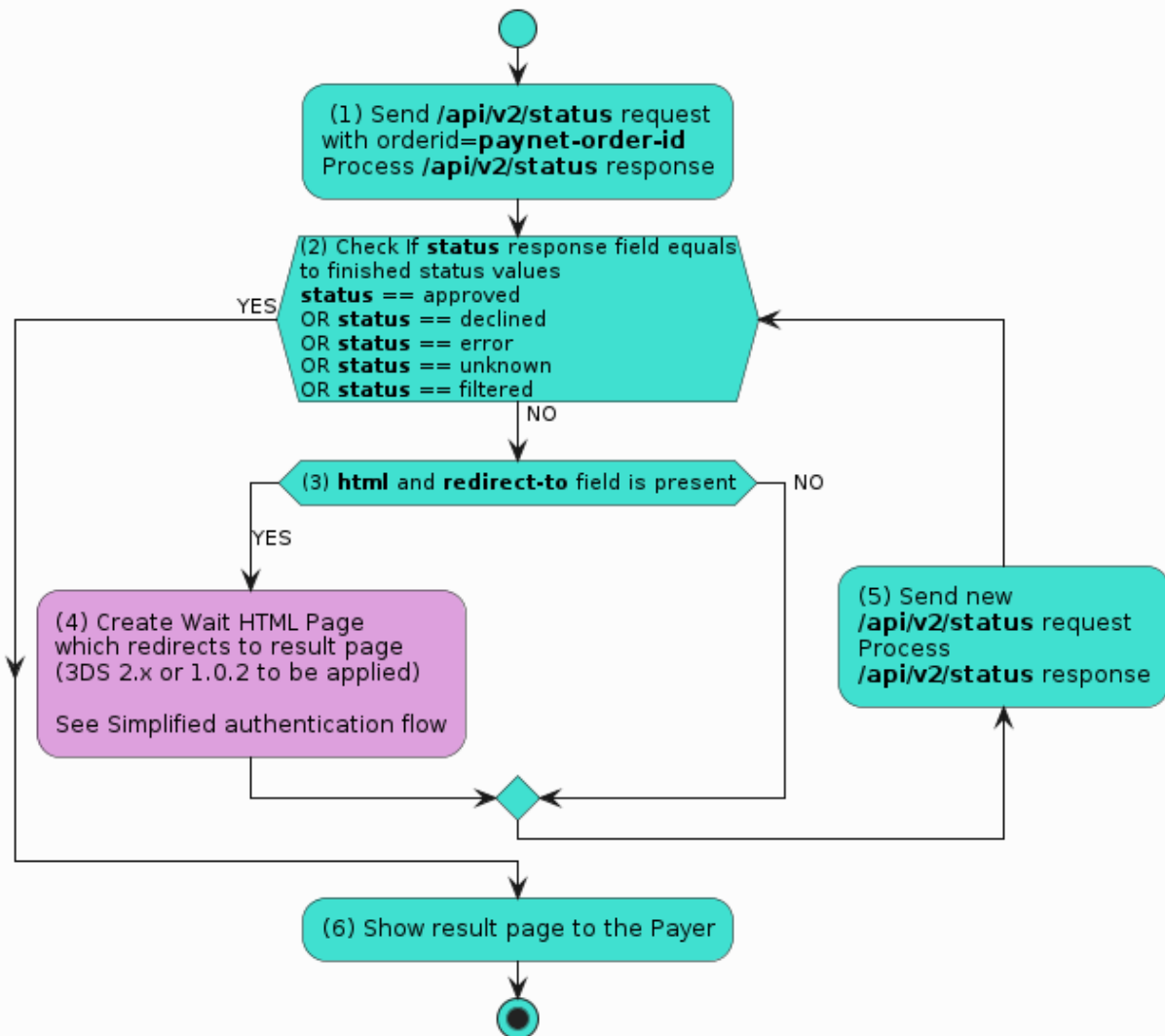
- 1. Steps 1-2-(5)-6 of [3DS decision making schema](#) were followed.
- 2. tds_status, html and redirect-to parameters were not present.
- 3. Transaction received final status (approved, declined, error, filtered).

Note: Please note that transaction status “unknown” might appear for both 3DS and non3D transactions. See details in [Statuses](#).

Simplified Authentication Flow

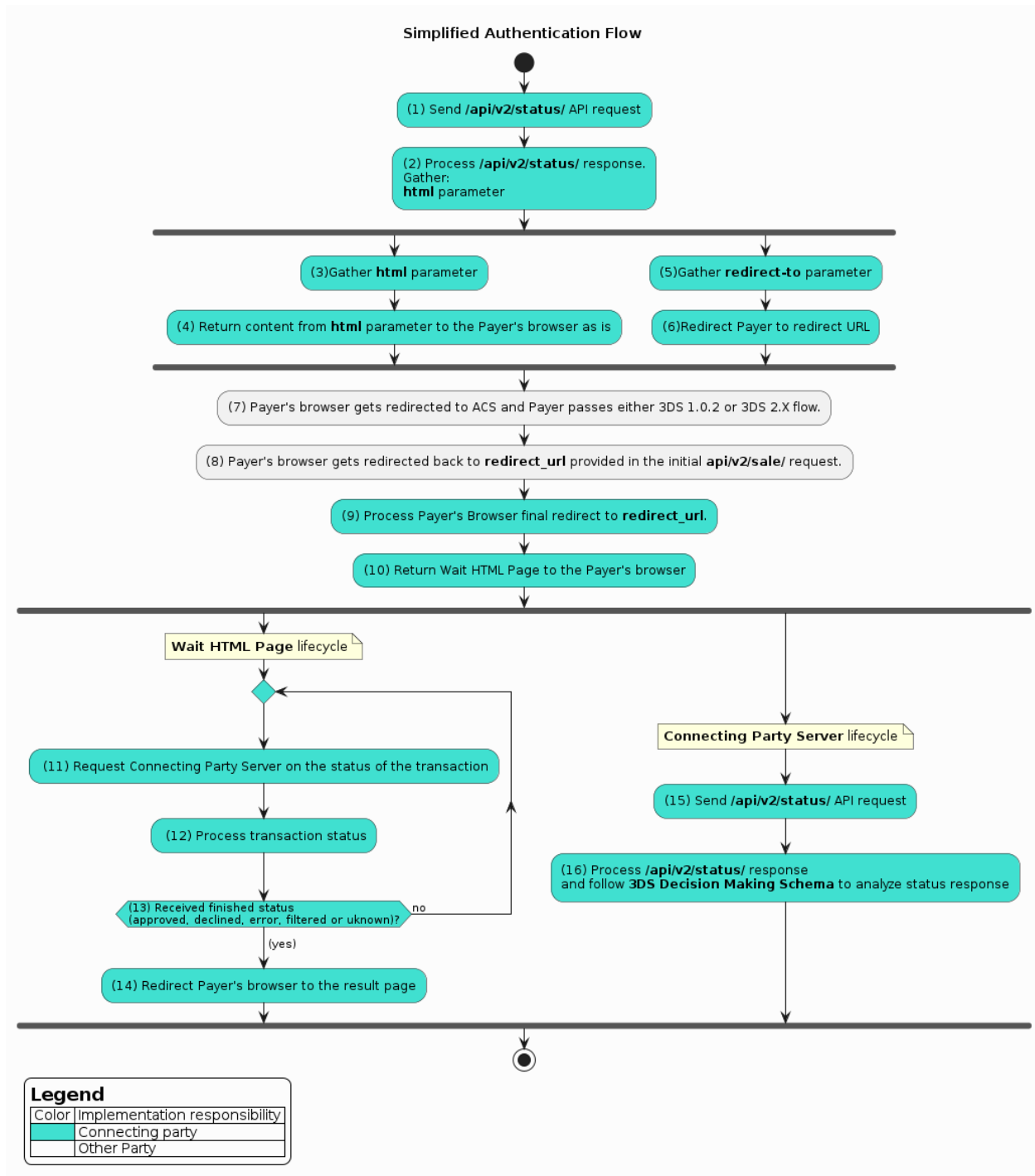
- (1) and (2) To implement order status request see </api/v2/status/>.
- (9) To implement final redirect see [Final redirect](#).
- (10) The HTML wait page on Connecting Party side can have custom design and should communicate with Connecting Party server as described on the diagram.
- (15) and (16) The same as point (1) and (2).

3DS Decision Making Schema



Legend

Color	Implementation responsibility
	Connecting party
	Connecting and other party
	Other Party



5.7 Server-to-Server Account Verification by SBP

- [Introduction](#)
- [SBP Account Verification Flow](#)

5.7.1 Introduction

Account verification is a non-financial transaction, which allows to validate and register Payer's account information. This verification procedure helps to prevent potential fraudulent activity and meet industry-specific compliance obligations, as well as to simplify future payments with this account. Server-to-Server integration allows Connecting Party to offer a Payer the option to validate the account information and register via SBP QR Code, directly received and displayed to Payer by Connecting Party.

5.7.2 SBP Account Verification Flow

- (2) To implement sale request see </api/v2/account-verification/>.
- (4,11) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until QR code or final status is received.
- (6) Connecting party sends received QR code to Payer.
- (7) Payer scans the QR code to pay.
- (9) To implement callback with final status handling see [Connecting Party Callback](#).

5.8 Sale Form

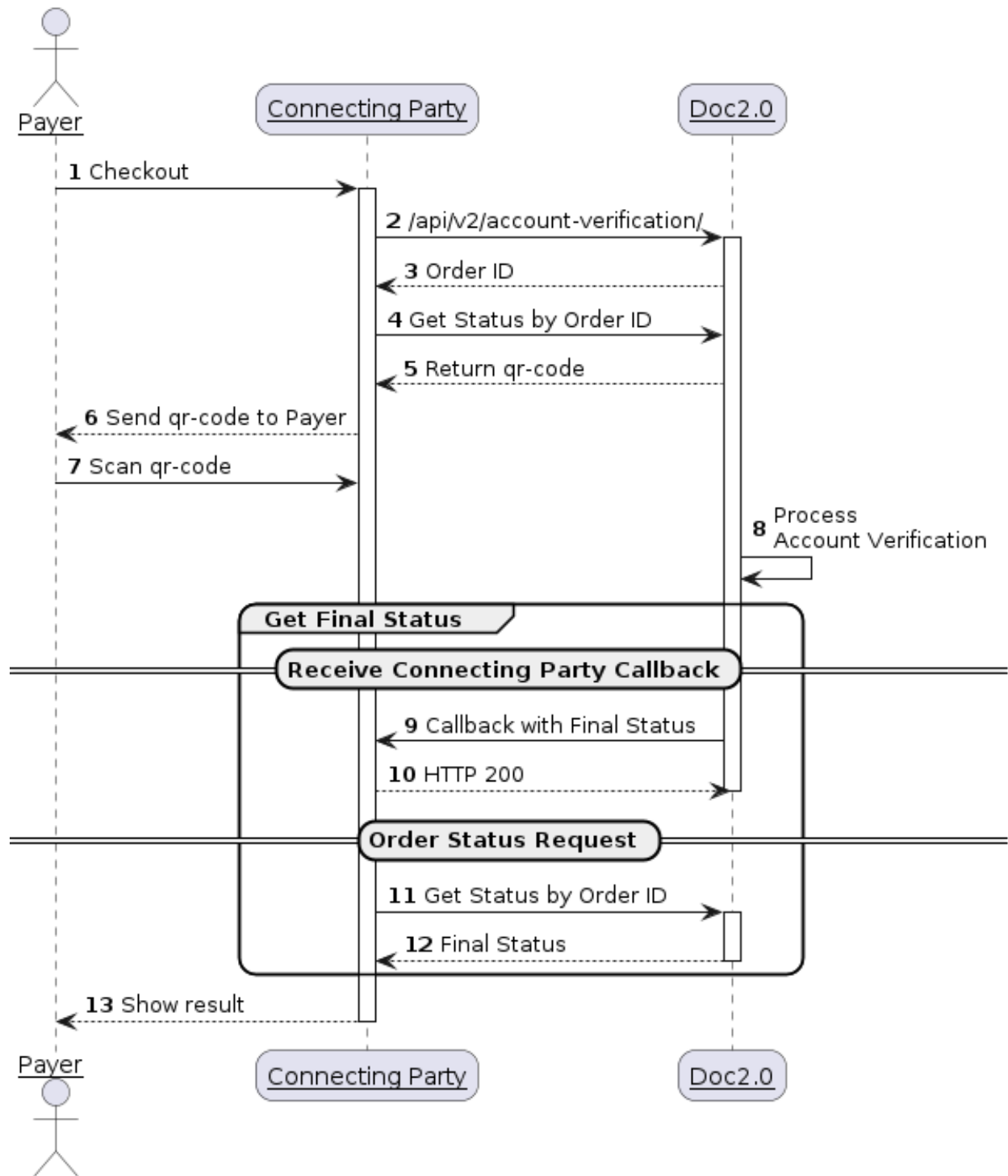
- [Introduction](#)
- [Sale Form Flow](#)

5.8.1 Introduction

Sale is a type of transaction, in which Payer receives goods or services from Connecting Party in exchange for money or other assets. Sale Form integration allows Connecting Party to exclude itself from storing, processing, or transmitting Payer's cardholder data or other sensitive payment details. Such data is submitted by Payer on Doc2.0 hosted payment form in PCI DSS certified environment.

All forms can be customized. Follow the [Forms Customization](#) reference to see examples and macros of [Payment Page Customization](#), [Prefilled Cardholder Data in Payment Page](#), [Wait Page Customization](#) and [Finish Page Customization](#).

See terms definitions in [Glossary](#).



5.8.2 Sale Form Flow

- (2) To implement sale-form request see </api/v2/sale-form/>.
(9) To implement final redirect see [Final Redirect](#).
(11,12) To implement callback with final status handling see [Connecting Party Callbacks](#).
(13) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

5.9 SBP Sale Form

- [Introduction](#)
- [SBP Sale Form Flow](#)

5.9.1 Introduction

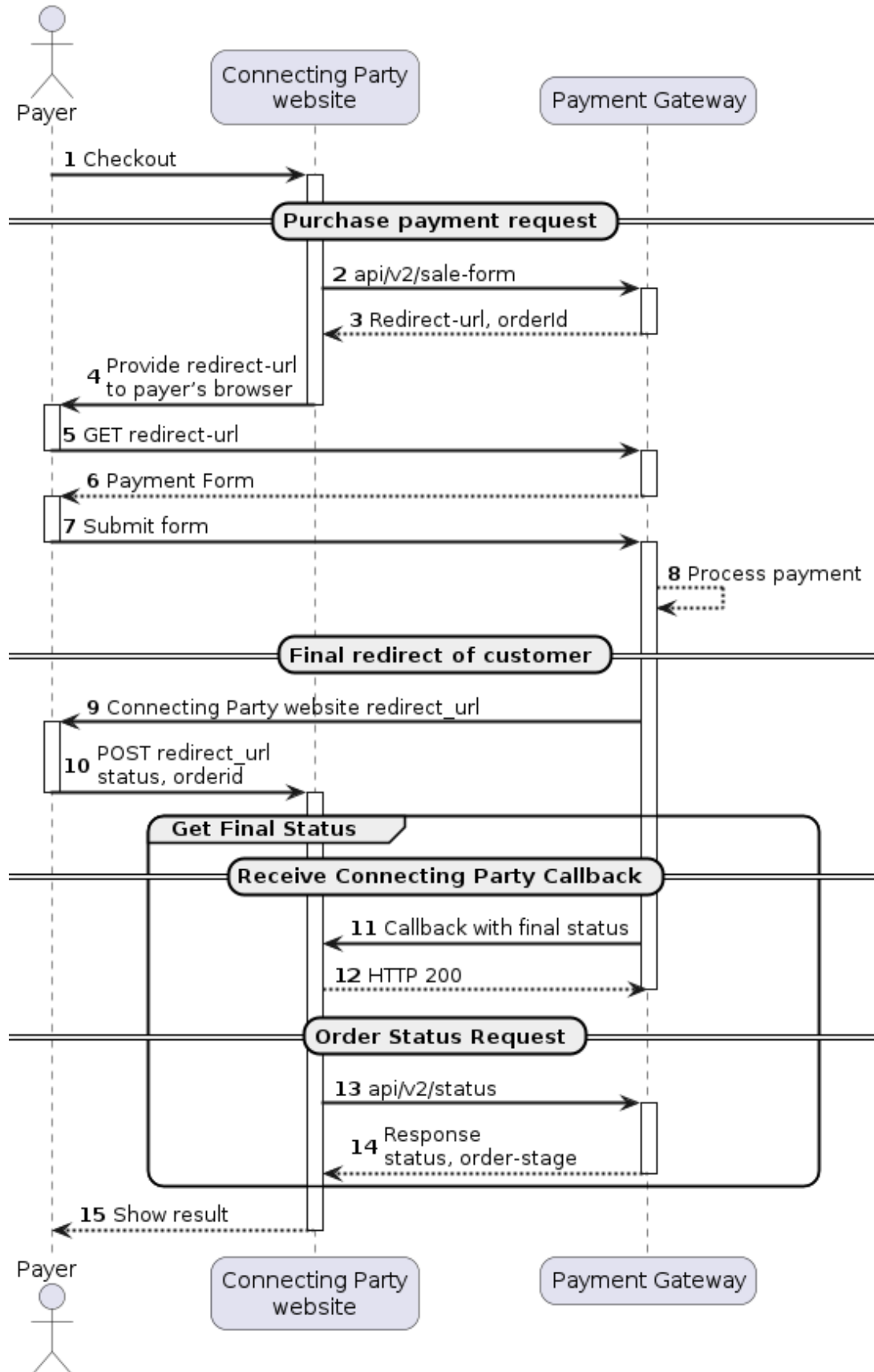
Sale is a type of transaction, in which Payer receives goods or services from Connecting Party in exchange for money or other assets. Sale Form SBP integration allows Connecting Party to offer a Payer the option to pay via SBP QR Code, hosted on Payment Gateway side.

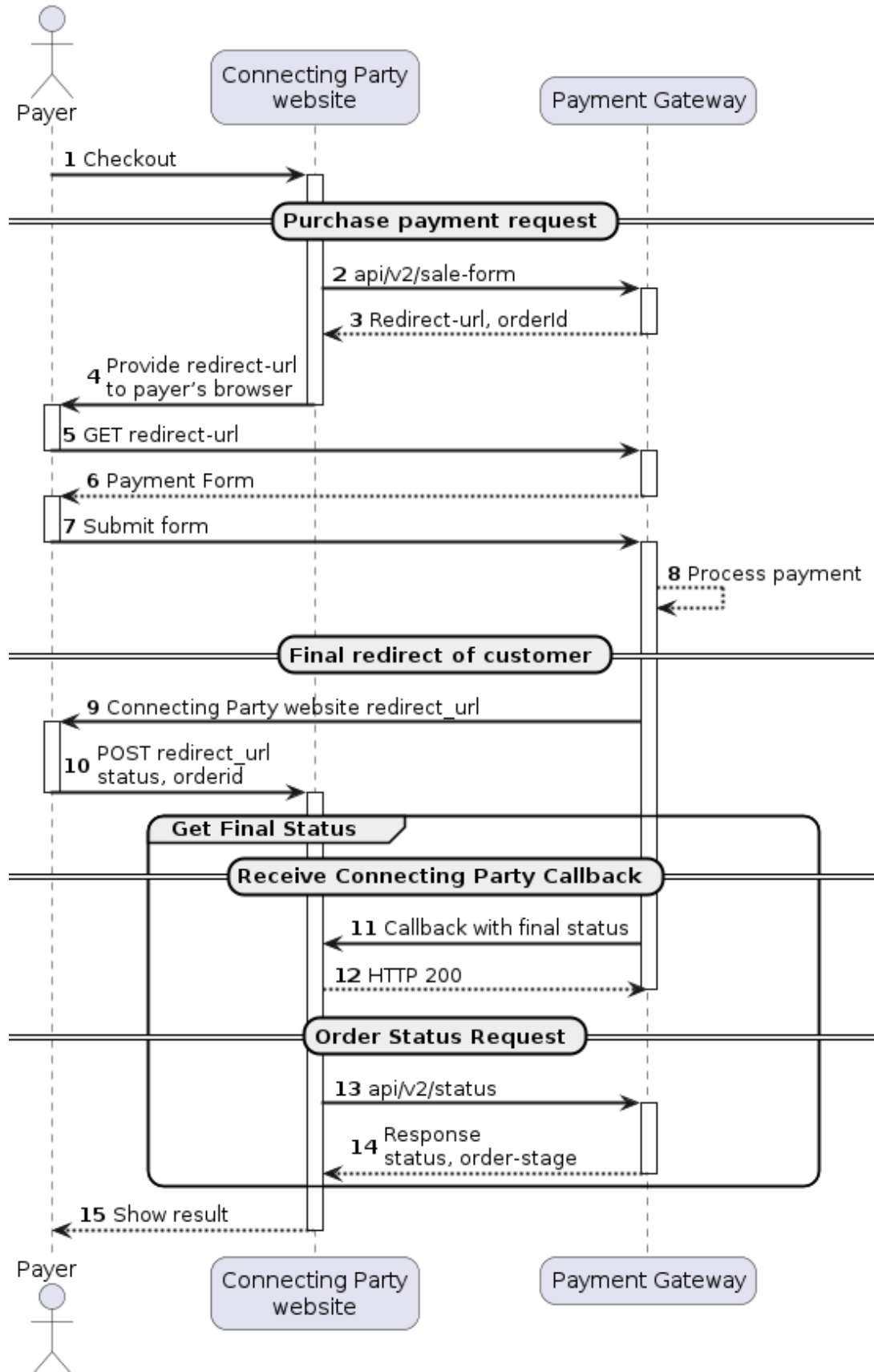
5.9.2 SBP Sale Form Flow

- (2) To implement sale-form request see </api/v2/sale-form/>.
(9) To implement final redirect see [Final Redirect](#).
(11,12) To implement callback with final status handling see [Connecting Party Callbacks](#).
(13) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

5.10 Preauth Form, Capture and Cancel

- [Introduction](#)
- [Preauth Form Flow](#)
- [Capture Flow](#)
- [Cancel Flow](#)





5.10.1 Introduction

Preauthorization is a transaction type in which bank blocks the specified amount in the Payer's card account and does not allow the cardholder to use this blocked money. It is important to know that the block remains for a definite period of time depending on whether this is a debit or a credit card (usually the maximum block period is 7 days for debit cards and 28 days for credit cards). Preauth Form integration allows Connecting Party to exclude itself from storing, processing, or transmitting Payer's cardholder data or other sensitive payment details. Such data is submitted by Payer on Doc2.0 hosted payment form in PCI DSS certified environment.

Capture is a transaction followed after preauth which deducts the locked amount from Payer's card.

Cancel is opposite of Capture which cancels the deduction and returns locked amount back to Payer's card.

All forms can be customized. Follow the [Forms Customization](#) reference to see examples and macros of [Payment Page Customization](#), [Prefilled Cardholder Data in Payment Page](#), [Wait Page Customization](#) and [Finish Page Customization](#).

See terms definitions in [Glossary](#).

5.10.2 Preauth Form Flow

(2) To implement preauth-form request see </api/v2/preauth-form/>.

(9) To implement final redirect see [Final Redirect](#).

(11,12) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

(13) To implement callback with final status handling see [Connecting Party](#).

5.10.3 Capture Flow

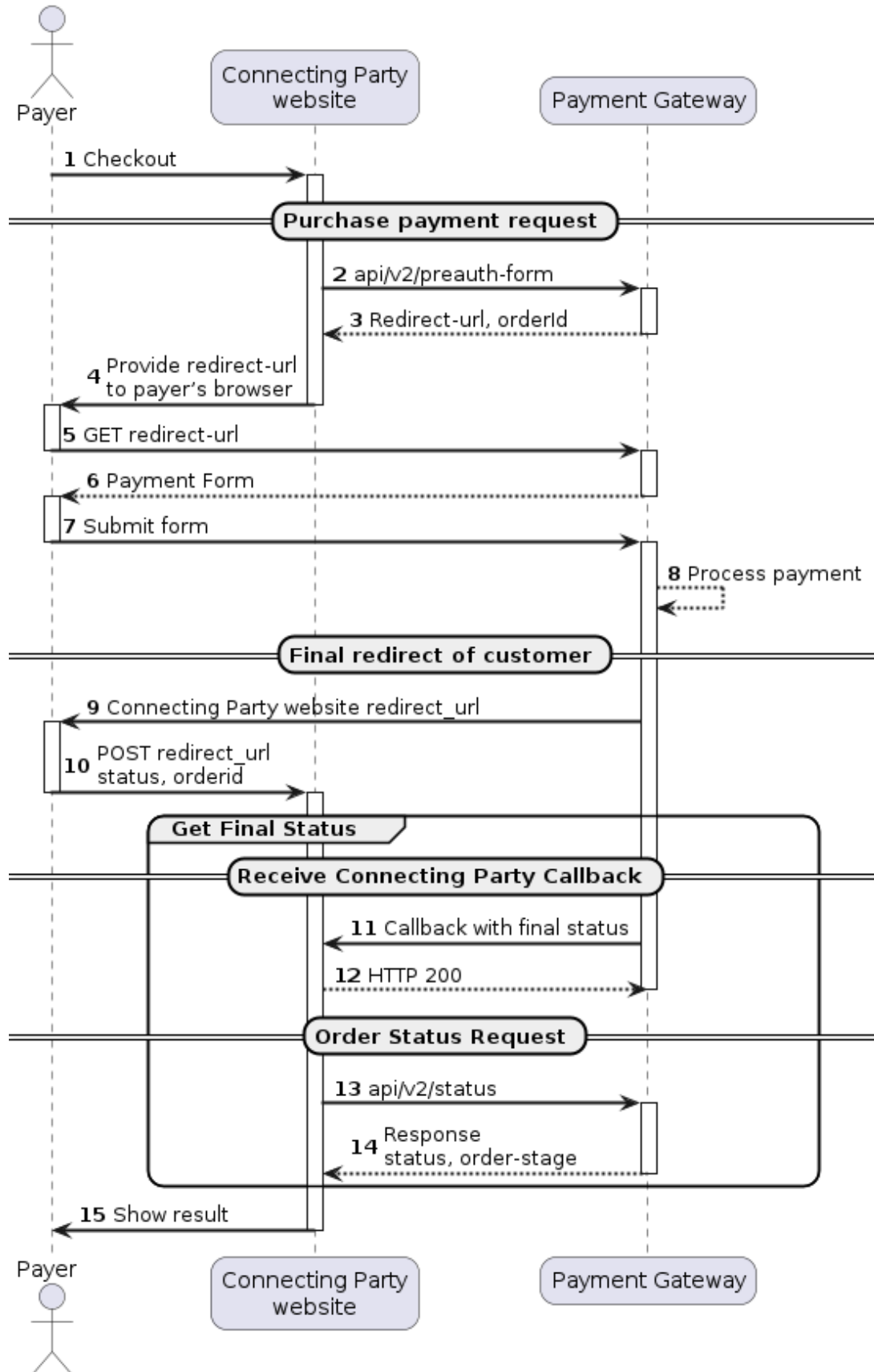
(1) Capture can be initiated by Connecting Party based on internal business model or Payer's request.

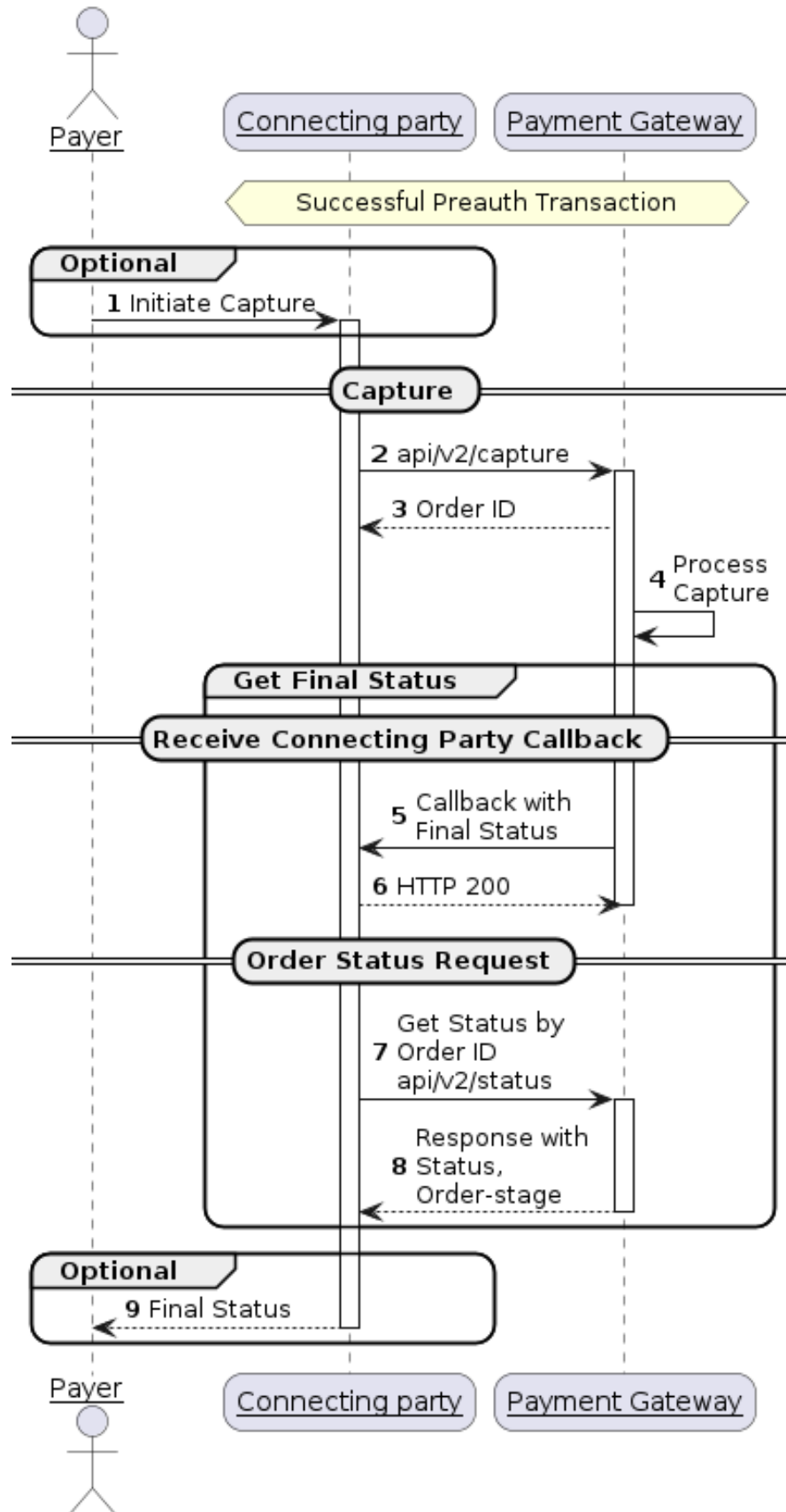
(2) To implement capture request see </api/v2/capture/>.

(5) Callback for Capture will be sent only if notify_url was provided in initial transaction request or additional callback URL for Capture transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Capture will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).

(7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

(9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.





5.10.4 Cancel Flow

- (1) Cancel can be initiated by Connecting Party based on internal business model or Payer's request.
- (2) To implement cancel request see [/api/v2/return/](#).
- (5) Callback for Cancel will be sent only if notify_url was provided in initial transaction request or additional callback URL for Cancel transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Cancel will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).
- (7) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.

5.11 Transfer Form

- [Introduction](#)
- [Transfer Flow](#)

5.11.1 Introduction

Transfer is a type of complex transaction that allows to move funds between bank cards ([PAN](#)), card tokens ([RPI](#)) and Connecting Party bank account ([deposits](#)).

Transfer-form integration is relevant for Connecting Party who are not able to accept sender card details (Connecting Party's website must complete PCI DSS certification). Using Transfer Form integration Connecting Party is released of accepting payment details. In this case the accepting and processing of payment details is carried out on the Doc2.0 side (if PAN is not sent). This type of integration also allows the Sender (for transfers between cards) or Receiver (for deposit to card) to submit their PAN on Doc2.0 side.

All forms can be customized. Follow the [Forms Customization](#) reference to see examples and macros of [Payment Page Customization](#), [Prefilled Cardholder Data in Payment Page](#), [Wait Page Customization](#) and [Finish Page Customization](#).

See terms definitions in [Glossary](#).

Transfer Form Use-Cases diagram:

Possible transfer form Use-Cases:

Form to PAN	Transfer from Form to PAN occurs when transferring funds from payment details indicated in form to bank card for which its number is indicated.

continues on next page

Table 2 - continued from previous page

Form to RPI	Transfer from Form to RPI occurs when funds are transferred from payment details indicated in form to bank card for which its recurring payment id is indicated.
PAN to Form	Transfer from PAN to Form occurs when funds are transferred from one bank card for which its number is indicated to payment details indicated in form.
RPI to Form	Transfer from RPI to Form occurs when funds are transferred from one bank card for which its recurring payment id is indicated to payment details indicated in form.
Form to Form	Transfer from Form to Form occurs when funds are transferred from payment details indicated in form to another payment details indicated in form.
deposit to Form	Transfer from deposit to Form occurs when funds are transferred from Connecting Party account to payment details indicated in form.

5.11.2 Transfer Flow

- (1) Transfers between cards are initiated by Sender. Transfer from Connecting Party ([deposits](#)) are initiated by Receiver, with the same transaction flow.
- (2) To implement transfer-form request see </api/v4/transfer-form/>.
- (9) To implement final redirect see [Final Redirect](#).
- (11) To implement callback with final status handling see [Connecting Party Callbacks](#).
- (13) To implement order status see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

5.12 Payout Form

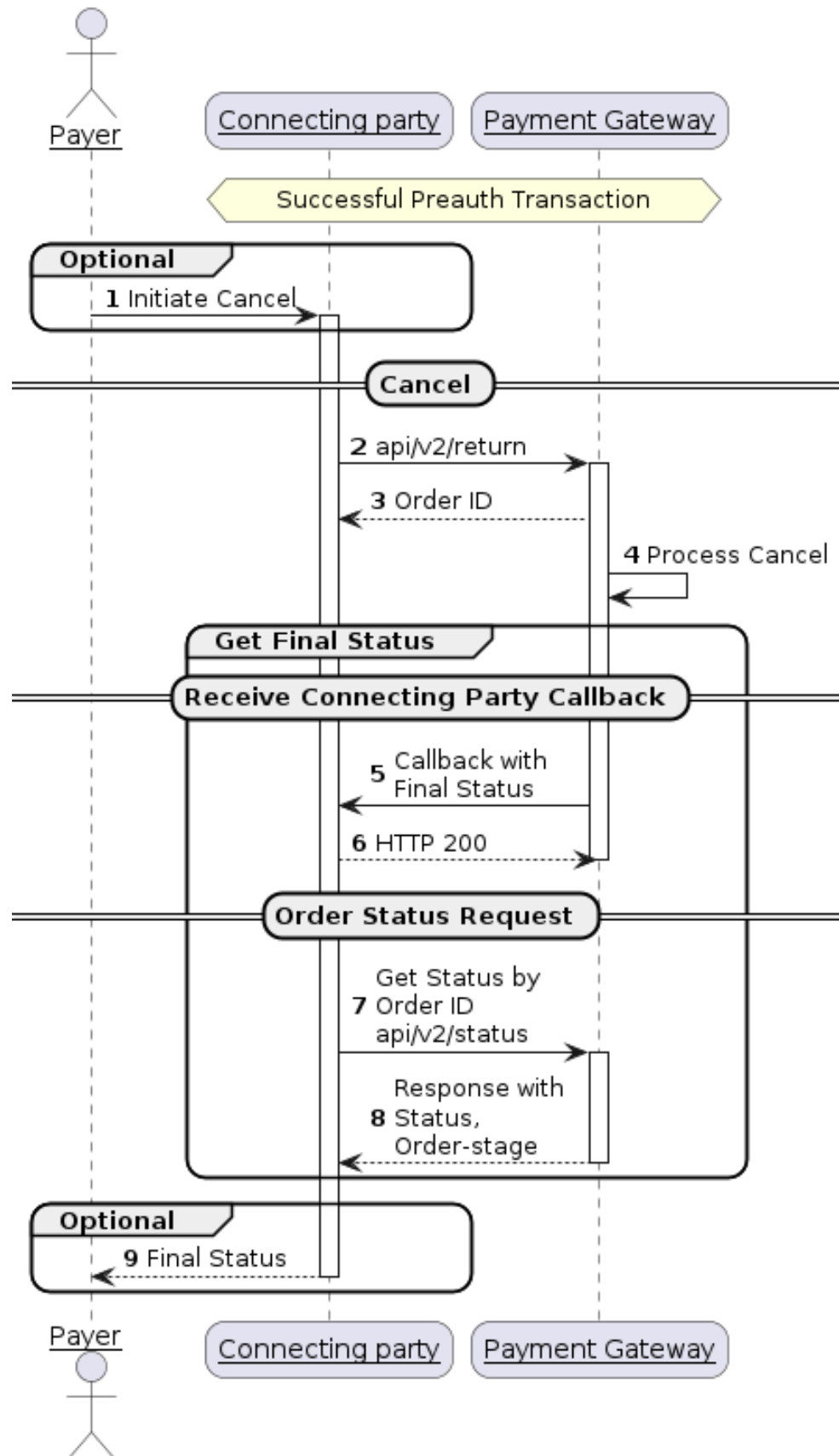
- [Introduction](#)
- [Payout Form Flow](#)

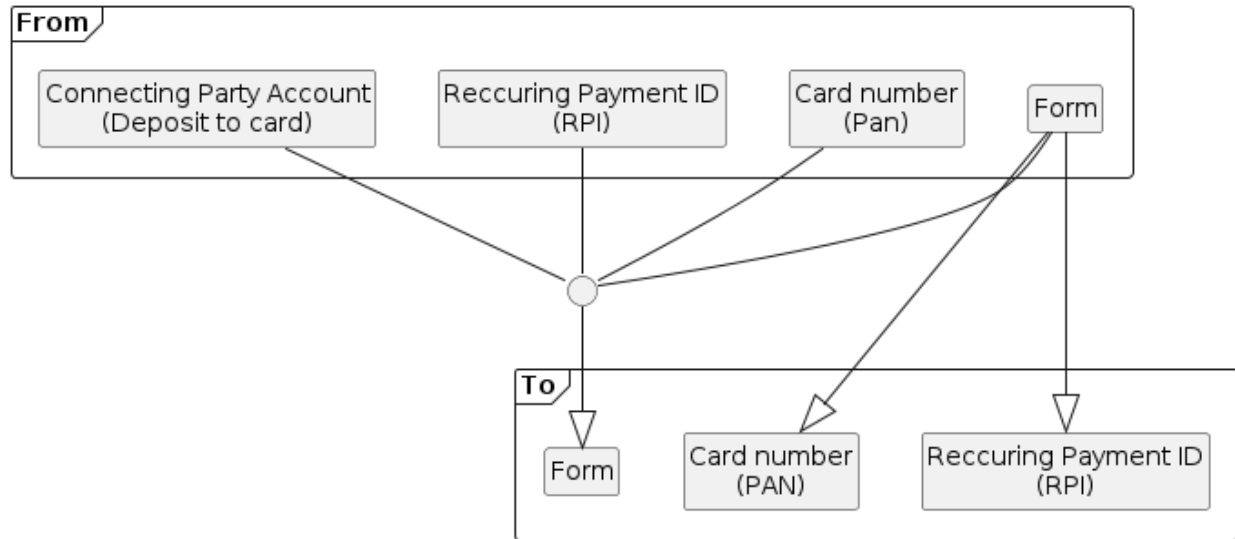
5.12.1 Introduction

Payout-form is a type of transaction which results in funds transfer from Connecting Party banking account to customer (receiver) banking account or digital wallet. Payout-form transaction in most cases is used for bank account funding. Unlike Server-to-Server Payout integration, in which Connecting Party sends receiver account in initial request, Payout Form integration allows the receiver to submit the banking account credentials on the form hosted on Doc2.0 side.

See terms definitions in [Glossary](#).

1. Receiver fill out a payout form and sends data;





Order summary

test

Total: 10.42 USD

Please fill the following

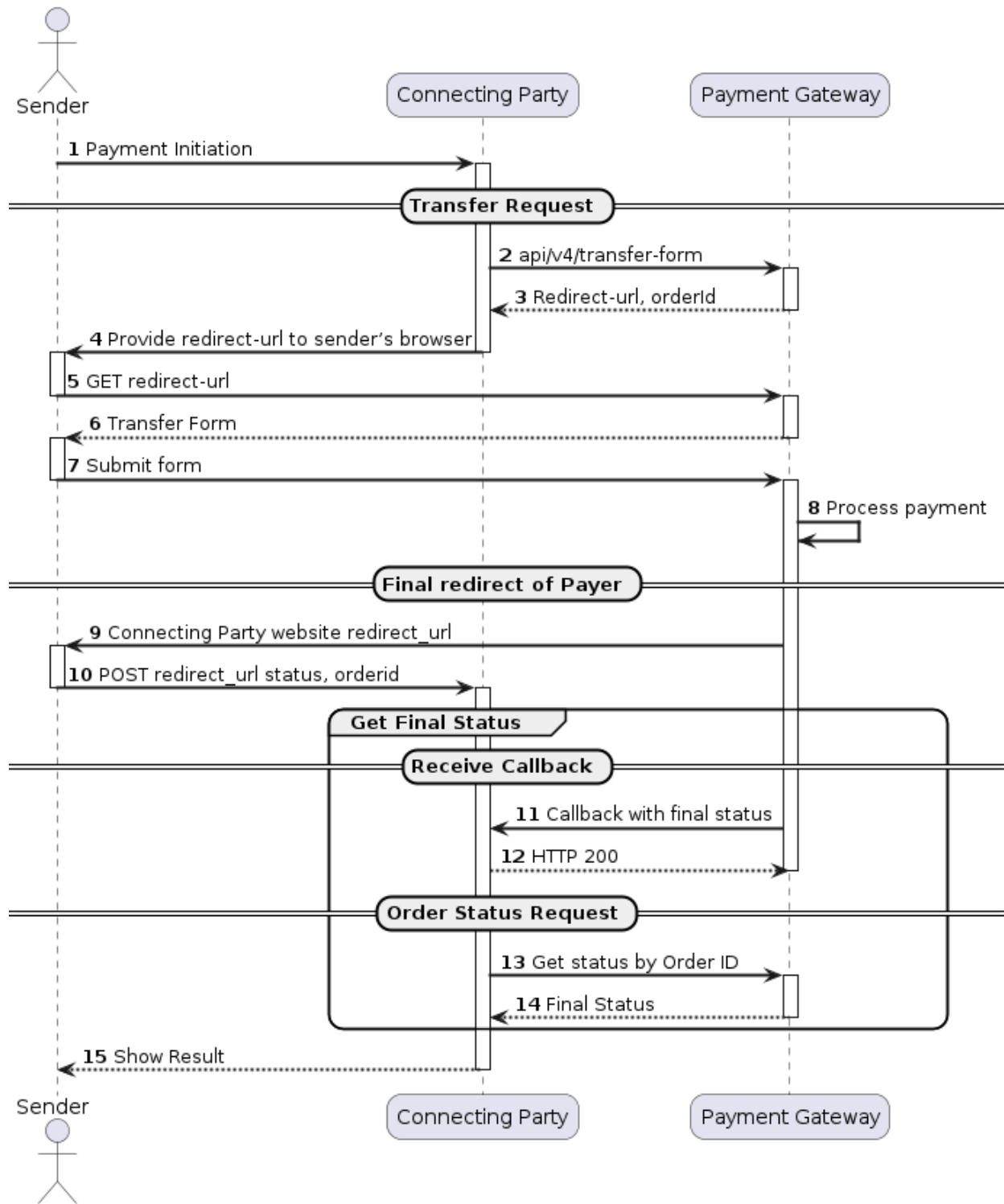
email

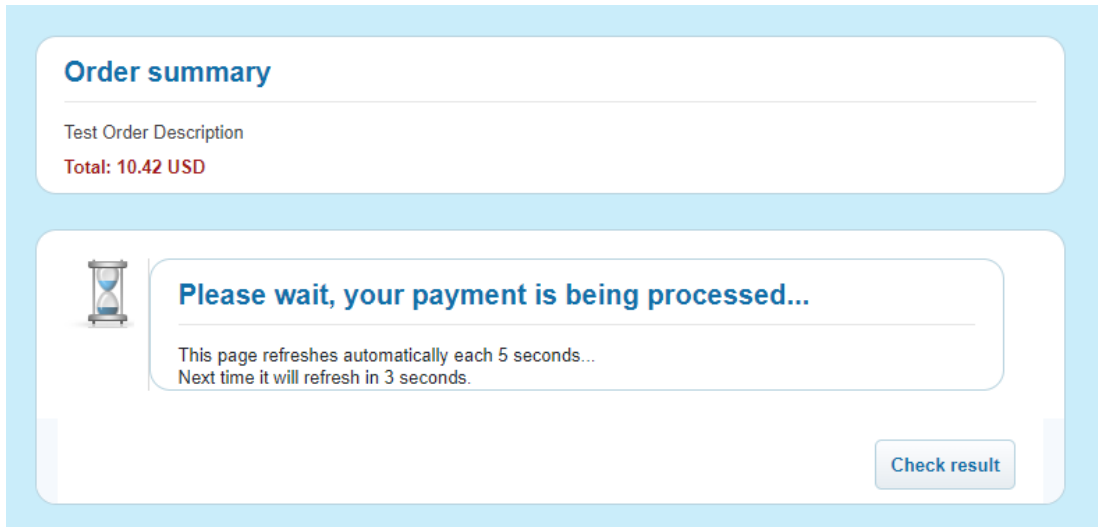
account_number

account_name

 **SECURE CONNECTION**
Process payment

2. Contextual data is gathered by Doc2.0 to process the transaction;





The screenshot displays a light blue rectangular container. Inside, at the top, is a white rounded rectangle titled "Order summary" in blue. Below the title, it shows "Test Order Description" and "Total: 10.42 USD" in red. Below this is another white rounded rectangle. On the left of this rectangle is a small hourglass icon. To its right, the text "Please wait, your payment is being processed..." is displayed in blue. Below this text, in smaller font, it says "This page refreshes automatically each 5 seconds..." and "Next time it will refresh in 3 seconds." In the bottom right corner of the light blue container is a button labeled "Check result".

- Receiver's browser gets redirected to the Connecting Party website to the resultant page.



Payment successful

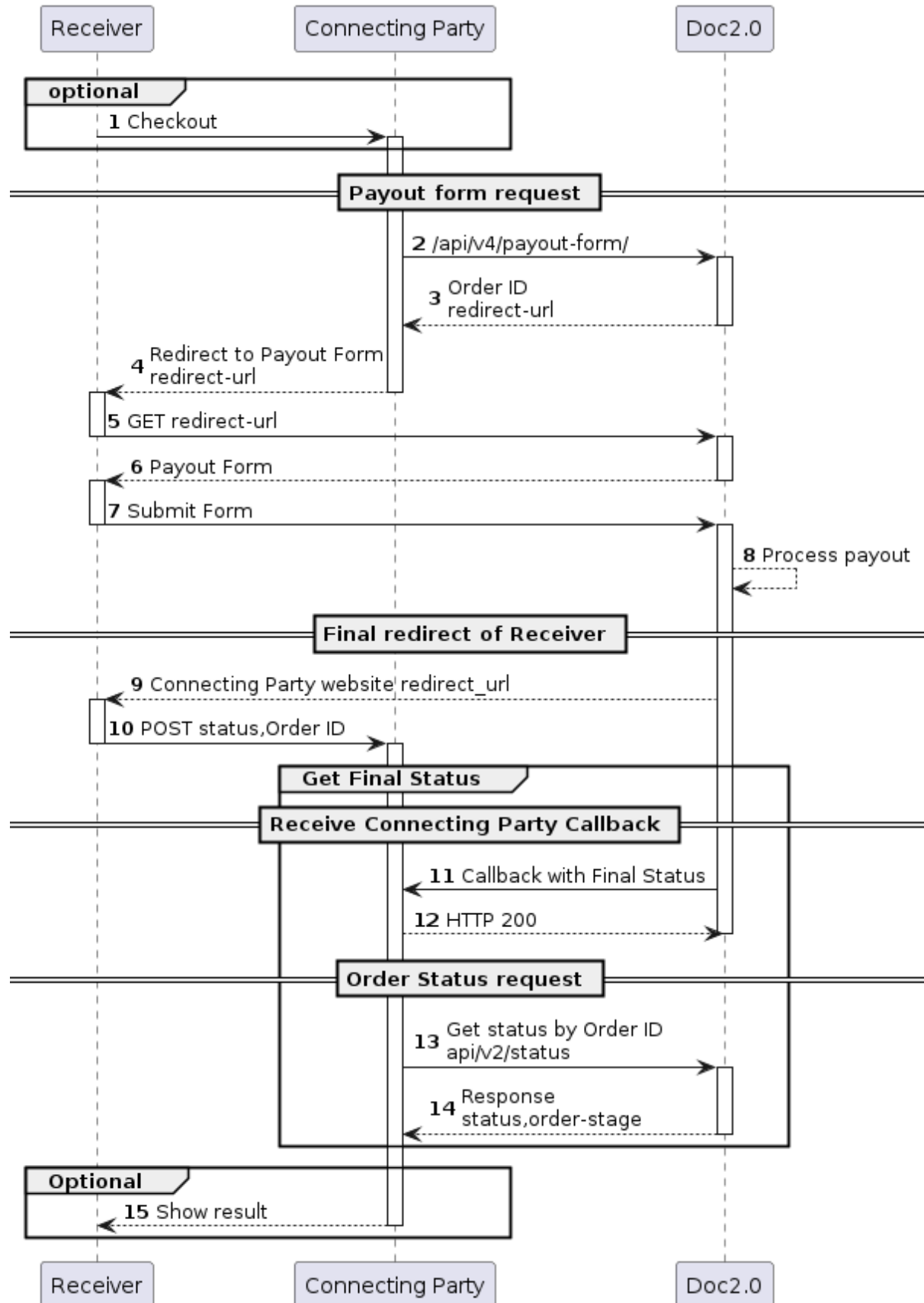
Payment type	Net banking
Bank	HDFC
Mobile	8897131444
Email	su@gmail.com
Amount paid	500.00
Transaction id	123456789

PRINT

CLOSE

5.12.2 Payout Form Flow

- (1) Payout-form can be initiated by Connecting Party based on internal business model or Receiver's request.
- (2) To implement payout-form see </api/v2/payout-form>.
- (9) To implement final redirect see [Final Redirect](#).
- (11) To implement callback with final status handling see [Connecting Party Callback](#).
- (13) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (15) Final Status can be sent by Connecting Party based on internal business model or by Receiver's request.



5.13 Payout By Reference

- [Introduction](#)
- [Payout Flow](#)

5.13.1 Introduction

Payout-by-ref is a type of transaction which results in funds transfer from Connecting Party banking account to customer (receiver) bank card, banking account or digital wallet. Payout transaction in most cases is used for bank account funding.

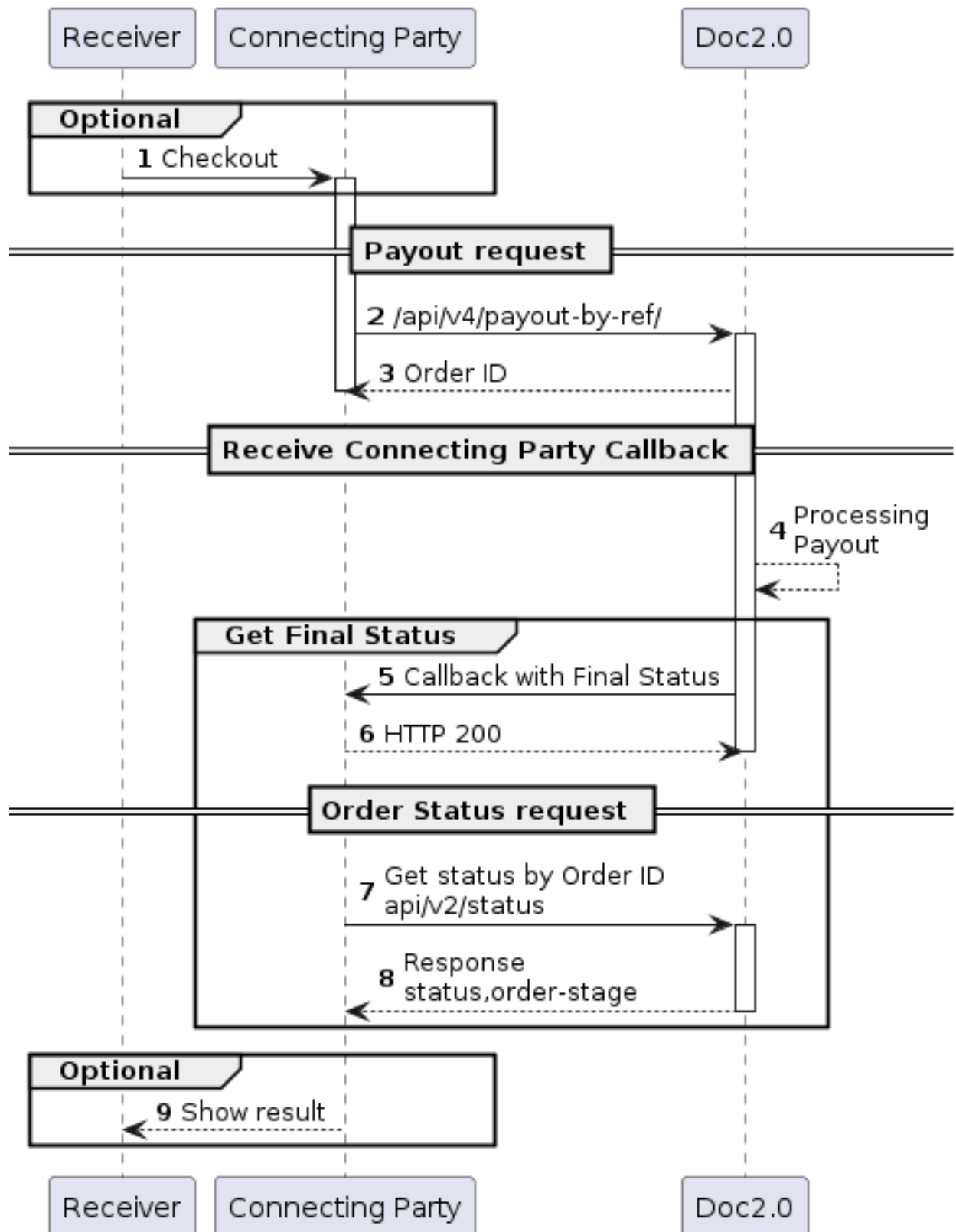
Receiver account Card Reference ID (destination-card-ref-id) for Payout By Reference can be obtained on Card registration step, see card-ref-id in [v2/create-card-ref](#) or [v4/create-card-ref](#).

Card registration requires a previously made transaction in final status, so Payout-by-ref can be initiated only for already registered Receivers.

See terms definitions in [Glossary](#).

5.13.2 Payout Flow

- (1) Payout-by-ref can be initiated by Connecting Party based on internal business model or Receiver's request.
- (2) To implement payout-by-ref transaction see [/api/v4/payout-by-ref/](#).
- (5) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (7) To implement callback with final status handling see [Connecting Party Callback](#).
- (9) Final Status can be sent by Connecting Party based on internal business model or by Receiver's request.



5.14 Payout with bank confirmation

- [Introduction](#)
- [Payout Flow](#)

5.14.1 Introduction

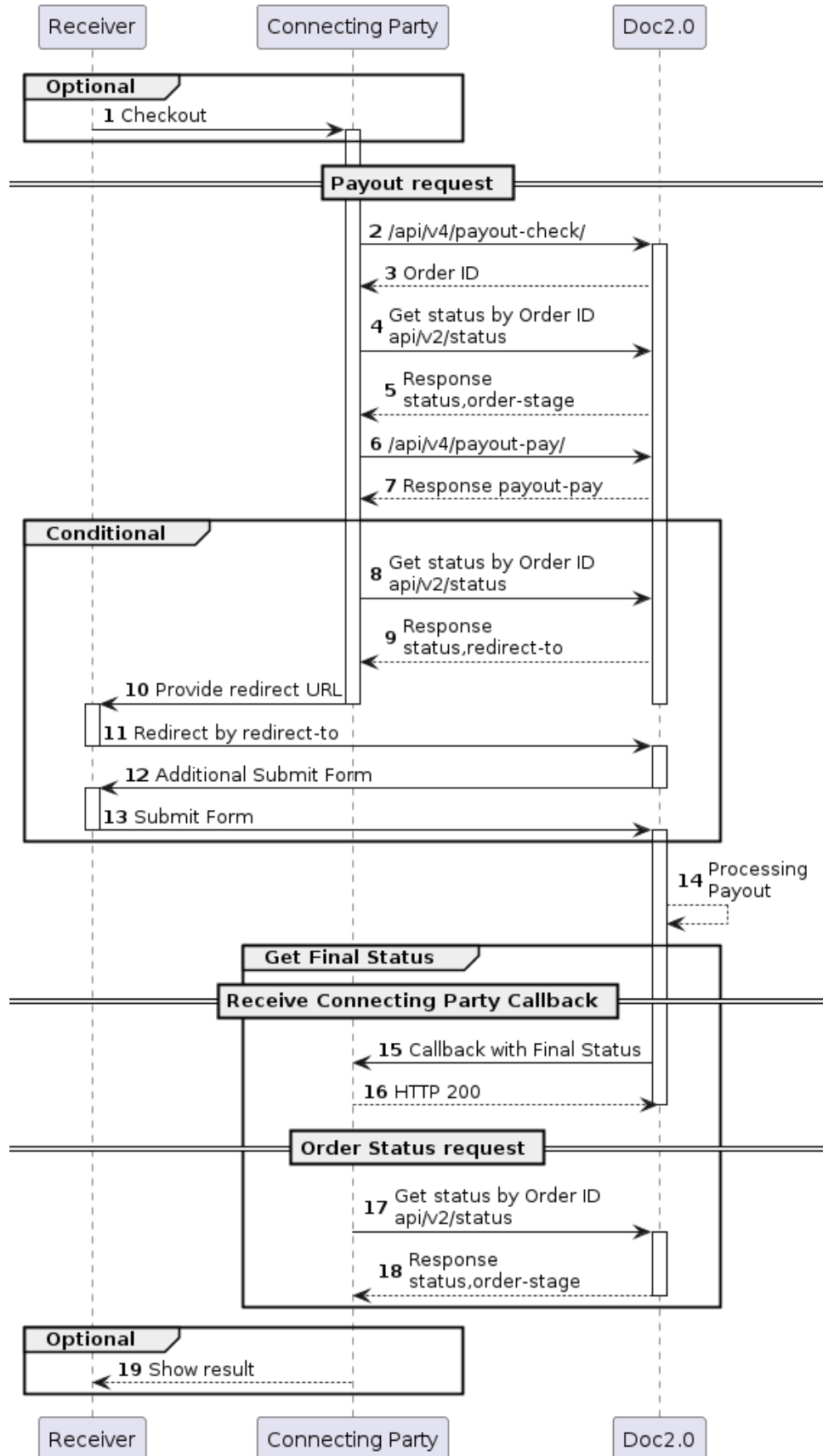
Payout is a type of transaction which results in funds transfer from Connecting Party banking account to customer (receiver) banking account or digital wallet. This Use-Case describes the integration of payouts which require additional confirmation from the bank. See terms definitions in [Glossary](#).

5.14.2 Payout Flow

- (1) Payout can be initiated by Connecting Party based on internal business model or Receiver's request.
- (2) To initiate payout, implement payout-check request, see [/api/v4/payout-check](#).
- (4) To implement order status request see [/api/v2/status/](#).
- (5) The order-stage parameter should be payout_check_validated. If payout_check_validating is received, the payout request is not confirmed by the bank yet. In this case the Connecting party should continue to request transaction status.
- (6) To continue payout processing, implement payout-pay request, see [/api/v4/payout-pay](#).
- (9) Some payout methods require the Receiver to fill the additional data on the form. The form to redirect the customer will return in status response in redirect-to parameter.
- (12) The Receiver submits the payout form.
- (13) The Receiver gets redirected back to Connecting Party. See [Final redirect](#).
- (15) To implement callback with final status handling see [Connecting Party Callback](#).
- (17) Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (19) Final Status can be sent by Connecting Party based on internal business model or by Receiver's request.

5.15 Account Verification Form

- [Introduction](#)
- [Account Verification Form Flow](#)



5.15.1 Introduction

Account verification is a non-financial transaction, which allows to validate Payer's card account information. This verification procedure helps to prevent potential fraudulent activity and meet industry-specific compliance obligations. Account Verification Form integration allows Connecting Party to exclude itself from storing, processing, or transmitting Payer's cardholder data or other sensitive payment details. Such data is submitted by Payer on Doc2.0 hosted payment form in PCI DSS certified environment.

See terms definitions in [Glossary](#).

5.15.2 Account Verification Form Flow

- (2) To implement Account Verification Form request see </api/v2/account-verification-form/>.
- (9) To implement final redirect see [Final Redirect](#).
- (11,12) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (13) To implement callback with final status handling see [Connecting Party Callbacks](#).

5.16 SBP Account Verification Form

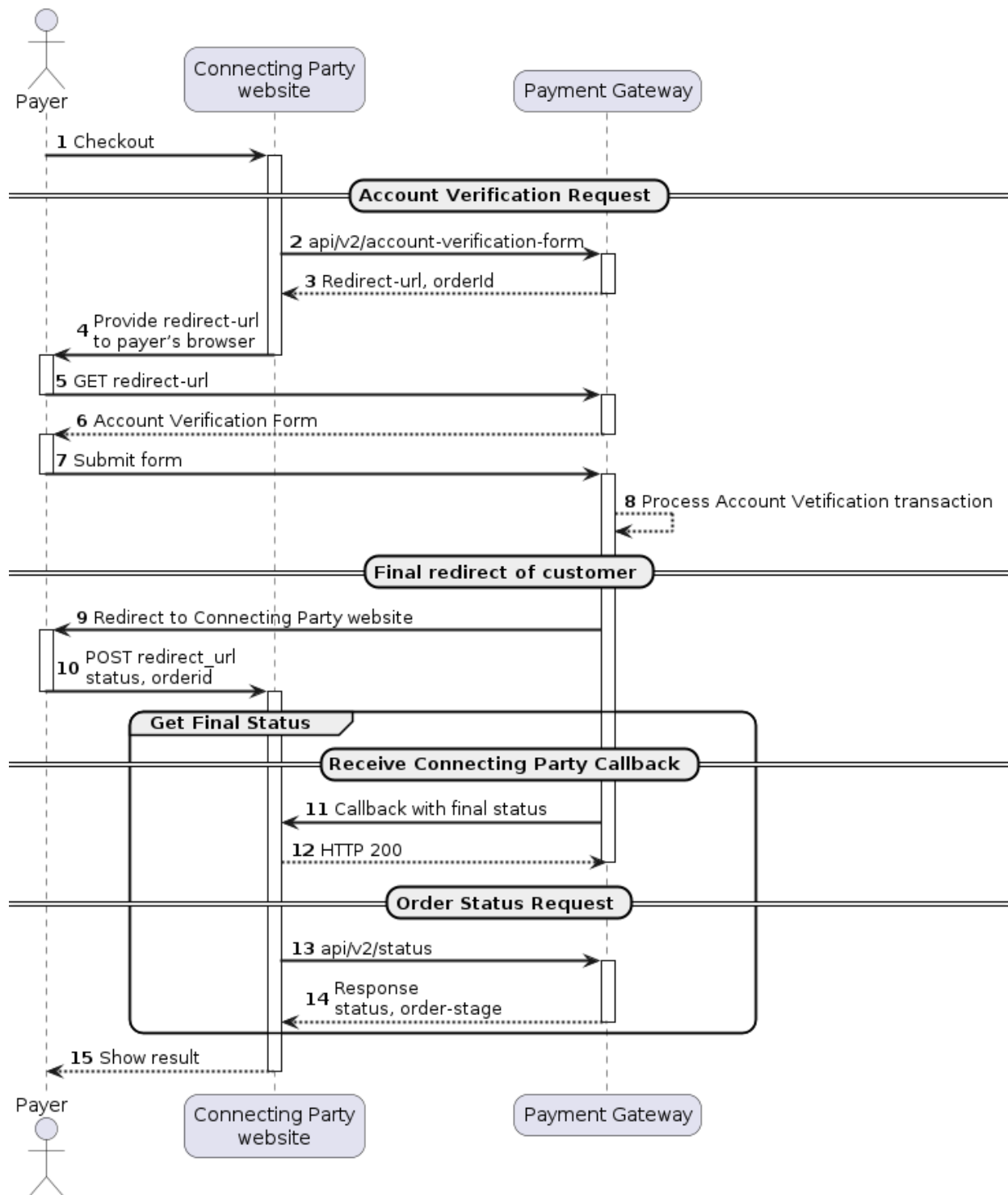
- [Introduction](#)
- [Account Verification Form Flow](#)

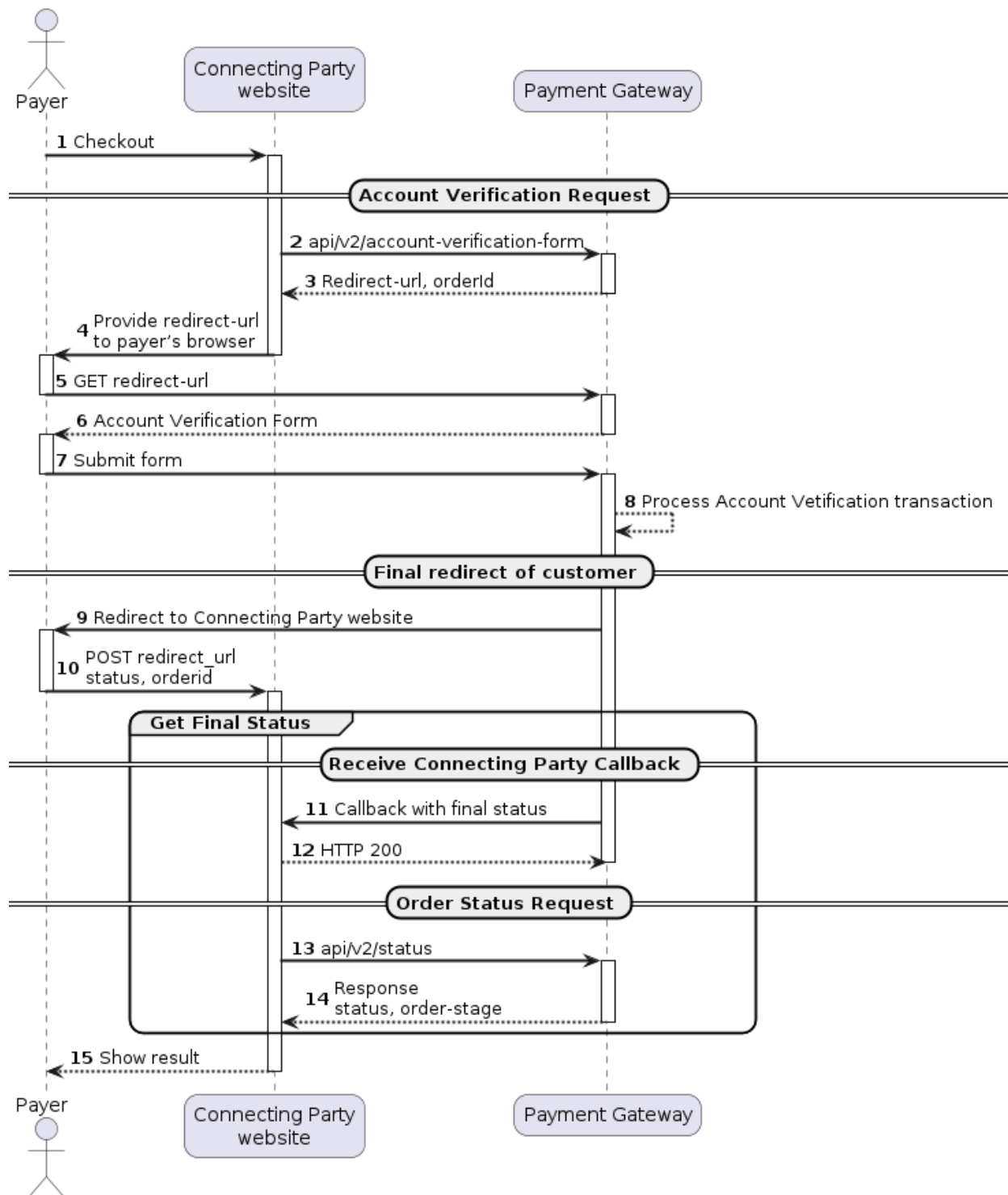
5.16.1 Introduction

Account verification is a non-financial transaction, which allows to validate and register Payer's account information. This verification procedure helps to prevent potential fraudulent activity and meet industry-specific compliance obligations, as well as to simplify future payments with this account. Account verification Form SBP integration allows Connecting Party to validate and register the Payer's account information via SBP QR Code, hosted on Payment Gateway side.

5.16.2 Account Verification Form Flow

- (2) To implement Account Verification Form request see </api/v2/account-verification-form/>.
- (9) To implement final redirect see [Final Redirect](#).
- (11,12) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (13) To implement callback with final status handling see [Connecting Party Callbacks](#).





5.17 Withdrawal by SBP

- [Introduction](#)
- [Withdrawal Flow](#)

5.17.1 Introduction

Withdrawal is a type of transaction which results in funds transfer from Connecting Party banking account to customer (receiver) banking account via faster payments system (SBP).

See terms definitions in [Glossary](#).

5.17.2 Withdrawal Flow

- (1) Withdrawal can be initiated by Connecting Party based on internal business model or Receiver's request.
- (2) To implement SBP withdrawal transaction see </api/v4/withdrawal-sbp>.
- (5) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (7) To implement callback with final status handling see [Connecting Party Callback](#).
- (9) Final Status can be sent by Connecting Party based on internal business model or by Receiver's request.

5.18 Deposit To Card Transfer

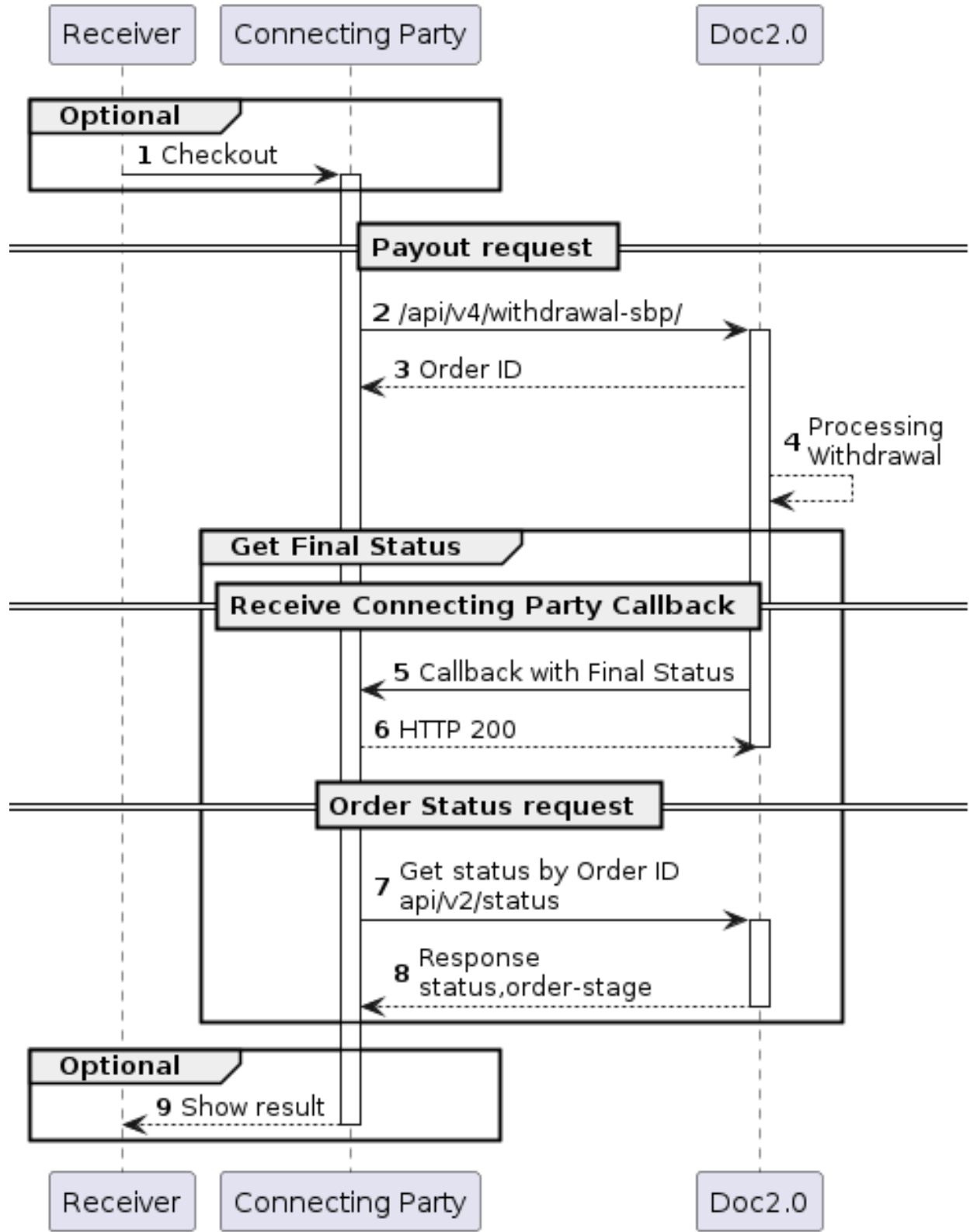
- [Introduction](#)
- [Deposit To Card Transfer Flow](#)

5.18.1 Introduction

Deposit to card transfer (D2C) is a money transfer from Connecting Party bank account (Deposit) to Receiver bank card number (PAN) or tokenized card data (Card Reference ID). Deposit to card transfer is made in three steps:

Initial payment – any transaction type with present cardholder data will work as initial payment - sale, preauth, transfer, etc. It is important to make the initial payment the most secure way in order to guarantee that a real card holder is registering the card.

Card registration – get card-ref-id (Card Reference ID) and register Customer's card in their profile. Card Reference ID can be used unlimited number of times, while the bank card is active.



Money transfer – run v4/transfer-by-ref/ API command using destination-card-ref-id obtained on previous step.

See terms definitions in [Glossary](#).

5.18.2 Deposit To Card Transfer Flow

- (1) To implement Card registration and get Card Reference ID see [/api/v2/create-card-ref](#).
- (3) Connecting Party assigns Card Reference ID to Customer's profile, after which Connecting Party shows the page saying that the card registration is successful and initial payment approved.
- (4) To implement Card information request by Card Reference ID see [/api/v2/get-card-info](#). The contents of response to this request can be used to show information about previously used card to Payer or to update Connecting Party database. This request can be made anytime if Connecting Party has Card Reference ID.
- (6) Connecting Party initiates deposit to card transfer (D2C). To implement deposit to card request see [/api/v4/transfer-by-ref](#)
- (9) To implement callback with final status handling see [Connecting Party Callback](#).
- (11) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (12) Final Status can be sent by Connecting Party based on internal business model or by Receiver's request.

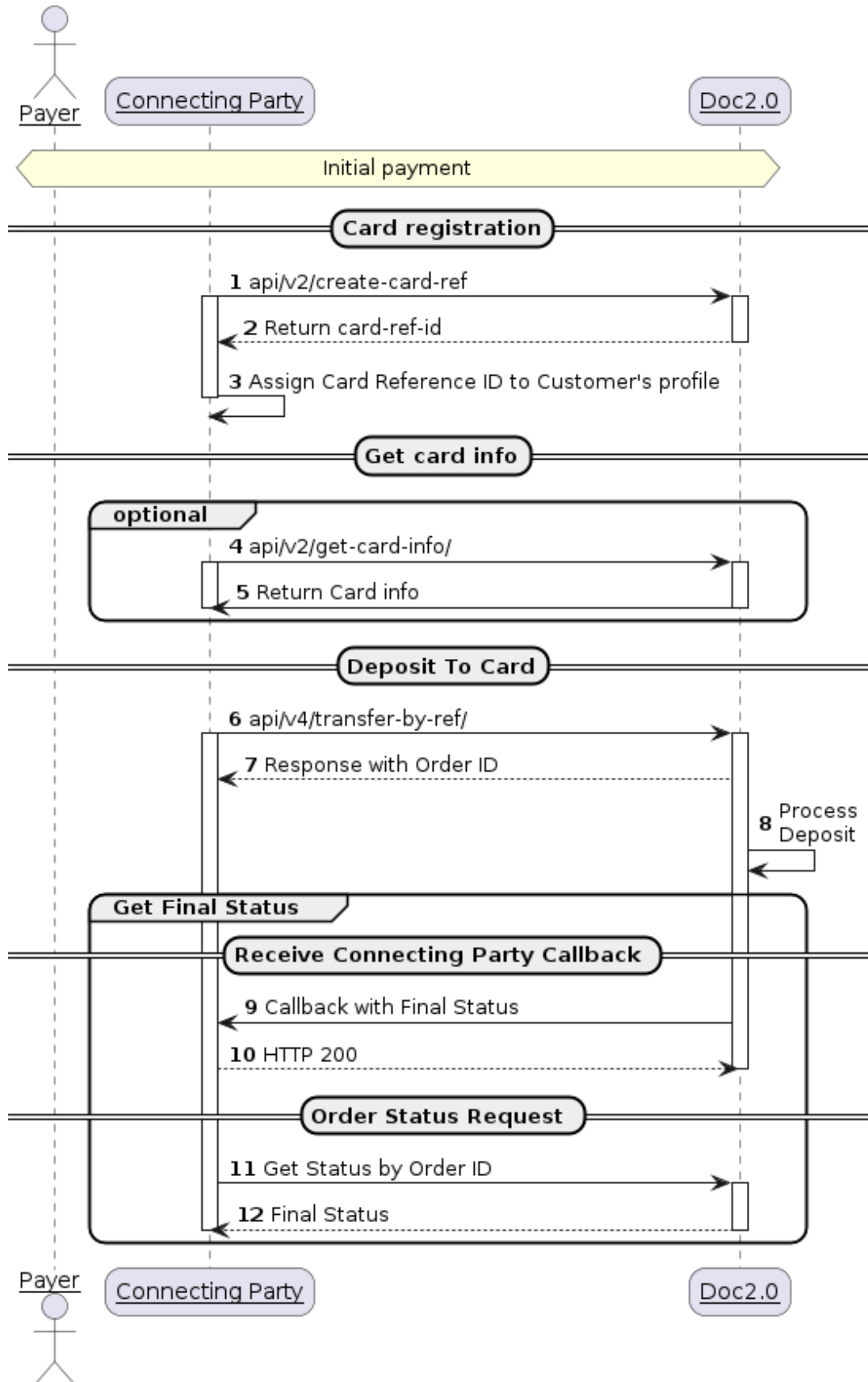
5.19 Recurring Sale

- [Introduction](#)
- [Recurring Sale Flow](#)
 - [3DS Decision Making Schema](#)
 - * [Non3D Flow](#)
 - * [Simplified Authentication Flow](#)

5.19.1 Introduction

Recurring (Rebill) Sale is a type of transaction in which funds are debited from the Payer's side to the Connecting Party's bank account with previously saved cardholder data. Recurring sale transaction can be initiated manually on Payer's behalf (this simplifies the payment process for the Payer) or automatically at certain intervals by Connecting Party (i.e. as a subscription model). Payer does not have to re-enter card information, Connecting Party uses Card reference ID to authorize payment. The necessity of Payer's 3DS check and submit of card security code (CVC, CVV, etc) for recurring sale transactions is determined by bank account settings and related to Connecting Party business type.

See terms definitions (Connecting Party, 3DS Method, etc) in [Glossary](#).



Recurring payments are made in three steps:

1. Initial payment – make initial payment to verify and authorize the Payer’s card. Any transaction type with present cardholder data will work as initial payment - sale, preauth, transfer, etc.
2. Card registration – get Card reference ID and register Payer’s card in Payer’s profile. Card registration requires initial payment to be in [final status](#).
3. Recurring sale – process recurring sale transaction using Card reference ID.

5.19.2 Recurring Sale Flow

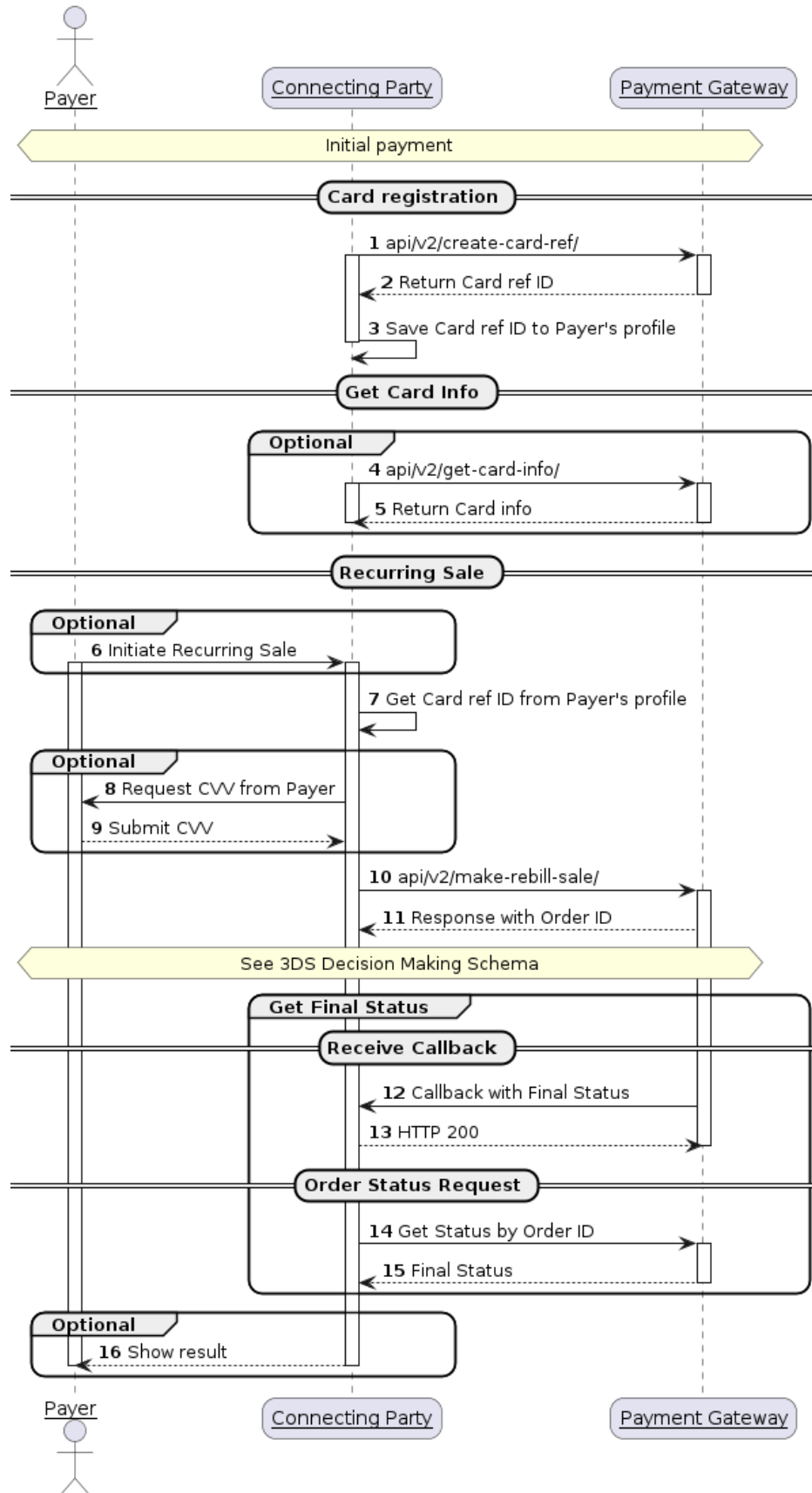
Note: Initial payment transaction type and implementation depends on Connecting Party business model. Please contact Doc2.0 support team to get relevant integration Use-Case for initial payment.

- (1) To implement Card registration and get Card reference ID see [/api/v2/create-card-ref/](#).
- (4) To implement Card information request by Card reference ID see [/api/v2/get-card-info/](#). The contents of response to this request can be used to show information about previously used card to Payer or to update Connecting Party database. This request can be made anytime if Connecting Party has Card reference ID.
- (6) Recurring payment can be initiated by Connecting Party based on internal business model or Payer’s request.
- (8, 9) Connecting Party sends request to the Payer to submit the CVV/CVC (Card Verification Value/Code), if it is required for selected processing channel.
- (10) To initiate recurring sale transaction see [/api/v2/make-rebill-sale/](#). See [3DS Decision Making Schema](#) and [3DS Implementation Scenarios](#) to correctly implement possible 3DS scenarios.
- (12) To implement callback with final status handling see [Connecting Party Callbacks](#).
- (14) To implement order status see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (16) Results of operation can be sent by Connecting Party based on internal business model or Payer’s request.

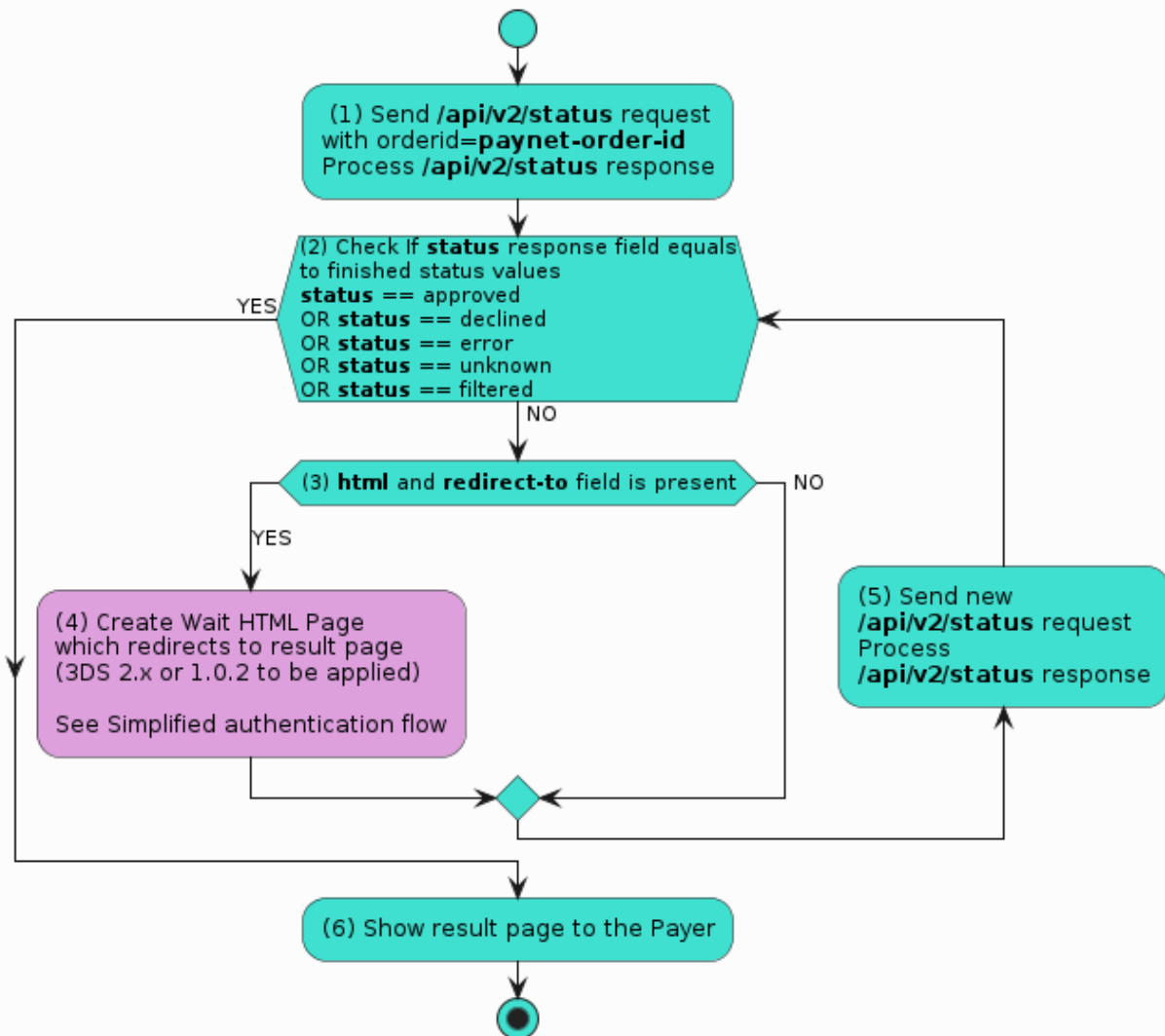
3DS Decision Making Schema

Connecting party has to implement all steps marked in green and purple. Below are the description for steps which reference specific API commands according to the step ID:

- (1) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (4) If html and redirect-to fields are present, see [Simplified authentication flow with html page](#).
- (5) The same as point (1).



3DS Decision Making Schema



Legend

Color	Implementation responsibility
	Connecting party
	Connecting and other party
	Other Party

Note: The [3DS decision making schema](#) is showcasing 3DS being initiated and performed by Payment Gateway. 3DS is initiated by Payment Gateway and performed on Connecting Party side., see [3DS Overview](#)

Non3D Flow

Sale transaction should be considered as non3D (no 3DS authentication) if all conditions are met:

1. Steps 1-2-(5)-6 of [3DS decision making schema](#) were followed.
2. tds_status, html and redirect-to parameters were not present.
3. Transaction received final status (approved, declined, error, filtered).

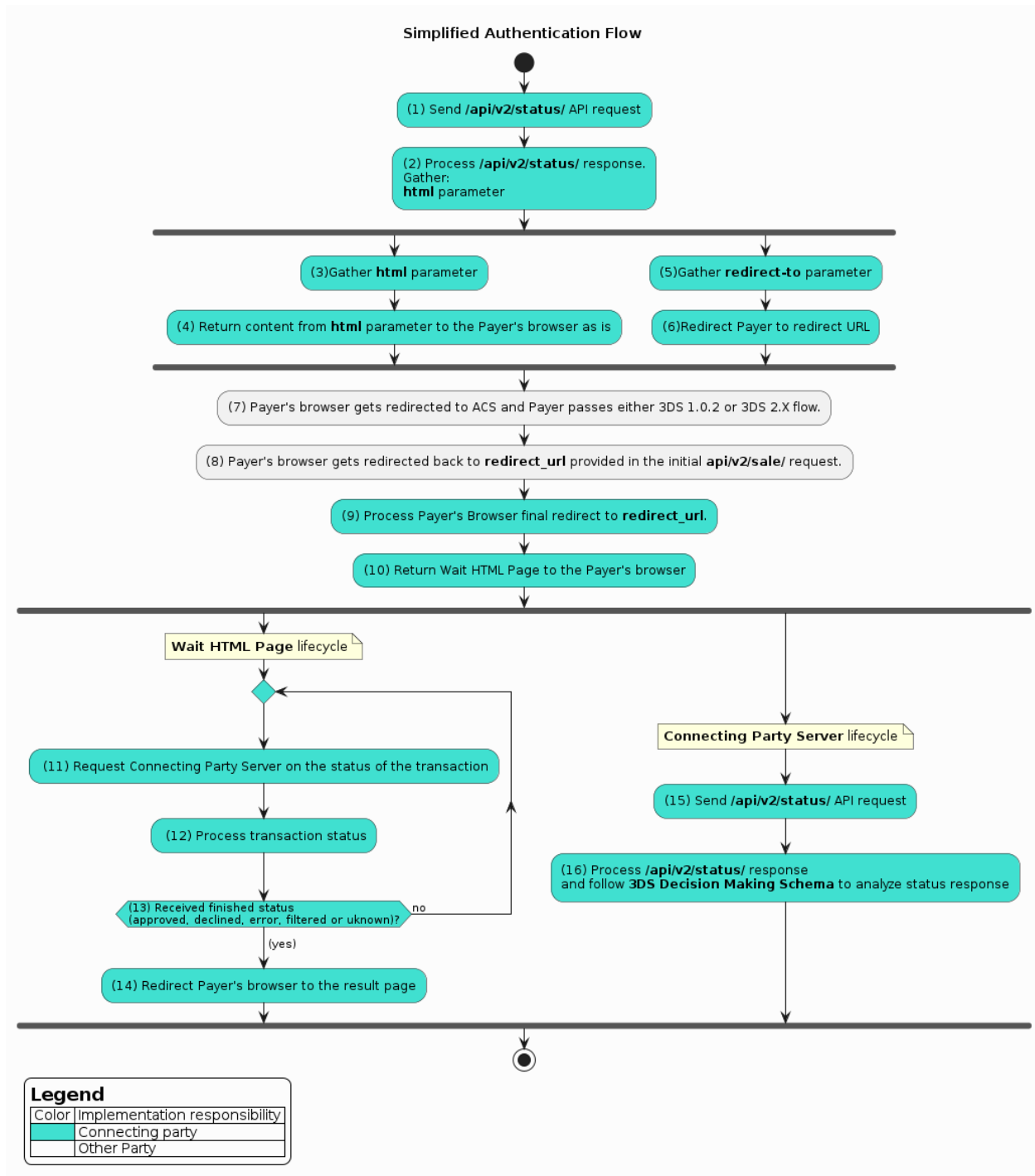
Note: Please note that transaction status “unknown” might appear for both 3DS and non3D transactions. See details in [Statuses](#).

Simplified Authentication Flow

- (1) and (2) To implement order status request see </api/v2/status/>.
- (9) To implement final redirect see [Final redirect](#).
- (10) The HTML wait page on Connecting Party side can have custom design and should communicate with Connecting Party server as described on the diagram.
- (15) and (16) The same as point (1) and (2).

5.20 Recurring Preauth, Capture and Cancel

- [Introduction](#)
- [Recurring Preauth Flow](#)
- [3DS Decision Making Schema](#)
 - [Non3D Flow](#)
 - [Simplified Authentication Flow](#)
- [Capture Flow](#)
- [Cancel Flow](#)



5.20.1 Introduction

Preauthorization is a transaction type in which bank blocks the specified amount in the Payer's card account and does not allow the cardholder to use this blocked money. It is important to know that the block remains for a definite period of time depending on whether this is a debit or a credit card (usually the maximum block period is 7 days for debit cards and 28 days for credit cards).

See terms definitions (Connecting Party, 3DS Method, etc) in [Glossary](#).

Recurring (Rebill) Preauth is a type of transaction in which funds are debited from the Payer's side to the Connecting Party's bank account with previously saved cardholder data. Recurring preauth transaction can be initiated manually on Payer's behalf (this simplifies the payment process for the Payer) or automatically at certain intervals by Connecting Party (i.e. as a subscription model). Payer does not have to re-enter card information, Connecting Party uses Card reference ID to authorize payment. The necessity of Payer's 3DS check and submit of card security code (CVC, CVV, etc) for recurring preauth transactions is determined by bank account settings and related to Connecting Party business type.

[Capture](#) is a transaction followed after preauth which deducts the locked amount from Payer's card.

[Cancel](#) is opposite of Capture which cancels the deduction and returns locked amount back to Payer's card.

Recurring payments are made in three steps:

1. Initial payment – make initial payment to verify and authorize the Payer's card. Any transaction type with present cardholder data will work as initial payment - sale, preauth, transfer, etc.
2. Card registration – get Card reference ID and register Payer's card in Payer's profile. Card registration requires initial payment to be in [final status](#).
3. Recurring preauth – process recurring preauth transaction using Card reference ID.

5.20.2 Recurring Preauth Flow

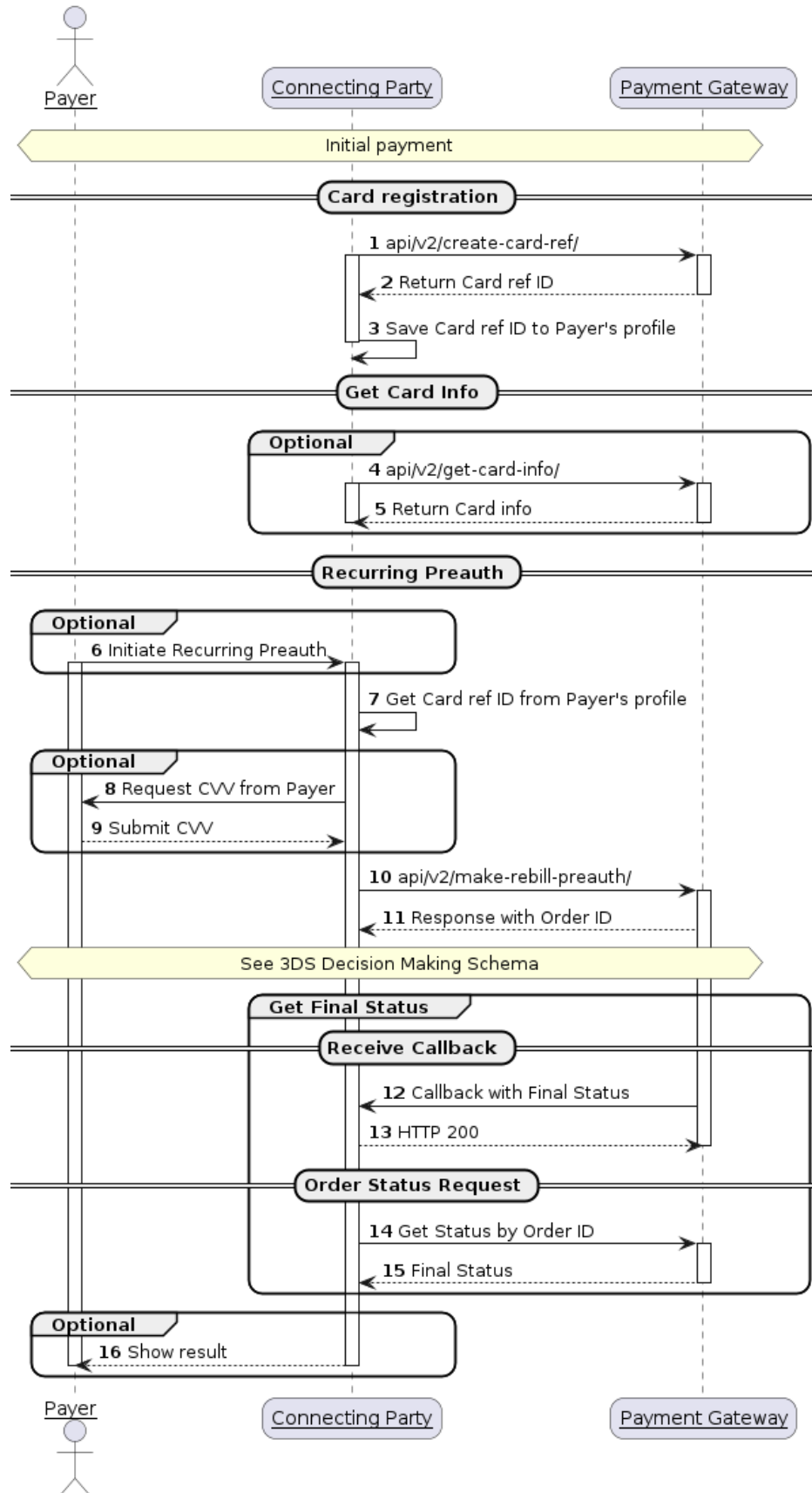
Note: Initial payment transaction type and implementation depends on Connecting Party business model. Please contact Doc2.0 support team to get relevant integration Use-Case for initial payment.

(1) To implement Card registration and get Card reference ID see [/api/v2/create-card-ref/](#).

(4) To implement Card information request by Card reference ID see [/api/v2/get-card-info/](#). The contents of response to this request can be used to show information about previously used card to Payer or to update Connecting Party database. This request can be made anytime if Connecting Party has Card reference ID.

(6) Recurring payment can be initiated by Connecting Party based on internal business model or Payer's request.

(8, 9) Connecting Party sends request to the Payer to submit the CVV/CVC (Card Verification



Value/Code), if it is required for selected processing channel.

(10) To initiate recurring preauth transaction see </api/v2/make-rebill-preauth/>. See [3DS Decision Making Schema](#) and [3DS Implementation Scenarios](#) to correctly implement possible 3DS scenarios.

(12) To implement callback with final status handling see [Connecting Party Callbacks](#).

(14) To implement order status see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

(16) Results of operation can be sent by Connecting Party based on internal business model or Payer's request.

5.20.3 3DS Decision Making Schema

Connecting party has to implement all steps marked in green and purple. Below are the description for steps which reference specific API commands according to the step ID:

(1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

(4) If html and redirect-to fields are present, see [Simplified authentication flow with html page](#).

(5) The same as point (1).

Note: The [3DS decision making schema](#) is showcasing 3DS being initiated and performed by Payment Gateway. 3DS is initiated by Payment Gateway and performed on Connecting Party side., see [3DS Overview](#)

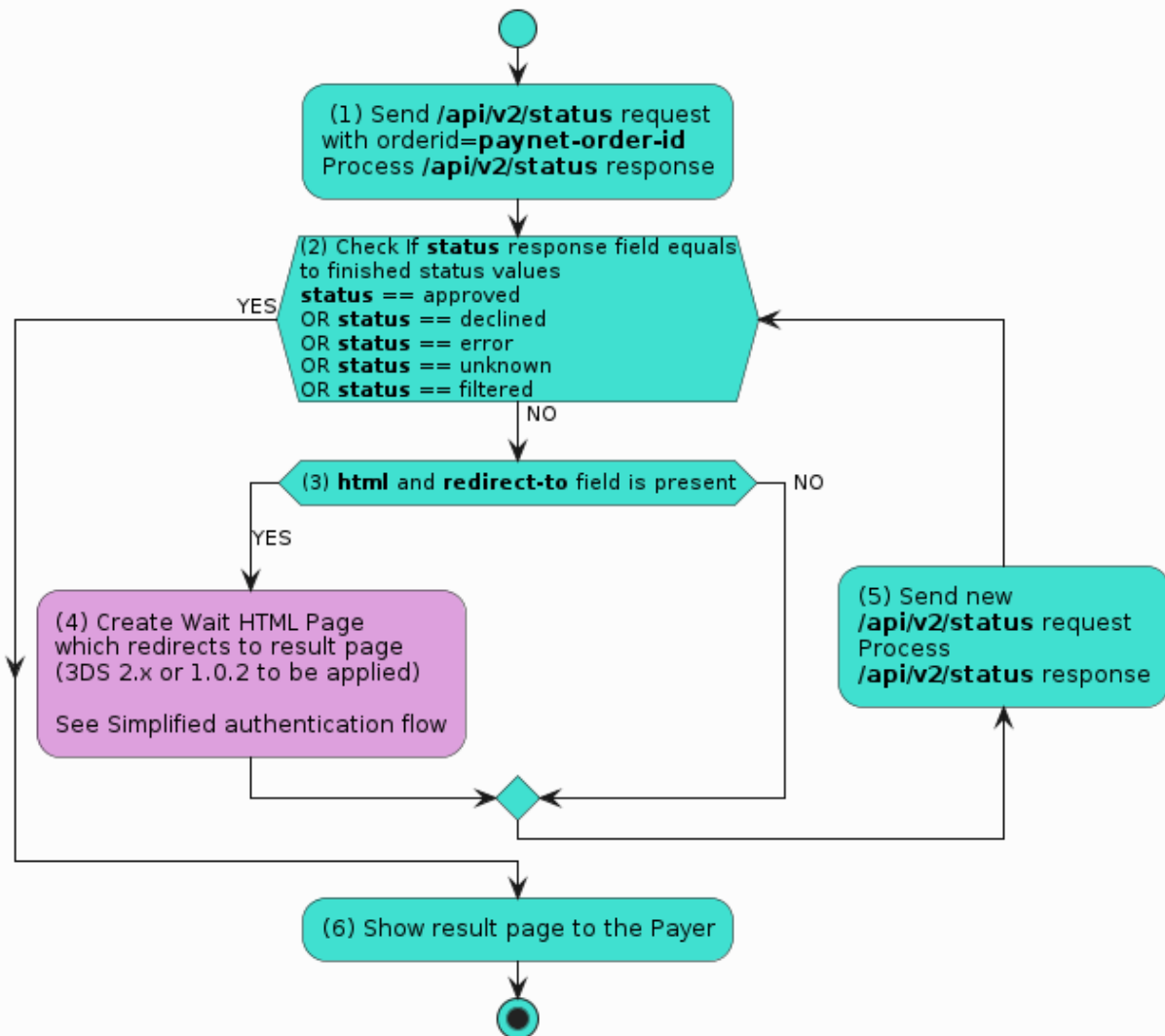
Non3D Flow

Sale transaction should be considered as non3D (no 3DS authentication) if all conditions are met:

1. Steps 1-2-(5)-6 of [3DS decision making schema](#) were followed.
2. tds_status, html and redirect-to parameters were not present.
3. Transaction received final status (approved, declined, error, filtered).

Note: Please note that transaction status "unknown" might appear for both 3DS and non3D transactions. See details in [Statuses](#).

3DS Decision Making Schema



Legend

Color	Implementation responsibility
	Connecting party
	Connecting and other party
	Other Party

Simplified Authentication Flow

- (1) and (2) To implement order status request see [/api/v2/status/](#).
- (9) To implement final redirect see [Final redirect](#).
- (10) The HTML wait page on Connecting Party side can have custom design and should communicate with Connecting Party server as described on the diagram.
- (15) and (16) The same as point (1) and (2).

5.20.4 Capture Flow

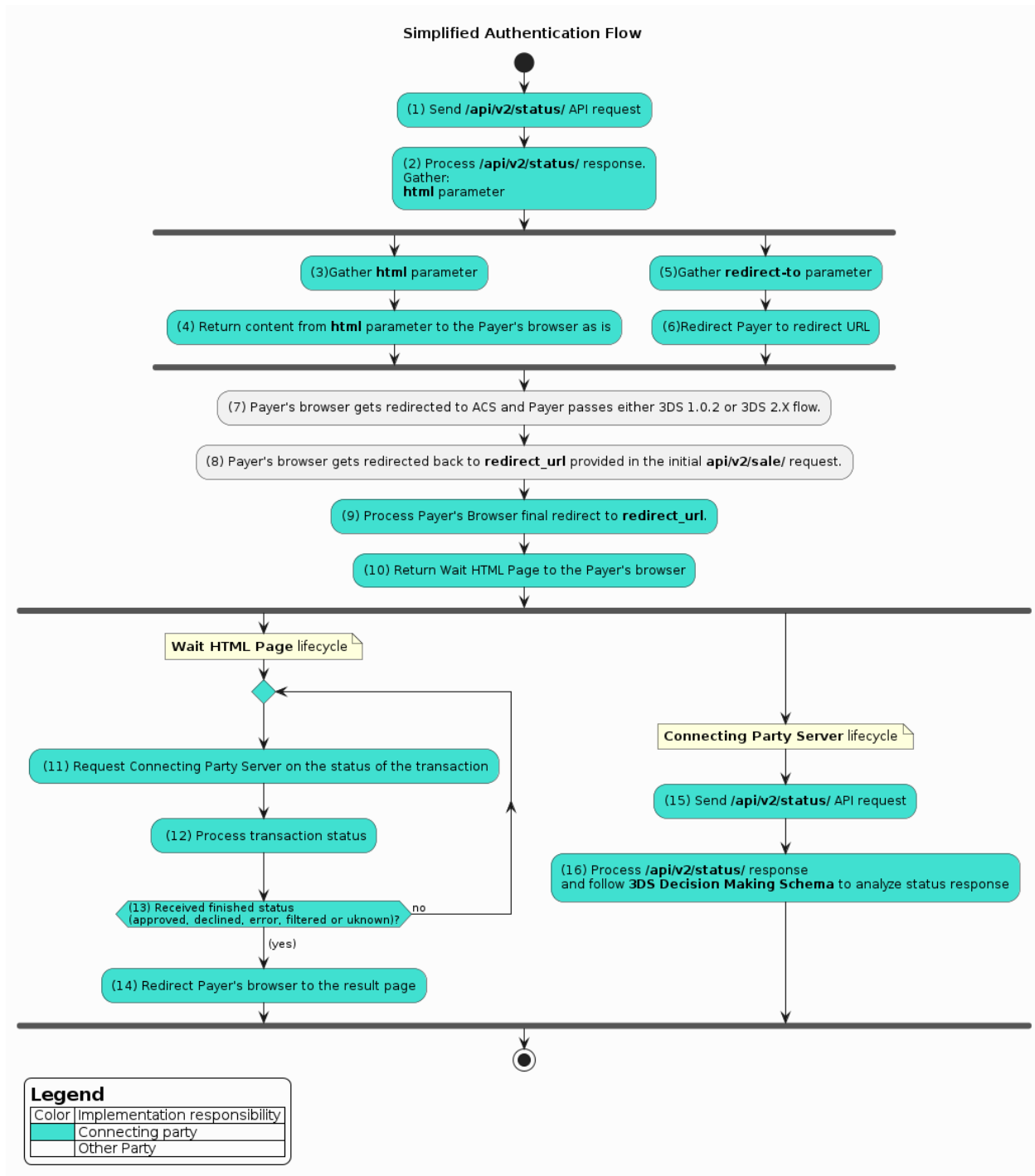
- (1) Capture can be initiated by Connecting Party based on internal business process or Payer's request.
- (2) To implement capture request see [/api/v2/capture/](#).
- (5) Callback for Capture will be sent only if notify_url was provided in initial transaction request or additional callback URL for Capture transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Capture will not be sent. To implement callback with final status handling see [Connecting Party Callback](#).
- (7) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.

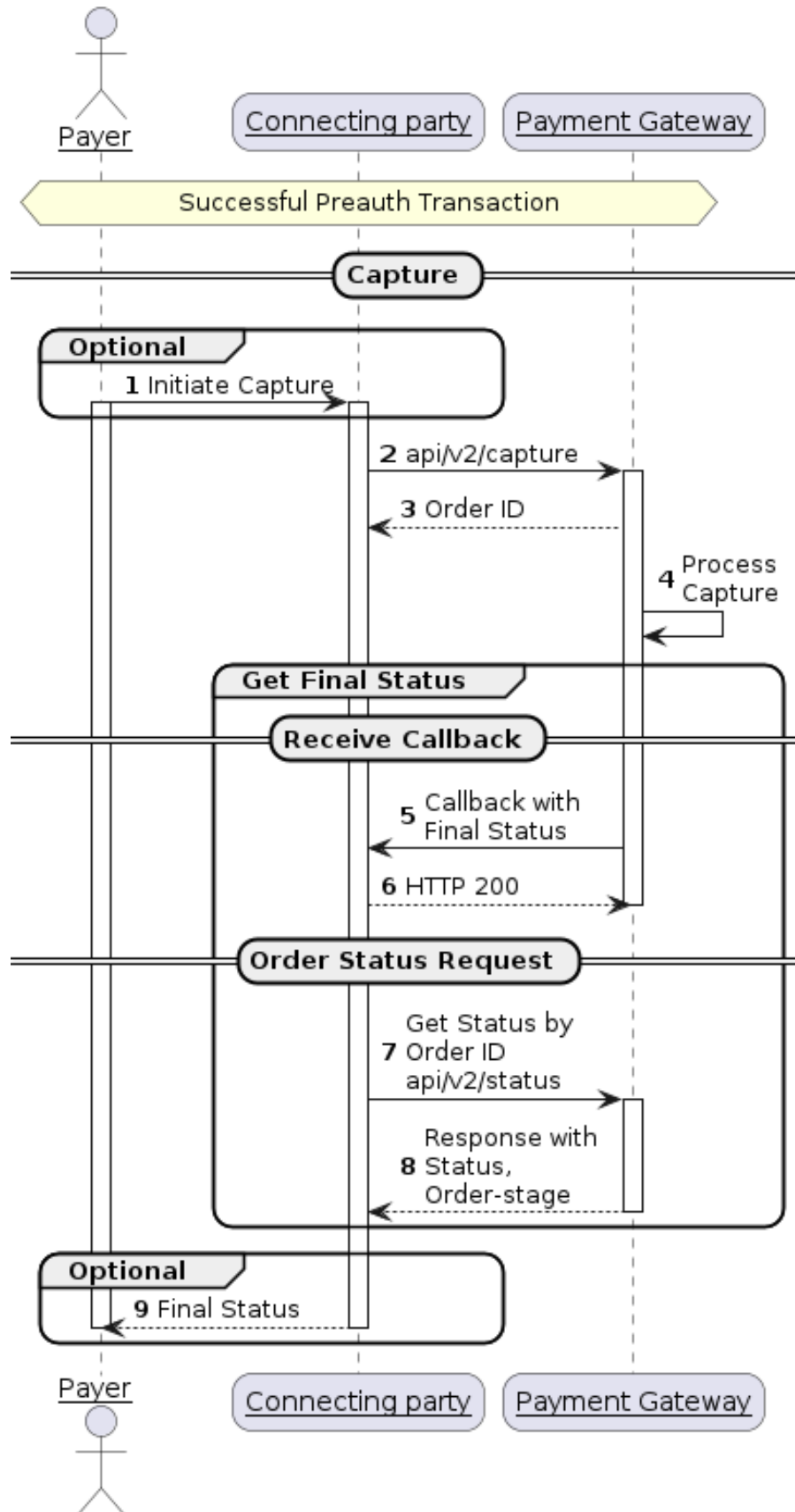
5.20.5 Cancel Flow

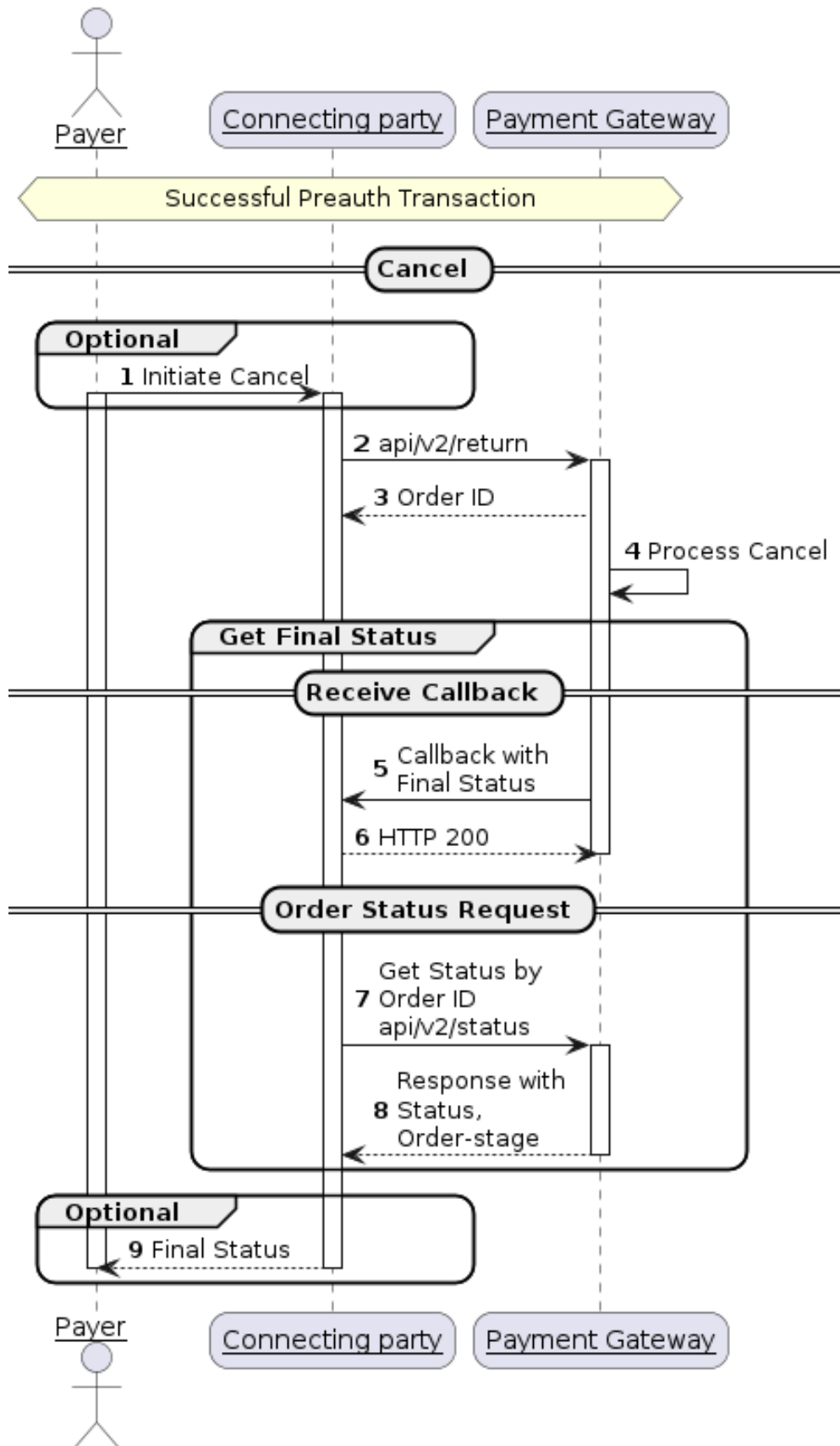
- (1) Cancel can be initiated by Connecting Party based on internal business process or Payer's request.
- (2) To implement cancel request see [/api/v2/return/](#).
- (5) Callback for Cancel will be sent only if notify_url was provided in initial transaction request or additional callback URL for Cancel transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Cancel will not be sent. To implement callback with final status handling see [Connecting Party Callback](#).
- (7) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.

5.21 Recurring Payments Service

- [Introduction](#)
- [Direct Recurring Payments Flow](#)
- [Recurring Payments With Initial Transaction Flow](#)







5.21.1 Introduction

Recurring (also Recurrent) transaction is a type of bank payment, where charges are applied to a cardholder in predetermined intervals for services or goods of an ongoing nature (memberships, subscriptions, loan payments). Prior to the charges, card must be registered in Payment Gateway. Recurring Payments Service enables automatic processing of payments on a regular schedule, such as daily, weekly, monthly, or annually. With this service, Payers can set up their payment details once, and the system will automatically charge their credit card or bank account at the specified intervals. The Connecting Party is not required to specify the recurring schedule on Payment Gateway side: the recurring payment requests can also be initiated manually according to Connecting Party internal schedule, business model or Payer's request. To view and manage recurring payments service from UI, contact support manager.

5.21.2 Direct Recurring Payments Flow

This flow implies that Connecting Party has PCI DSS certificate and is able to work with cardholder data.

(2) To create recurring payment see [/api/v4/create-recurring-payment/](#). To create recurring payments for multiple Payers in a single request with CSV data see [/api/v4/create-recurring-payments/](#).

(7, 8, 13, 14, 20, 21) To implement callback with final status handling see [Connecting Party Callback](#).

(10) To process recurring payment see [/api/v4/process-recurring-payment/](#). To process recurring payments for multiple Payers in a single request with CSV data see [/api/v4/process-recurring-payments/](#).

(16) To update recurring payment see [/api/v4/update-recurring-payment/](#). To update recurring payments for multiple Payers in a single request with CSV data see [/api/v4/update-recurring-payments/](#).

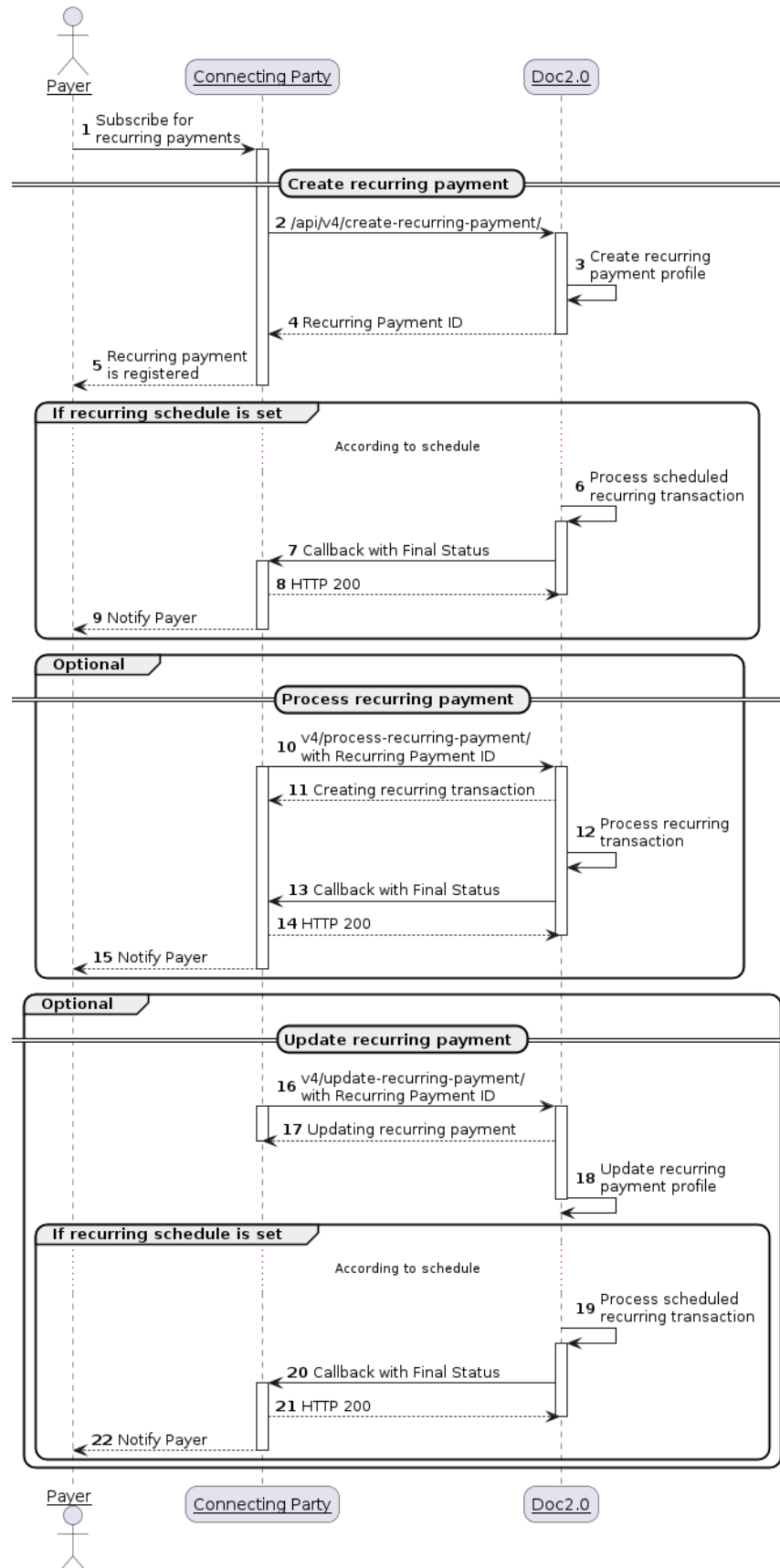
(6, 19) According to schedule means that if the schedule is set, the system will automatically process transactions at the specified intervals, update the profile and notify the Connecting Party of each completed transaction. If the schedule is not set, client can be charged only manually via UI or API call. The schedule can be changed later (see step 16).

(9, 15, 22) Payment results can be sent by Connecting Party based on internal business model or Payer's request.

5.21.3 Recurring Payments With Initial Transaction Flow

In contrast to Direct Recurring Payments Flow, in Recurring Payments With Initial Transaction Flow the initial payment must be performed to verify and authorize the Payer's card. Any transaction type that contains payment data will be suitable for this purpose: sale, preauth, transfer, etc. That transaction must be previously processed in Payment Gateway and have final status.

This flow doesn't contain cardholder data in communication between Payer and Connecting Party, as well as between Connecting Party and Payment Gateway, therefore, PCI DSS certification for Connecting Party is not required.



Discuss with support manager which API Use-Case will be most suitable for initial payment.

(2) To initiate Card registration request and get Recurring Payment ID see [api/v4/create-card-ref](#).

(9, 10, 16, 17) To implement callback with final status handling see [Connecting Party Callback](#).

(6) To process recurring payment see [/api/v4/process-recurring-payment/](#). To process recurring payments for multiple Payers in a single request with CSV data see [/api/v4/process-recurring-payments/](#).

(12) To update recurring payment see [/api/v4/update-recurring-payment/](#). To update recurring payments for multiple Payers in a single request with CSV data see [/api/v4/update-recurring-payments/](#).

(15) According to schedule means that if the schedule is set, the system will automatically process transactions at the specified intervals, update the profile and notify the Connecting Party of each completed transaction. If the schedule is not set, client can be charged only manually via UI or API call. The schedule can be changed later (see step 12).

(11, 18) Payment results can be sent by Connecting Party based on internal business model or Payer's request.

5.22 Return Transactions

- [Introduction](#)
- [Return Transaction Flow](#)
- [Void Transaction Flow](#)

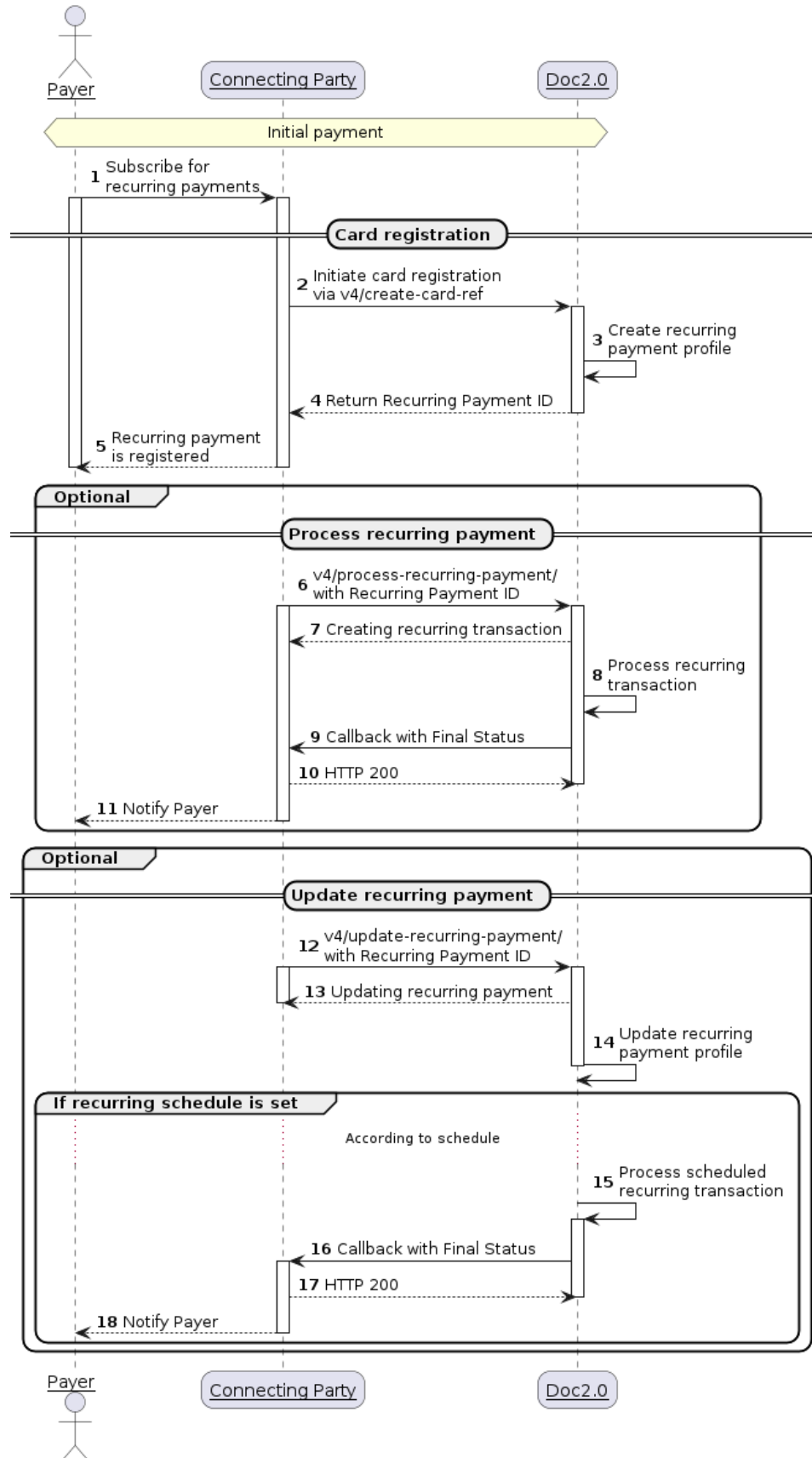
5.22.1 Introduction

Return: Return transaction means returning money back to a Payer's credit card or other type of account. Returned funds are not instantly displayed on Payer's banking account, refund processing by Acquirer can take several days after it has been initiated. Return can only be processed for Sale, Preauth and Capture transactions in the final successful status. For Preauth it makes Cancel transaction, for Capture and Sale – Reversal. See [Statuses](#).

Void: Void transaction sometimes can be used instead of Refund transaction to return money back to Payer's account. Void request excludes transaction from Acquirer's clearing list. The transaction can only be voided if it has been authorized but not settled yet. If clearing already happened or Void is not supported, it's only possible to return funds to Payer's account with Refund transaction.

Void can only be processed for Sale and Capture transactions in the final successful status, Void request makes Void transaction. See [Statuses](#). Void might not be supported in most cases, ask support manager for details.

See terms definitions in [Glossary](#).



5.22.2 Return Transaction Flow

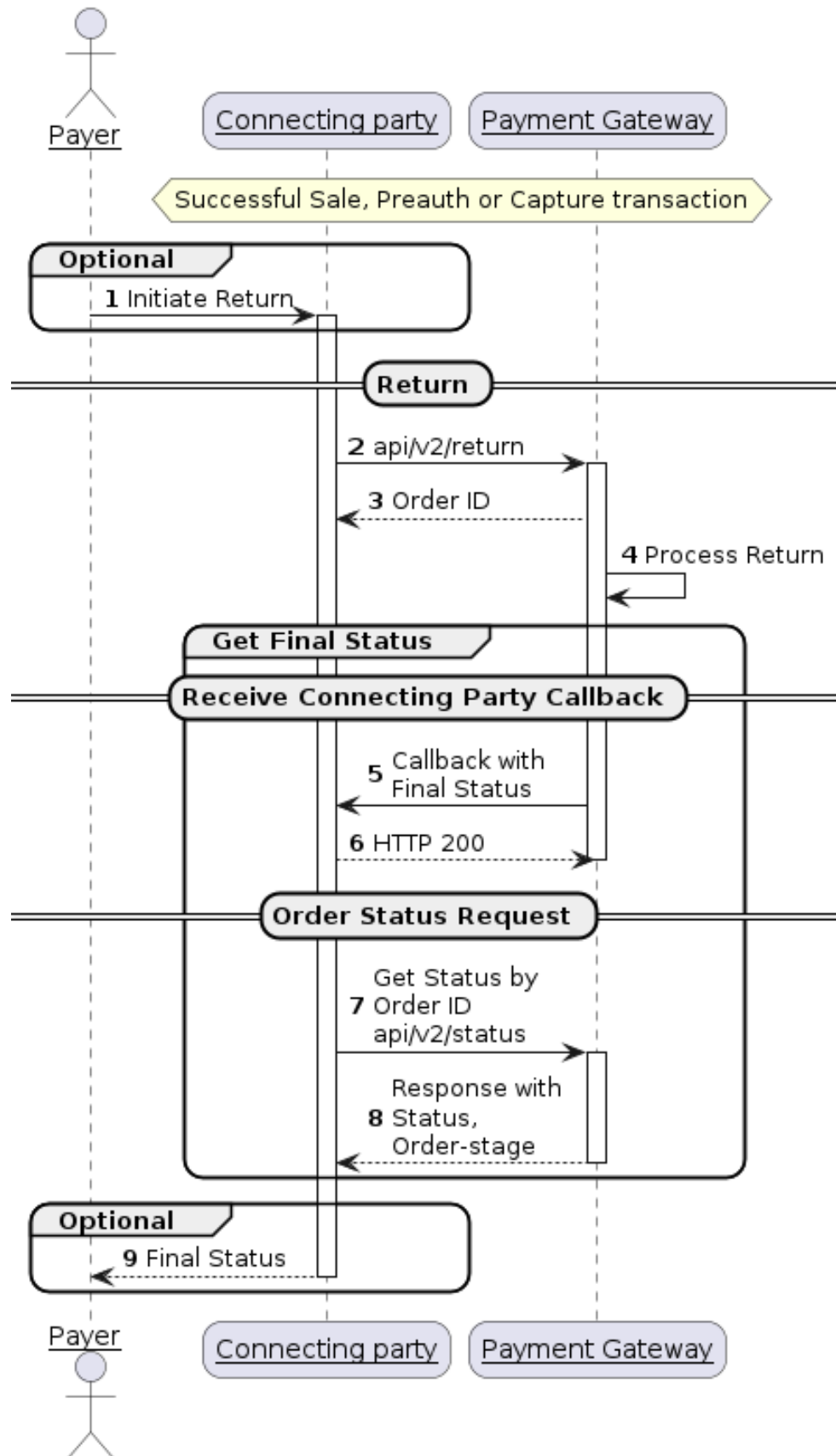
- (1) Return can be initiated by Connecting Party based on internal business process or by Payer's request.
- (2) To implement return request see </api/v2/return/>.
- (5) Callback for Return will be sent only if notify_url was provided in initial transaction request or additional callback URL for Return transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Return will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business process or Payer's request.

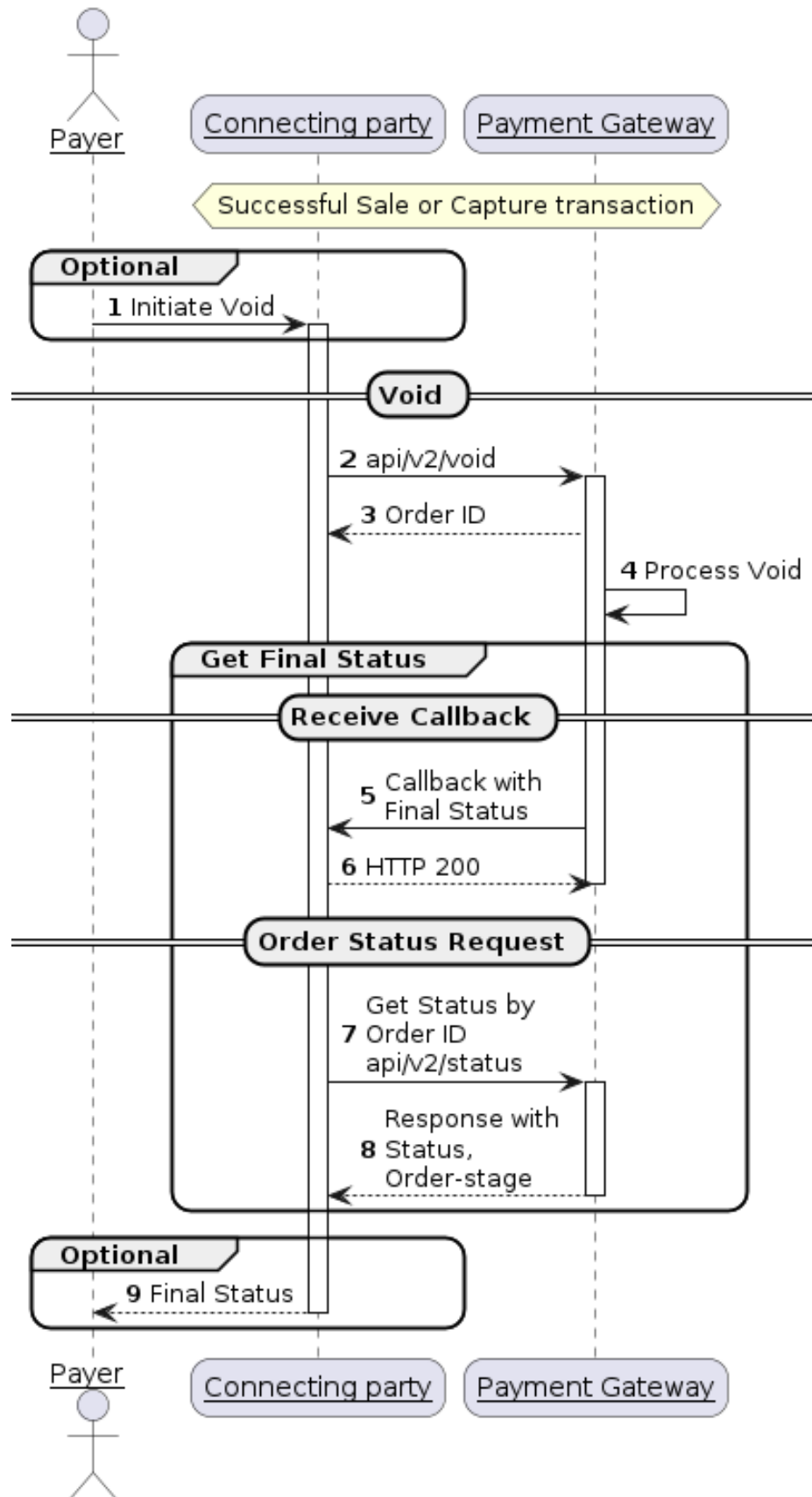
5.22.3 Void Transaction Flow

- (1) Void can be initiated by Connecting Party based on internal business process or by Payer's request.
- (2) To implement void request see </api/v2/void/>.
- (5) Callback for Void will be sent only if notify_url was provided in initial transaction request or additional callback URL for Void transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Void will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business process or Payer's request.

5.23 Payment Cashier

- [Introduction](#)
- [Payment Cashier Flow](#)
- [Options For Multi-Currency Integration](#)
- [Payment Flow Customization Scenarios](#)
 - [Callback Notification Options](#)
 - [Redirect Payer After Any Decline](#)
 - [Keep Payer on Finish Page](#)
 - [Forced Payer Redirect](#)
 - * [Option 1](#)
 - * [Option 2](#)
- [Subsequent Transactions on Payment Cashier](#)





- [Cancel Flow](#)
- [Capture Flow](#)
- [Reversal Flow](#)

5.23.1 Introduction

Payment Cashier integration allows Payer to choose the payment method for transaction. Payment Cashier can be configured on the Connecting Party side or the Payment Gateway (it is also referred to as Parallel Form) side which is covered in this Use-Case. Connecting Party redirects the Payer to Parallel Form hosted on the Payment Gateway side, the Payer selects one of the available payment methods and tries to perform the payment. Parallel Form can initiate sale or preauth transactions for each configured payment method. Once the transaction is successful, the Payer gets redirected back to the Connecting Party. For more information please see [Payment Flow Customization](#).

To see how the forms can be customized, please follow the [Forms Customization](#) reference to see examples and macros of [Payment Cashier Page Customization](#), [Payment Page Customization](#), [Prefilled Cardholder Data in Payment Page](#), [Wait Page Customization](#), and [Finish Page Customization](#).

See terms definitions in [Glossary](#).

Payment Cashier configured in the Payment Gateway has internal structure of Master Endpoints that combine regular Endpoints. Regular Endpoints connected to the Master Endpoint are called Auxiliary Endpoints. Each Auxiliary Endpoint is configured for specific payment method. When making a payment, the Payer can choose the payment method on the Parallel Form. When Payer selects the payment method, Parallel Form initiates an auxiliary transaction on the corresponding Auxiliary Endpoint. If Auxiliary Endpoint currency is different from Master Endpoint currency, the master transaction amount will be converted to respective currency of this payment method. Additionally, Master Endpoints can be consolidated in the Endpoint Group to create single logical unit for multi-currency integrations. See the [Options for multi-currency integration](#) and select the preferred option with Doc2.0 support manager.

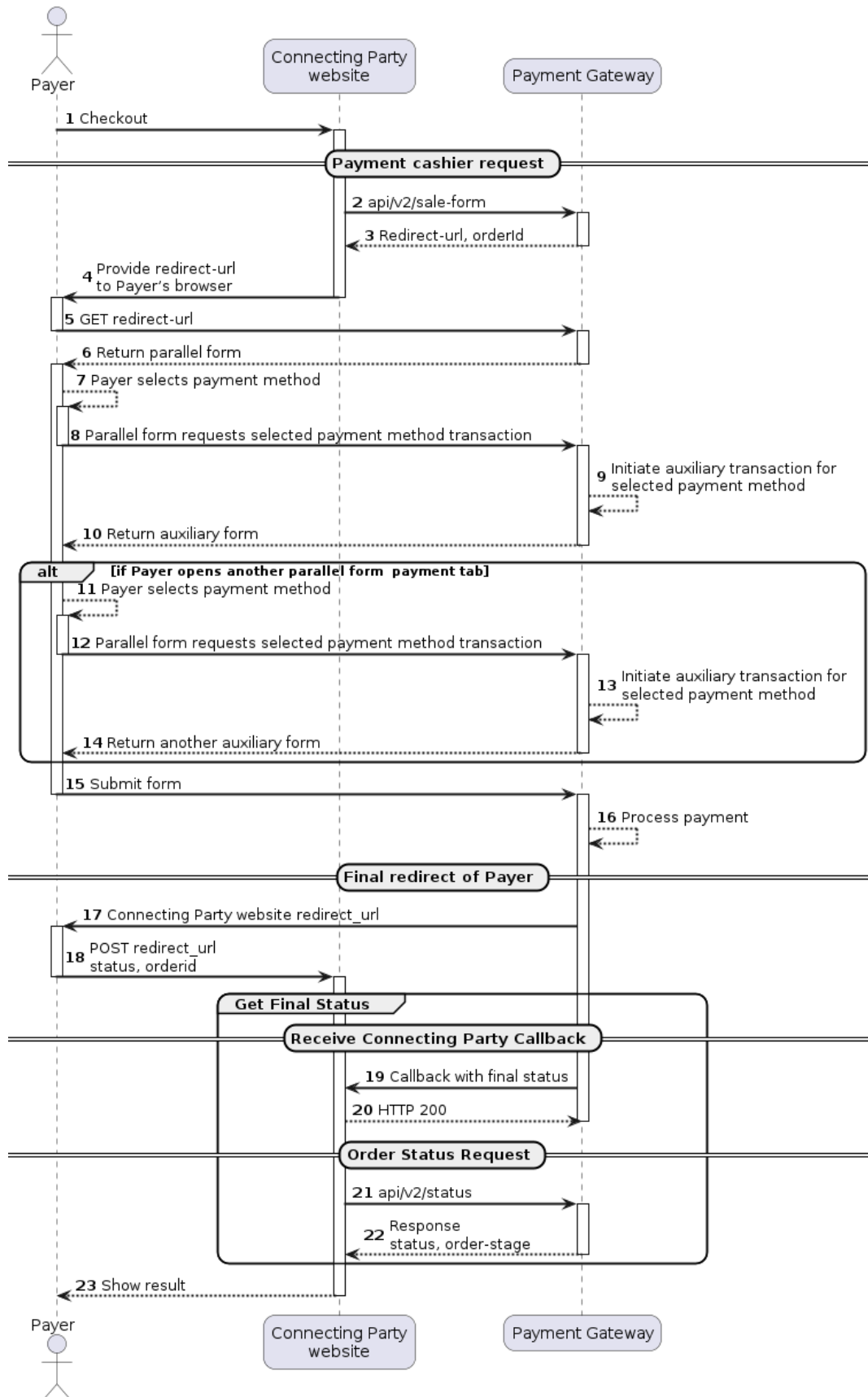
5.23.2 Payment Cashier Flow

(2) To implement a sale-form request see [/api/v2/sale-form/](#).

(14) To implement the final redirect see [Final Redirect](#).

(16,17) To implement an order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5-second intervals until final status will be received in response.

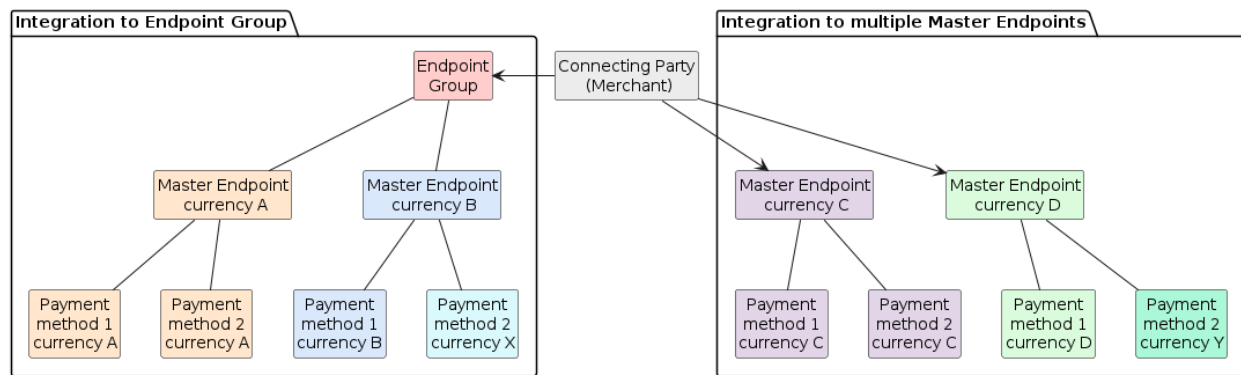
(18) To implement callback with final status handling see [Connecting Party Callbacks](#). Callback is sent with the status of the master transaction (cashier transactions). To get additional callbacks about the result of each initiated transactions, please contact Doc2.0 support manager.



5.23.3 Options For Multi-Currency Integration

Payment Cashier can be configured for multiple currencies. For each transaction currency required by Connecting Party, a separate Master Endpoint with unique ID is created. In order to avoid integration to new Master Endpoint IDs for additional currencies support in future, Connecting Party can choose integration to Endpoint Group instead. Endpoint Group is a single logical unit which consolidates Master Endpoints in different currencies. Master Endpoints in new currencies can be added to the same Endpoint Group in future.

Options for multi-currency processing integration



5.23.4 Payment Flow Customization Scenarios

Callback Notification Options

By default, callbacks have the following form:

Master Endpoint Callback: client_orderid=A&orderid=B

Auxiliary Endpoint Callback: client_orderid=A&orderid=C (if it is set on an Auxiliary Endpoint)

For correct callback handling on Connecting Party side it is important to note that the callback from the Auxiliary Endpoint may come earlier than the callback from the Master Endpoint (because final status on auxiliary transaction triggers final status on master transaction).

There is another setup that can be preferred over the default one (to enable this feature please contact Doc2.0 support manager) :

Master Endpoint Callback: client_orderid=A&orderid=B

Auxiliary Endpoint Callback: client_orderid=B&orderid=C (if it is set on an Auxiliary Endpoint)

A callback on the Auxiliary Endpoint may be required in two cases:

1. The Connecting Party wants to have complete information about all payment attempts on the Payment Cashier.
2. The Connecting Party implements preauth -> capture/cancel integration on the Payment Cashier. Capture/cancel transactions can be initiated for approved preauth transaction with request sent to Master Endpoint ID by using orderid of the auxiliary transaction. This orderid comes in a callback from an Auxiliary Endpoint when preauth transaction receives the final status.

Redirect Payer After Any Decline

By default, the Payer stays on the Parallel Form if one of the auxiliary transactions is declined, so the Payer can try to pay with another available payment method. There is an option to redirect the Payer to the Connecting Party's website after receiving an unsuccessful status (decline, filtered, error, etc.) on any payment method tab of the Cashier (auxiliary transaction decline will cause the decline on the master transaction). Please contact Doc2.0 support manager to enable this feature.

Keep Payer on Finish Page

When one of the auxiliary transactions is approved, Parallel Form redirects the Payer back to the Connecting Party's website (see [Final Redirect](#)). There is an option to keep the Payer on Finish Page instead of redirecting for approved auxiliary transaction (it might be useful if Payment Cashier is displayed in an iframe on the Connecting Party's website). Please contact Doc2.0 support manager to enable this feature.

Forced Payer Redirect

If it is necessary to redirect the Payer to the Finish Page while transaction remains in process, the following code must be added to the forms mentioned below:

Option 1

Auxiliary Finish Form Template:

```
<script>
    function backToMerchant() {
        window.parent.postMessage(
            "redirect-to-merchant",
            "https://pay.connectingparty.com"
        );
    }
</script>
</head>
<body onload="backToMerchant()">
<div class="container">
    <div class="main">
        <h2 class="summary__title">Deposit <span class="status-title">${STATUS}</span>
    </h2>
    <div class="back">
        <a class="link" href="#" onclick="backToMerchant()">Back to merchant
    </div>
</div>
```

(continues on next page)

(continued from previous page)

```

<website</a>
  </div>
</div>
</div>

```

Master Payment Form Template:

```

<script>
  window.addEventListener("message", function(event) {
    if (event.origin !== "https://pay.connectingparty.com")
      return;
    if (event.data === "redirect-to-merchant") {
      window.location.replace("https://connectingparty.com/result");
    }
  }, false);
</script>

```

Option 2

Auxiliary Finish Form Template:

```

<div class="container">
  <div class="main">
    <h2 class="summary__title">Deposit <span class="status-title">${STATUS}</span>
  </h2>
  <div class="back">
    <a class="link" target="_parent" href="${CUSTOMER_REDIRECT_URL}">Back to
  <merchant website</a>
  </div>
</div>
</div>

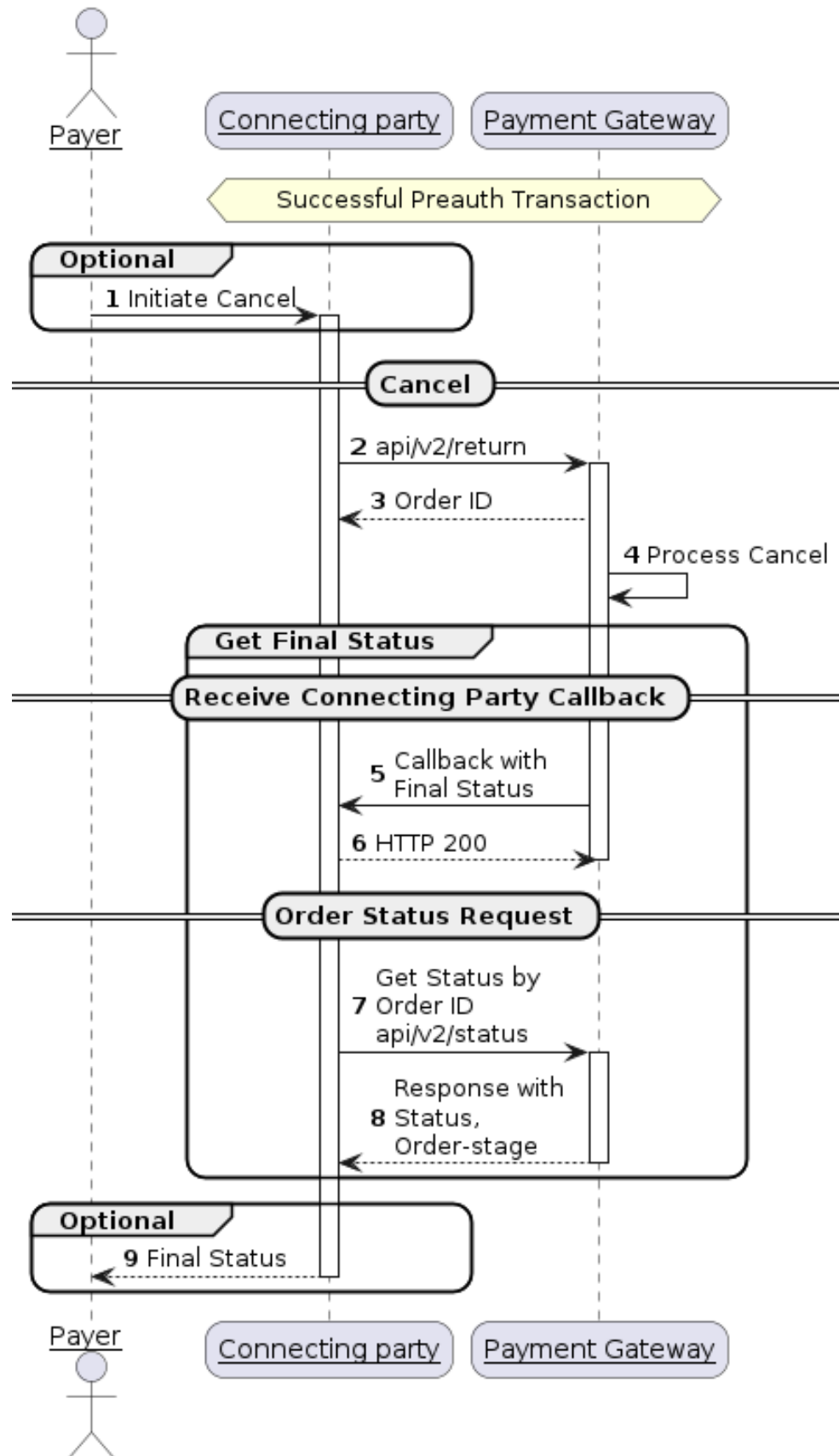
```

5.23.5 Subsequent Transactions on Payment Cashier

If the Connecting Party wants to initiate cancel, reversal or capture transactions on Payment Cashier, then Connecting Party needs to send a request to the Master Endpoint ID by using an auxiliary transaction's order_id. This value can be received in callback from Auxiliary Endpoint together with other details about this transaction. See [Callback Notification Options](#) for details and check the transaction flow below.

Cancel Flow

- (1) Cancel can be initiated by Connecting Party based on internal business model or Payer's request.
- (2) To implement cancel request see </api/v2/return/>.
- (5) Callback for Cancel will be sent only if notify_url was provided in initial transaction request or additional callback URL for Cancel transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Cancel will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).



- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.

Capture Flow

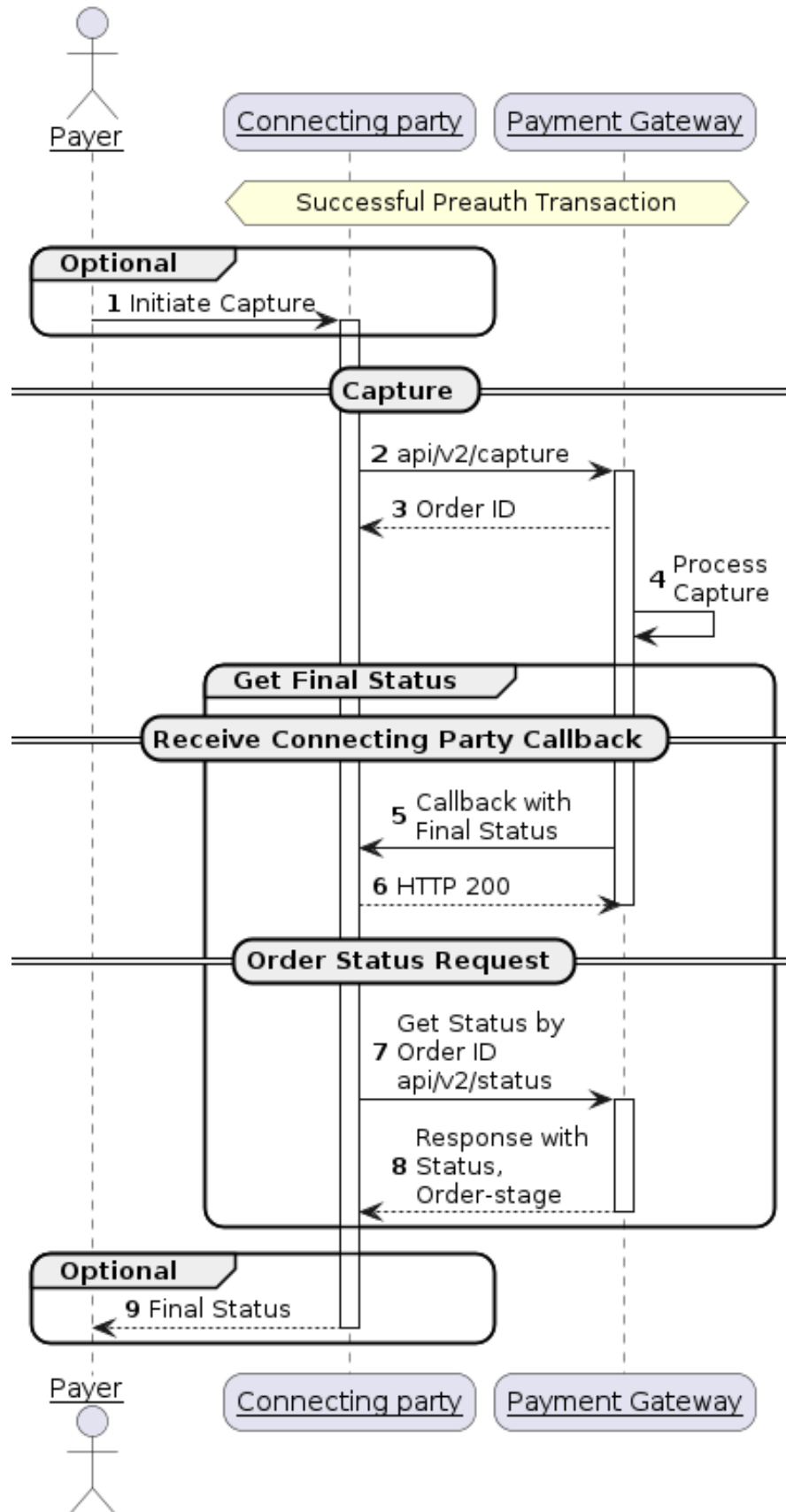
- (1) Capture can be initiated by Connecting Party based on internal business model or Payer's request.
- (2) To implement capture request see </api/v2/capture/>.
- (5) Callback for Capture will be sent only if notify_url was provided in initial transaction request or additional callback URL for Capture transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Capture will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.

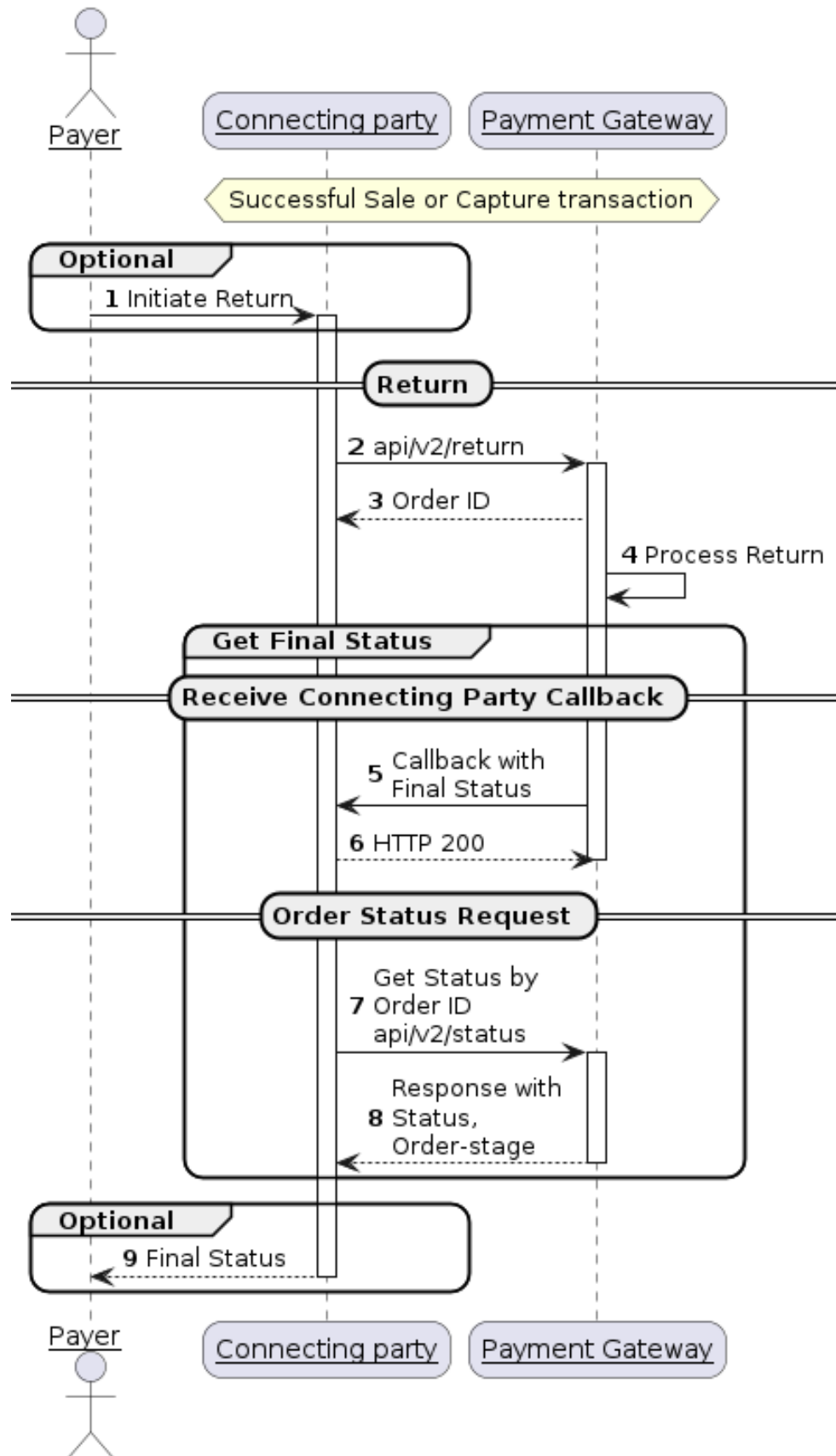
Reversal Flow

- (1) Return can be initiated by Connecting Party based on internal business process or by Payer's request.
- (2) To implement return request see </api/v2/return/>.
- (5) Callback for Return will be sent only if notify_url was provided in initial transaction request or additional callback URL for Return transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Return will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business process or Payer's request.

5.24 Google Pay™ App

- [Introduction](#)
- [How to integrate Google Pay through Doc2.0](#)
- [Terms of Service](#)
- [Google Pay™ Account Setup](#)
- [Google Pay™ Integration in an Android App](#)
 - [Google Pay™ Authentication Types](#)
 - [Google Pay™ Allowed Networks](#)





- [Required Parameters for Google Pay™ Integration](#)
- [Google Pay™ Sale Flow](#)

5.24.1 Introduction

Google Pay™ is an electronic wallet system that allows making one-click payment using a card stored securely on Google side. There is no need to enter card details during Google Pay™ transaction.

Access to the payment mechanism is possible from any device, which supports Google Pay™.

After choosing Google Pay™ payment method, the payer will see the form on which he confirms his Google account and card he is going to use to pay for order. If payer doesn't have a previously registered card, he may do it while making payment.

See terms definitions in [Glossary](#).

Integrate Google Pay™ in App with the following steps:

1. Read and accept [Terms of Service](#).
2. Follow [Google Pay™ Account Setup](#).
3. Get the [Required Parameters for Google Pay™ Integration](#).
4. Check the possible [Google Pay™ Authentication Types](#), and [Google Pay™ Allowed Networks](#).
5. Follow the transaction flow instructions. [Google Pay™ Sale Flow](#).

5.24.2 How to integrate Google Pay through Doc2.0

Please contact Doc2.0 support manager to get help with Google Pay test account.

5.24.3 Terms of Service

The following rules must be complied to use Google Pay™:

- [Google Pay™ API terms of service](#)¹
- [Google Pay™ APIs Acceptable Use Policy](#)²

The following rules must be complied to use Google Pay™ on mobile apps:

¹ <https://payments.developers.google.com/terms/sellertos>

² <https://payments.developers.google.com/terms/aup>

- [Google Pay™ for Android payments overview](#)³
- [Integration checklist for mobile integration](#)⁴
- [Google Pay™ Android Brand Guidelines](#)⁵

5.24.4 Google Pay™ Account Setup

1. Connecting Party must register in . Instead of gmail, Connecting Party can use a corporate email address.

accounts.google.com/signup/v2/webcreateaccount?continue=https%3A%2F%2Fwww.google.ru%2F&hl=en&biz=false&flowName=GlifWebSignIn&flowEntry=SignUp

Google

Create your Google Account

First name | Last name

Username | @gmail.com

You can use letters, numbers & periods

[Use my current email address instead](#)

Password | Confirm

Use 8 or more characters with a mix of letters, numbers & symbols

☐ Show password

[Sign in instead](#) [Next](#)

One account. All of Google working for you.

English (United States) Help Privacy Terms

³ <https://developers.google.com/pay/api/android/overview>

⁴ <https://developers.google.com/pay/api/web/guides/test-and-deploy/integration-checklist>

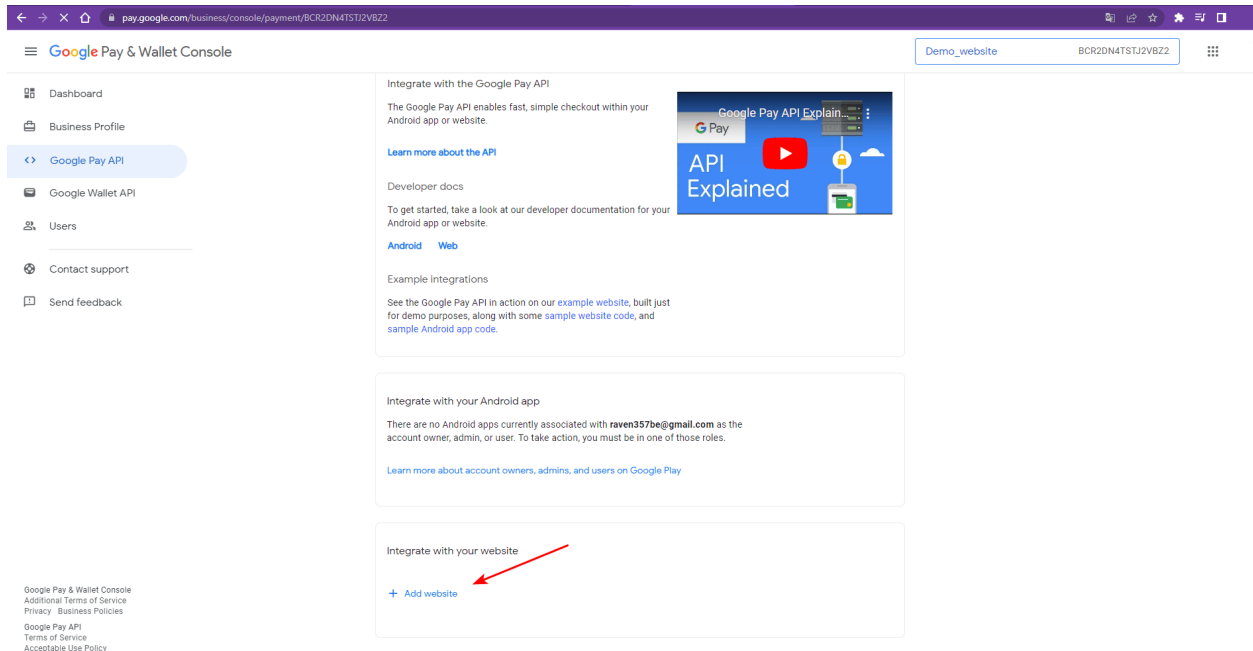
⁵ <https://developers.google.com/pay/api/android/guides/brand-guidelines>

The screenshot shows the Google Account creation page. The URL in the browser is `accounts.google.com/signup/v2/webcreateaccount?continue=https%3A%2F%2Fwww.google.ru%2F&hl=en&biz=false&flowName=GifWebSignIn&flowEntry=SignUp`. The page has a white background with the Google logo at the top. Below the logo, it says "Create your Google Account". There are input fields for "First name", "Last name", and "Your email address". Below the email field, it says "You'll need to confirm that this email belongs to you." and "Create a new Gmail address instead". There are also input fields for "Password" and "Confirm", with a note "Use 8 or more characters with a mix of letters, numbers & symbols" and a checkbox for "Show password". At the bottom, there are links for "Sign in instead" and a "Next" button. On the right side, there is a blue shield icon with a person silhouette and a text box that says "One account. All of Google working for you." Below the form, there are links for "English (United States)", "Help", "Privacy", and "Terms".

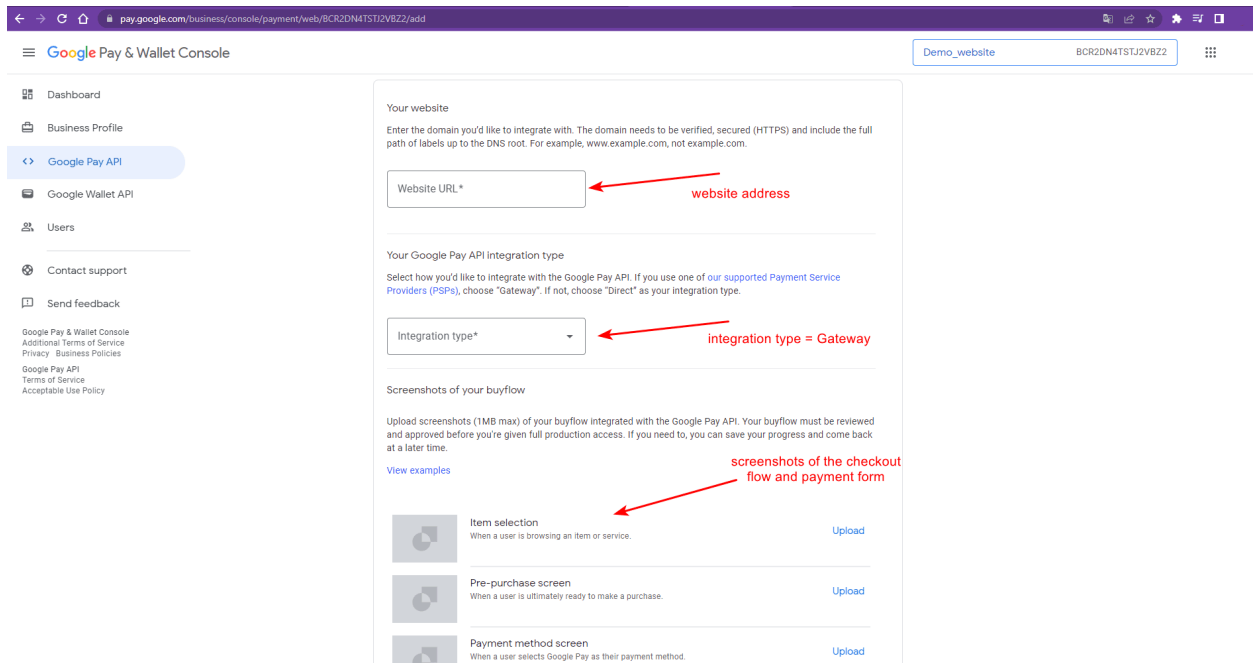
2. Then, from the account screen in the console, save the Merchant ID and proceed to the registration of the website in the Google Pay API section.

The screenshot shows the Google Pay & Wallet Console dashboard. The URL in the browser is `https://pay.google.com/business/console/home/BCR2DN4TSTJ2VBZ2`. The dashboard has a left sidebar with a menu: "Dashboard", "Business Profile", "Google Pay API", "Google Wallet API", "Users", "Contact support", and "Send feedback". The main content area has a "Warning!" banner at the top. Below the banner, there are two main sections: "Business Profile" and "Google Wallet API". The "Business Profile" section has a "Get started" button. The "Google Wallet API" section has a "Learn more" button. In the top right corner, there is a "Demo_website" field and a "Merchant ID" field containing the value "BCR2DN4TSTJ2VBZ2". A red arrow points from the "Merchant ID" field to the "Google Wallet API" section. Another red arrow points from the "Google Pay API" menu item to the "Business Profile" section.

3. Add a website using the "Add website" button.



- Specify screenshots of the checkout and payment form in the website settings, as well as integration type = Gateway and the address of the Connecting Party's website from where the purchase starts.



- Send the website address and Merchant ID to Doc2.0 support managers.

5.24.5 Google Pay™ Integration in an Android App

Google Pay™ integration in an Android app supports only CRYPTOGRAM_3DS (tokenized) authentication type. All requests using PAN_ONLY (non-tokenized) authentication type will be automatically declined.

Google Pay™ Authentication Types

Connecting Party can independently choose the card types in Google Pay™ that will be available for payment. The card type is specified in the allowedAuthMethods Google request parameter.

There are two card types (authentication methods):

1. **PAN_ONLY** (non-tokenized) - cards stored in payer's Google account. These cards are available on any device of the payer.
The token contains the number and expiration date of the physical card. That is why 3-D Secure authentication is required for these cards. 3DS is required for PAN_ONLY cards. It can be indicated by the Success==3DS in Google payment response. 3-D Secure authentication does not differ from the standard 3-D Secure.
2. **CRYPTOGRAM_3DS** (tokenized) - cards that are stored tokenized on the payer's device. Tokenized cards are only available on the device where the card was added to the Google Pay™ app.
The token contains the number and expiration date of the virtual card as well as the 3-D Secure cryptogram. Customer 3-D Secure verification for tokenized cards is not required.

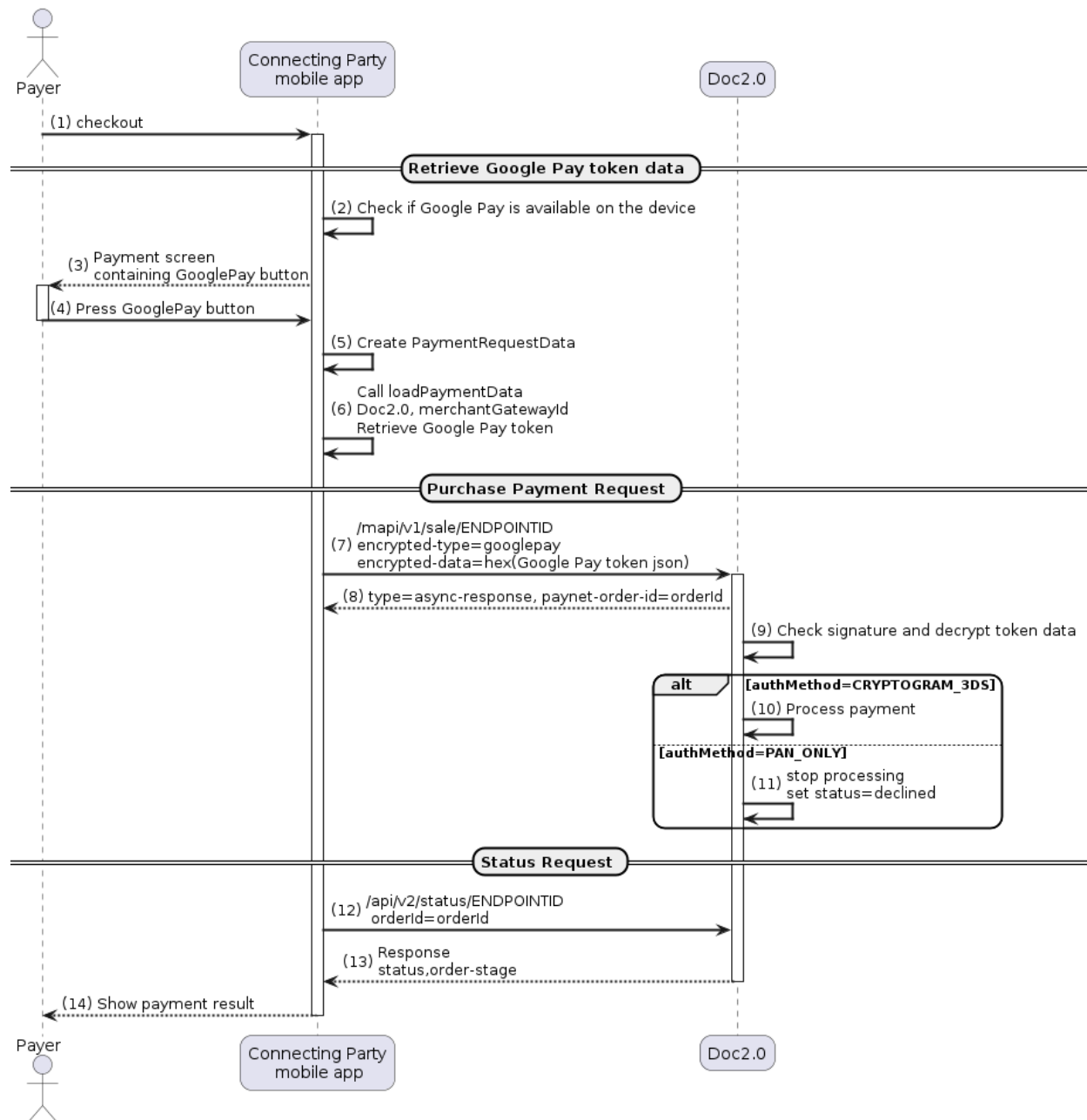
Google Pay™ Allowed Networks

Payment Gateway supports all networks specified in the allowedCardNetworks Google request parameter, such as: VISA, MASTERCARD, etc. For the complete list of allowed networks refer to Doc2.0 support.

Required Parameters for Google Pay™ Integration

1. [Register with Google](#), accept [Google Terms of Service](#), and receive a Google merchant ID after website passes a Google review.
2. Request googleMerchantGatewayID from Doc2.0 support, which looks like: eXXX.mXXX.gate.doc2.com, where:
eXXX - endpointID
mXXX - merchantID
An example of googleMerchantGatewayID: e123.m567.gate.doc2.com
3. Set gatewayID=google.gatewayid.
4. ENDPOINTID or ENDPOINTGROUPID received from Doc2.0 support.
5. Control-key and Connecting Party login received from Doc2.0 support.

Google Pay™ Sale Flow



(2)-(6) To implement retrieving of Google Pay token data within mobile application, please follow official tutorial from Google:

<https://developers.google.com/pay/api/android/guides/tutorial>.

(7) To implement Mobile API sale request, see </mapi/v1/sale/>.

(12) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

5.25 Google Pay™ Web

- [Introduction](#)
- [How to integrate Google Pay through Doc2.0](#)
- [Terms of Service](#)
- [Google Pay™ Account Setup](#)
- [Google Pay™ Integration in a Connecting Party Website](#)
 - [Google Pay™ Authentication Types](#)
 - [Google Pay™ Allowed Networks](#)
 - [Required Parameters for Google Pay™ Integration](#)
 - [Google Pay™ Web Flow](#)
 - [Capture Flow](#)
 - [Cancel Flow](#)

5.25.1 Introduction

Google Pay™ is an electronic wallet system that allows you to make one-click payment using a card stored securely on Google side. There is no need to enter card details during Google Pay™ transaction.

Access to the payment mechanism is possible from any device, which supports Google Pay™. After choosing Google Pay™ payment method, the payer will see the form on which he confirms his Google account and card he is going to use to pay for order. If payer doesn't have a previously registered card, he may do it while making payment.

See terms definitions in [Glossary](#).

Integrate Google Pay™ in App with the following steps:

1. Read and accept [Terms of Service](#).
2. Follow [Google Pay™ Account Setup](#).
3. Get the [Required Parameters for Google Pay™ Integration](#).
4. Check the possible [Google Pay™ Authentication Types](#), and [Google Pay™ Allowed Networks](#).
5. Follow the transaction flow instructions: [Google Pay™ Web Flow](#).

5.25.2 How to integrate Google Pay through Doc2.0

Please contact Doc2.0 support manager to get help with Google Pay test account.

5.25.3 Terms of Service

The following rules must be complied to use Google Pay™:

- [Google Pay™ API terms of service](#)⁶
- [Google Pay™ APIs Acceptable Use Policy](#)⁷

The following rules must be complied to use Google Pay™ on websites:

- [Google Pay™ for websites overview](#)⁸
- [Integration checklist for website integration](#)⁹
- [Google Pay™ Web Brand Guidelines](#)¹⁰

5.25.4 Google Pay™ Account Setup

1. Connecting Party must register in . Instead of gmail, you can use a corporate email address.

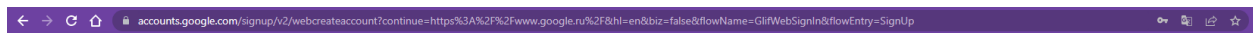
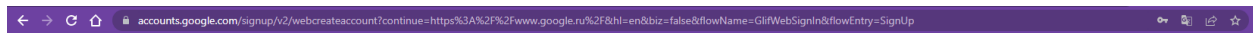
⁶ <https://payments.developers.google.com/terms/sellertos>

⁷ <https://payments.developers.google.com/terms/aup>

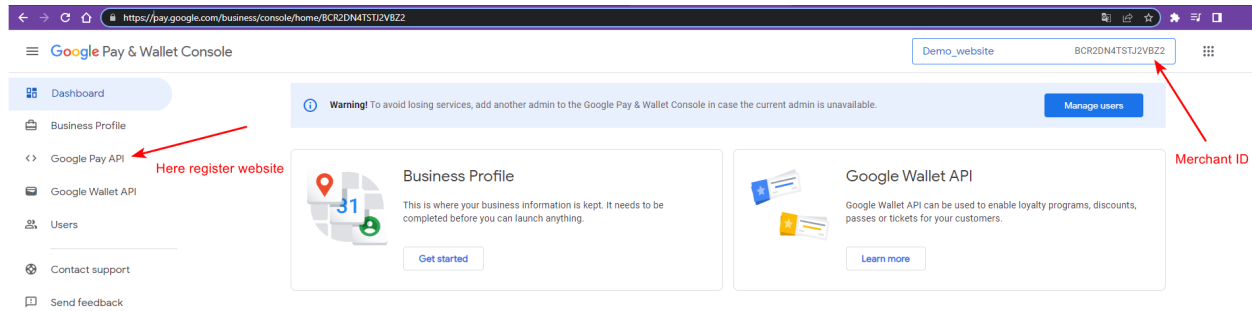
⁸ <https://developers.google.com/pay/api/web/overview>

⁹ <https://developers.google.com/pay/api/android/guides/test-and-deploy/integration-checklist>

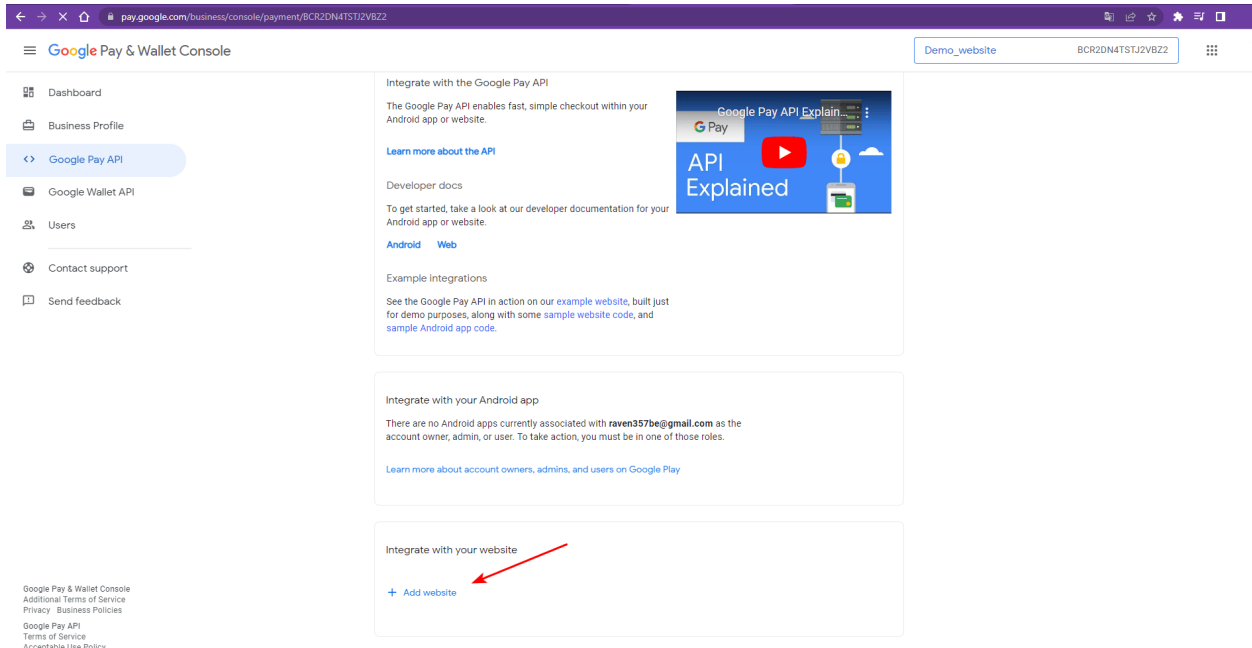
¹⁰ <https://developers.google.com/pay/api/web/guides/brand-guidelines>



2. Then, from the account screen in the console, save the Merchant ID and proceed to the registration of the website in the Google Pay API section.



3. Add a website using the “Add website” button.



4. Specify screenshots of the checkout and payment form in the website settings, as well as integration type = Gateway and the address of the Connecting Party’s website from where the purchase starts.

5. Send the website address and Merchant ID to Doc2.0 support managers.

5.25.5 Google Pay™ Integration in a Connecting Party Website

Google Pay™ Authentication Types

Connecting Party can independently choose the card types in Google Pay™ that will be available for payment. The card type is specified in the `allowedAuthMethods` Google request parameter.

There are two card types (authentication methods):

1. **PAN_ONLY** (non-tokenized) - cards stored in payer's Google account. These cards are available on any device of the payer.
The token contains the number and expiration date of the physical card. That is why 3-D Secure authentication is required for these cards. 3DS is required for PAN_ONLY cards. It can be indicated by the `Success==3DS` in Google payment response. 3-D Secure authentication does not differ from the standard 3-D Secure.
2. **CRYPTOGRAM_3DS** (tokenized) - cards that are stored tokenized on the payer's device. Tokenized cards are only available on the device where the card was added to the Google Pay™ app.
The token contains the number and expiration date of the virtual card as well as the 3-D Secure cryptogram. Customer 3-D Secure verification for tokenized cards is not required.

Google Pay™ Allowed Networks

Payment Gateway supports all networks specified in the allowedCardNetworks Google request parameter, such as: VISA, MASTERCARD, etc. For the complete list of allowed networks refer to Doc2.0 support.

Required Parameters for Google Pay™ Integration

1. [Register with Google](#), accept [Google Terms of Service](#), and receive a Google merchant ID after website passes a Google review.
2. ENDPOINT or ENDPOINTGROUPID received from Doc2.0 support.
3. Control-key and Connecting Party login received from Doc2.0 support.

Google Pay™ Web Flow

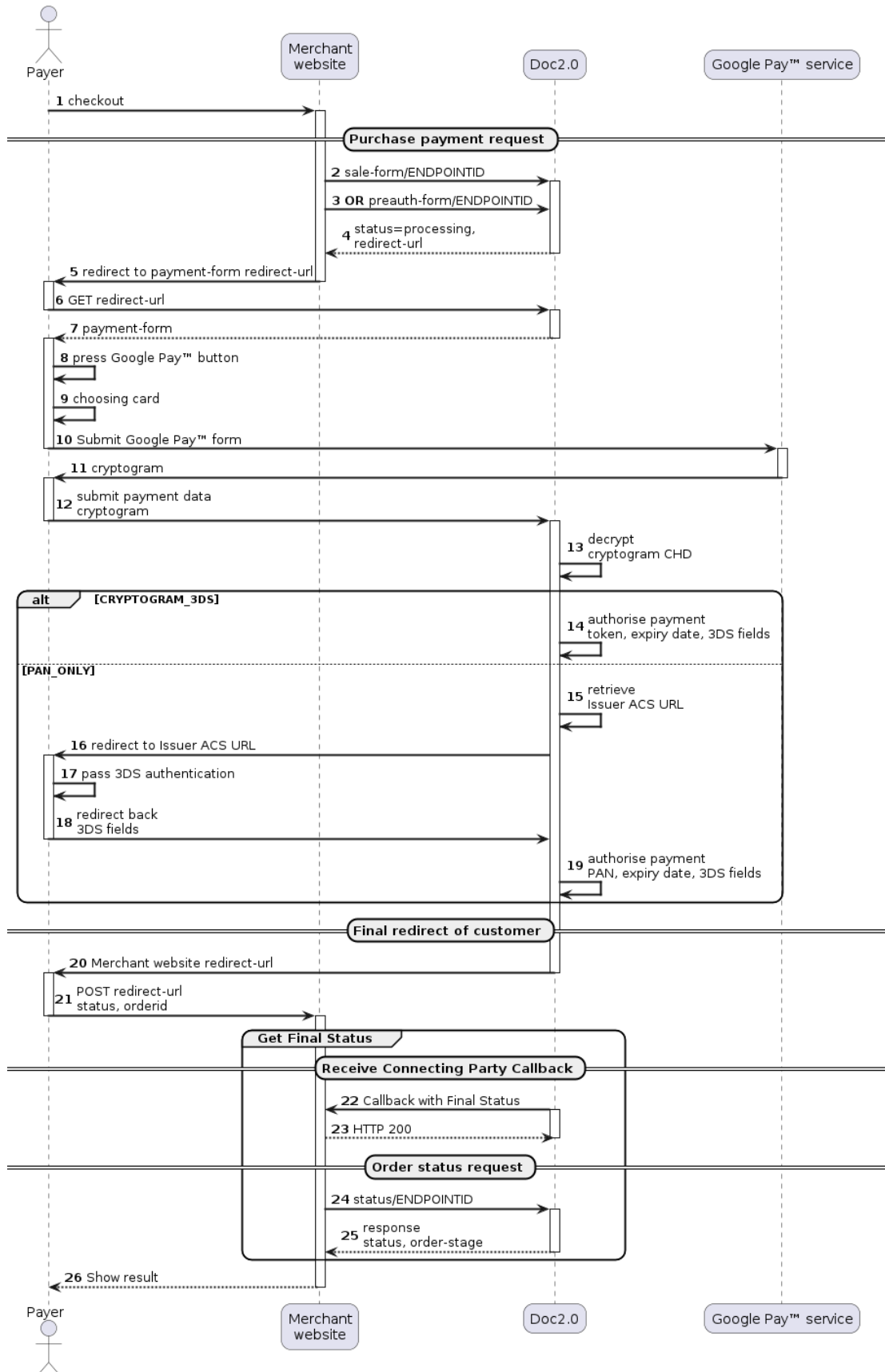
Below we can see Google Pay™ Purchase Flow and Integration steps:

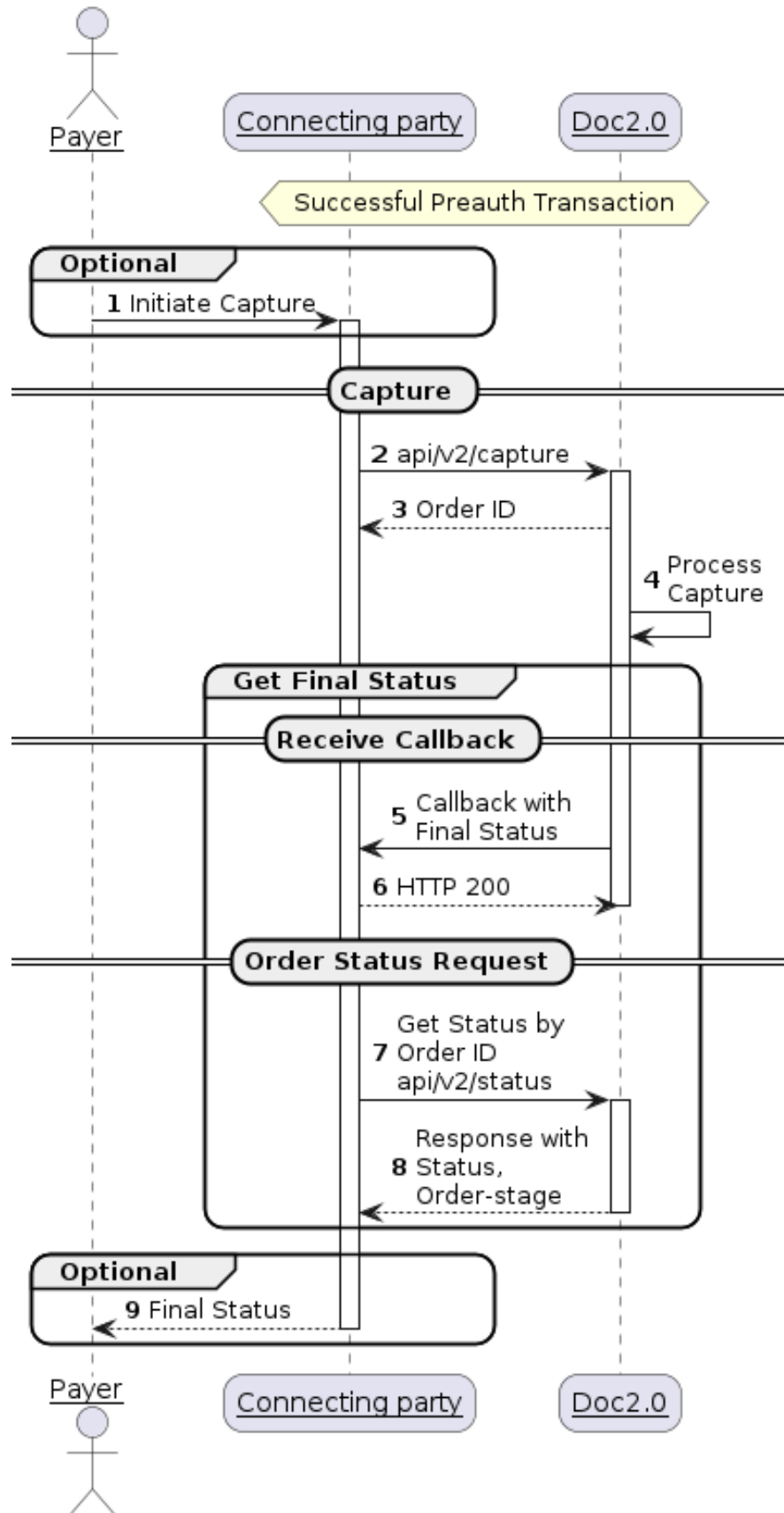
- (2) To implement sale-form payment request see [/api/v2/sale](#).
- (3) Instead of sale-form it is possible to implement preauth-form payment request see [/api/v2/preauth-form](#). Preauth allows to hold the specified amount in the Payer's card account for a limited time. Preauth can be followed by [Capture](#) or [Cancel](#).
- (20) To implement Final Redirect see [Final Redirect](#).
- (22) To implement callback with final status handling see [Connecting Party Callbacks](#).
- (24) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

Capture Flow

Capture is a transaction followed after preauth which deducts the locked amount from Payer's card. It is important to know that the block remains for a definite period of time depending on whether this is a debit or a credit card (usually the maximum block period is 7 days for debit cards and 28 days for credit cards).

- (1) Capture can be initiated by Connecting Party based on internal business model or Payer's request.
- (2) To implement capture request see [/api/v2/capture/](#).
- (5) Callback for Capture will be sent only if notify_url was provided in initial transaction request or additional callback URL for Capture transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Capture will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).
- (7) To implement order status request see [/api/v2/status/](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.





Cancel Flow

Cancel is opposite of Capture which cancels the deduction and returns locked amount back to Payer's card. It is important to know that the block remains for a definite period of time depending on whether this is a debit or a credit card (usually the maximum block period is 7 days for debit cards and 28 days for credit cards).

(1) Cancel can be initiated by Connecting Party based on internal business model or Payer's request.

(2) To implement cancel request see </api/v2/return/>.

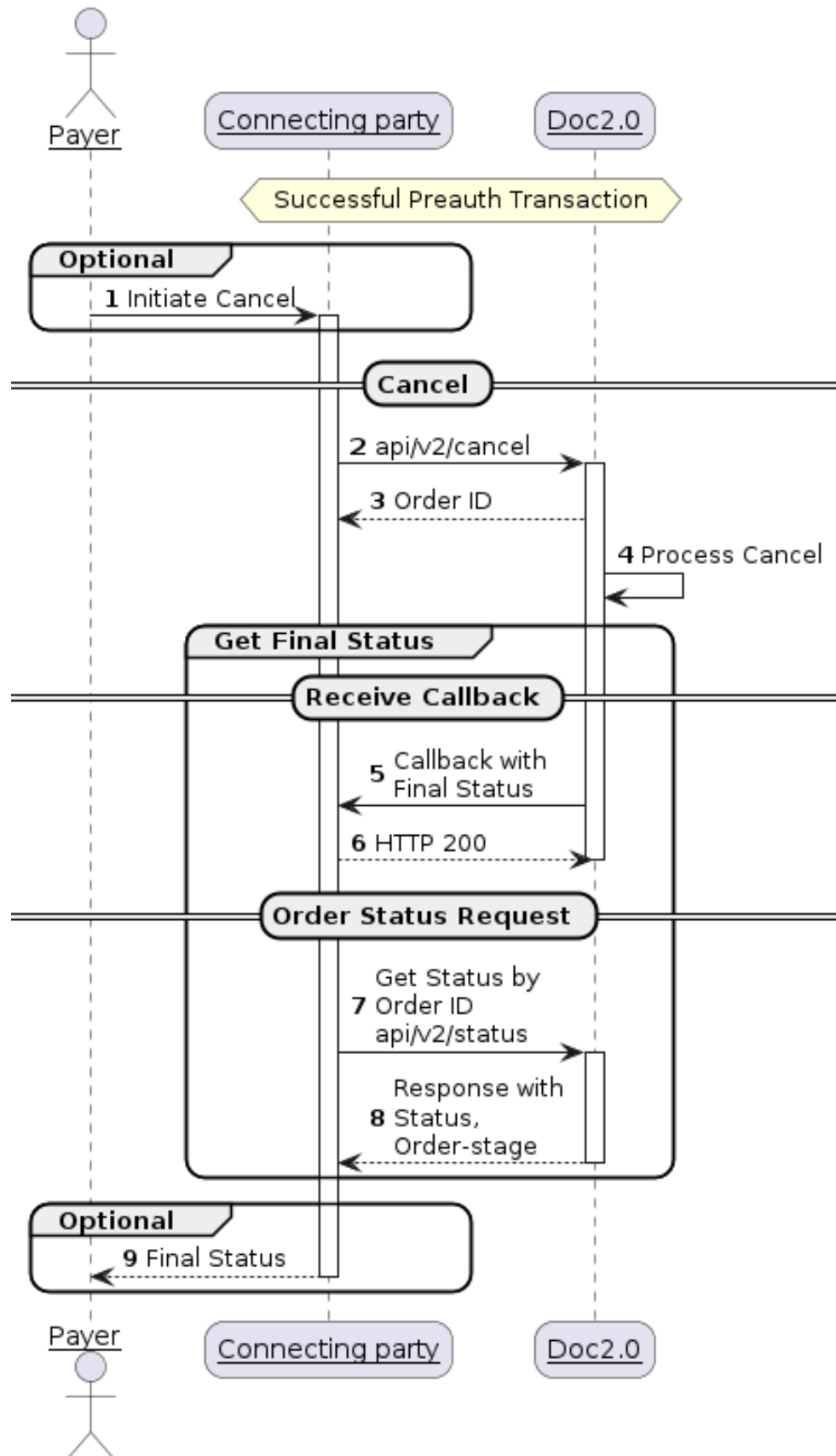
(5) Callback for Cancel will be sent only if notify_url was provided in initial transaction request or additional callback URL for Cancel transactions is specified on the endpoint level. If server_callback_url was provided in initial transaction request, callback for Cancel will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).

(7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

(9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.

5.26 Apple Pay Web

- [Introduction](#)
- [Apple Pay Flow](#)
- [Capture Flow](#)
- [Cancel Flow](#)
- [Apple Pay Account Setup](#)
 - [Register Merchant ID](#)
 - [Create Certificate Signing Request](#)
 - [Issue Payment Processing Certificate](#)
 - [Get the Certificate Private Key and Private Key Password](#)
 - [Reissue of Payment Processing Certificate](#)
- [Merchant Identity Certificate](#)
 - [Domain Registration and Verification](#)
 - [Creation of Merchant Identity Certificate](#)
 - [Create Certificate Signing Request](#)
 - [Issue Merchant Identity Certificate](#)
 - [Get the Certificate Private Key and Private Key Password](#)
 - [Domain Re-verification and Merchant Identity Certificate Reissue](#)



5.26.1 Introduction

Apple Pay is an electronic wallet system that allows Payer to make one-click payment using a card token issued together with Apple and Issuer bank. There is no need to enter card details during Apple Pay transaction. Access to this payment mechanism is possible from any device, which supports Apple Pay. After choosing Apple Pay payment method, the payer will see the form on which he confirms his Apple account and card he is going to use to pay for order. If payer doesn't have a previously registered card, he may do it while making payment.

See terms definitions in [Glossary](#).

If accepting payments occurs on the Doc2.0 payment page, no additional integration with Apple is required from the Connecting Party.

To use Apple Pay on the Doc2.0 payment page, do the following:

1. Enable Apple Pay on the payment page template by adding macro.
2. Inform Doc2.0 support about the need to connect Apple Pay and send updated templates.

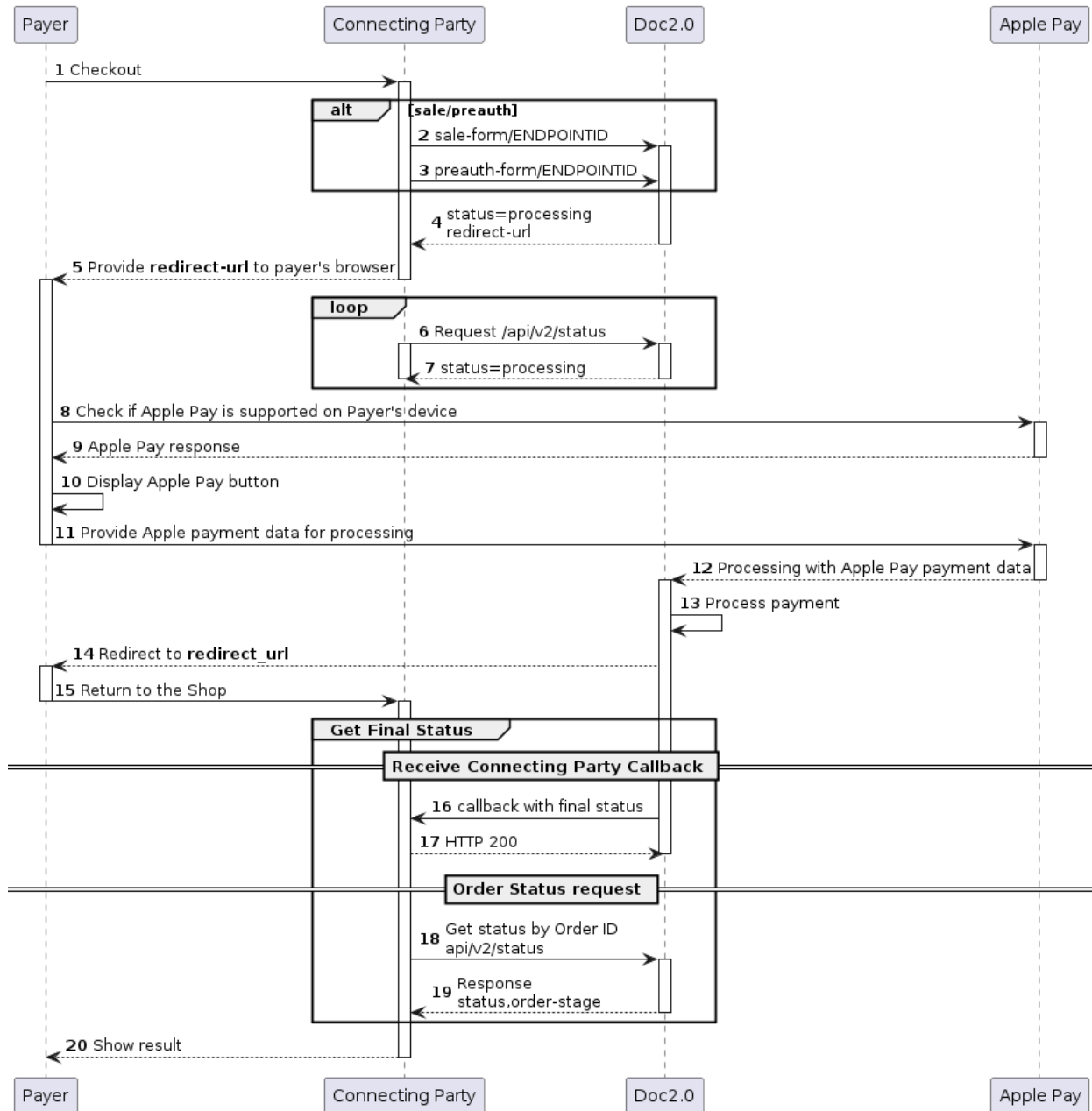
The Apple Pay button will automatically appear on the Doc2.0 checkout page if:

1. The payer uses the Safari browser on a device that supports Apple Pay.
2. The payer has Apple Pay cards available for payment.

Note: It might be necessary for Connecting Party to register [Merchant ID](#), issue [Payment Processing Certificate](#) and [Merchant Identity Certificate](#) in Apple in order to launch Apple Pay. For details please contact Doc2.0 support manager.

5.26.2 Apple Pay Flow

- (2) To implement sale-form payment request see [/api/v2/sale-form](#).
- (3) Instead of sale-form it is possible to implement preauth-form payment request, see [/api/v2/preauth-form](#). Preauth allows to hold the specified amount in the Payer's card account for a limited time. Preauth can be followed by [Capture](#) or [Cancel](#).
- (14) Payer is returned to Connecting Party site. To implement final redirect see [Final Redirect](#).
- (16) To implement callback with final status handling see [Connecting Party Callbacks](#).
- (18) To implement order status request see [/api/v2/status](#). Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.



5.26.3 Capture Flow

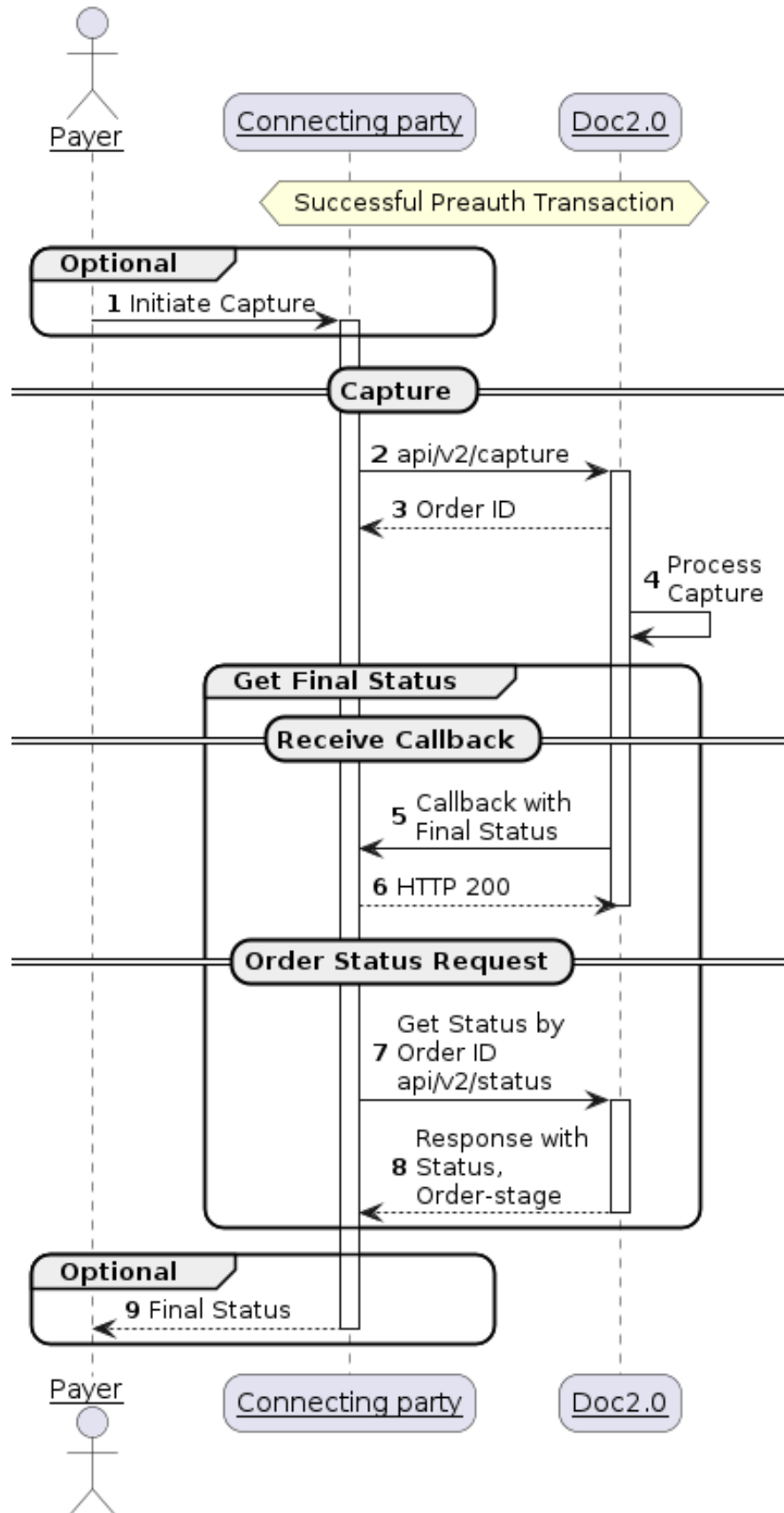
Capture is a transaction followed after preauth which deducts the locked amount from Payer's card. It is important to know that the block remains for a definite period of time depending on whether this is a debit or a credit card (usually the maximum block period is 7 days for debit cards and 28 days for credit cards).

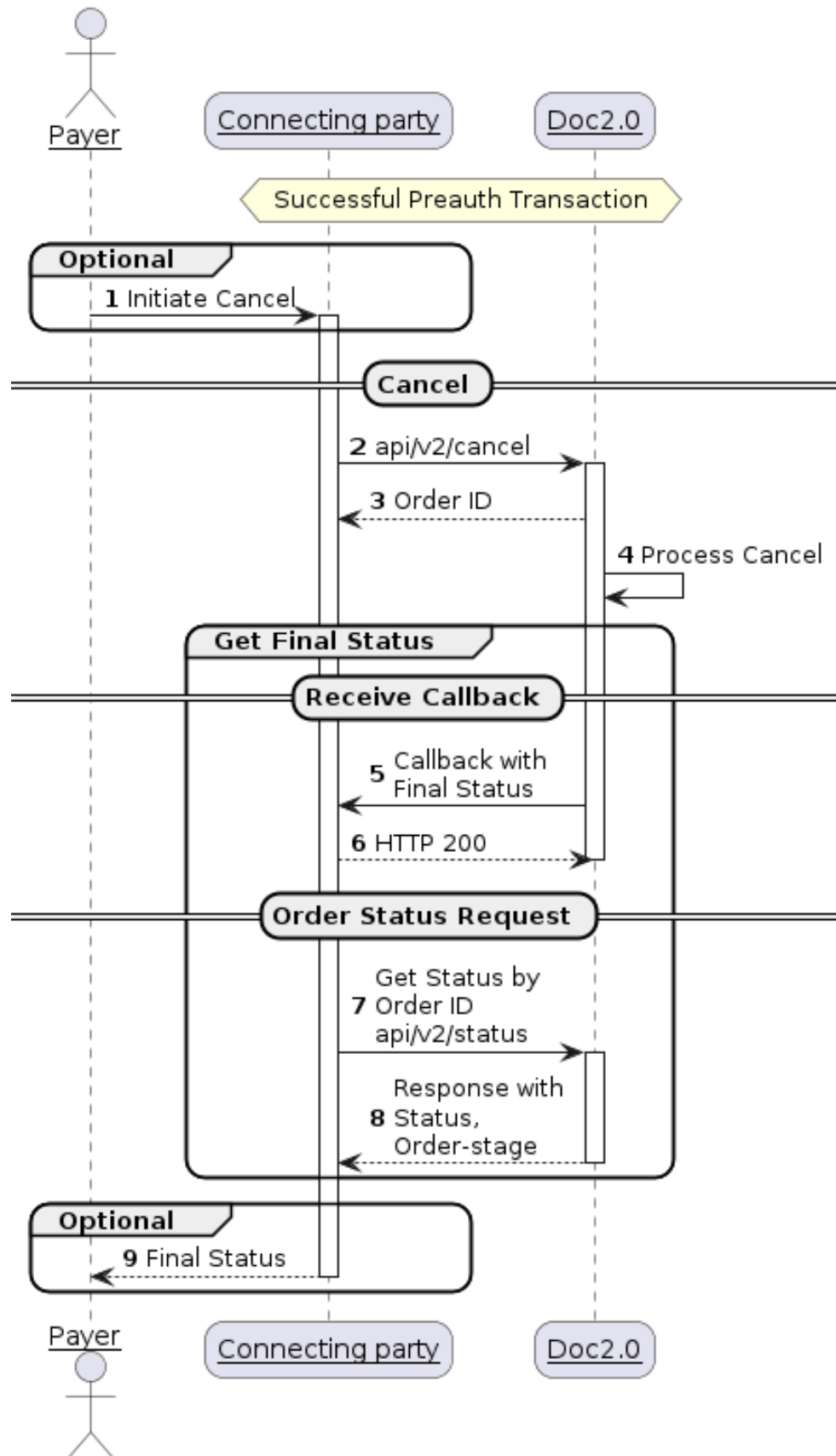
- (1) Capture can be initiated by Connecting Party based on internal business model or Payer's request.
- (2) To implement capture request see </api/v2/capture/>.
- (5) Callback for Capture will be sent only if `notify_url` was provided in initial transaction request or additional callback URL for Capture transactions is specified on the endpoint level. If `server_callback_url` was provided in initial transaction request, callback for Capture will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.

5.26.4 Cancel Flow

Cancel is opposite of Capture which cancels the deduction and returns locked by Preauth amount back to Payer's card. It is important to know that the block remains for a definite period of time depending on whether this is a debit or a credit card (usually the maximum block period is 7 days for debit cards and 28 days for credit cards).

- (1) Cancel can be initiated by Connecting Party based on internal business model or Payer's request.
- (2) To implement cancel request see </api/v2/return/>.
- (5) Callback for Cancel will be sent only if `notify_url` was provided in initial transaction request or additional callback URL for Cancel transactions is specified on the endpoint level. If `server_callback_url` was provided in initial transaction request, callback for Cancel will not be sent. To implement callback with final status handling see [Connecting Party Callbacks](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (9) Final Status can be sent by Connecting Party based on internal business model or by Payer's request.





5.26.5 Apple Pay Account Setup

To configure the terminal on the Payment Gateway side, the following data needs to be sent to the Doc2.0 support service:

- Merchant ID.
- The private key of the Payment Processing Certificate.
- Private key password.

To get the necessary data, the following steps are required:

1. [Register Merchant ID](#).
2. [Create Certificate Signing Request](#).
3. [Issue Payment Processing Certificate](#).
4. [Get the certificate private key and private key password](#).

Note:

Please note that the Payment Processing Certificate lasts **25 months**. To renew the certificate, see the section [“Reissue of Payment Processing Certificate”](#).

Register Merchant ID

Merchant ID - Apple Merchant Site ID (or MID) is a unique identification number that tells the payment processing systems involved in a transaction where to send which funds.

1. Open developer account. Go to Certificate, Identifiers & Profiles.
2. Go to Identifiers.
3. Click on “+”.
4. Select Merchant IDs.
5. Specify the description and the Merchant ID. It is recommended to use the website domain in reverse order with the Connecting Party’s prefix. Example: connecting party.com.Doc2.0.applepay.test.
6. Click Register.

Merchant ID is now registered.

Create Certificate Signing Request

Certificate Signing Request is a request to issue a certificate, in our case in order to get Payment Processing Certificate.

1. Open Keychain Access on Mac.
2. Certification assistant -> Request a certificate from a Certificate Authority.
3. Specify the email, to which the developer's account was registered, the name (also key name). Select Saved to disk and Specify information about a pair of keys manually.
4. Select the key size 256 bits and the ECC Algorithm. Click Continue button. Save the .certSigningRequest file.

Issue Payment Processing Certificate

1. Return to the developer's office. Go to Certificates, Identifiers & Profiles / Identifiers. Select needed Merchant ID.
2. Click Create Certificate in Apple Pay Payment Processing Certificate section.
3. Select "No" and click Continue.
4. Choose .certSigningRequest file from step 2 and click Continue.
5. Download **Payment Processing Certificate**.

Get the Certificate Private Key and Private Key Password

1. Double click Payment Processing Certificate on Mac to install.
2. Open Keychain Access and go to Certificates section.
3. Find the certificate. Select the created key by clicking on the arrow next to it.
4. Select Export with the right mouse button.
5. Select the folder to save the key file in .p12 format, click Save.
6. Create password and click OK. **Private key** <name>.p12 and **private key password** are ready to work.

Reissue of Payment Processing Certificate

The Connecting Party must independently monitor the validity period of the Apple Pay certificate (25 months). Before the certificate expires, a new certificate must be issued. This requires the following:

1. Create a new Certificate Signing Request, as described in [step 2](#).
2. Issue a new Payment Processing Certificate as in [step 3](#). After creating a new certificate for the current Merchant ID, there will be two certificates: **old** (active) and **new** (requiring activation). At this step **there is no need** to activate the new certificate.

3. Get the certificate private key, private key password and send them to Doc2.0 support service as in [step 4](#).
4. Activate a new certificate **only in consultation with Doc2.0 support service**. After activation, only the new certificate will be used, the old one will become inactive.

5.26.6 Merchant Identity Certificate

The following data needs to be sent to the Doc2.0 support service:

1. Merchant ID.
2. The private key of the Merchant Identity Certificate.
3. Private key password.

Note: Please note that the validity period of the Merchant Identity Certificate and domain verification is **25 months**. To renew the certificate and re-verify domains, see the section [Re-verification of domains and reissue of the Merchant Identity Certificate](#).

Domain Registration and Verification

Warning: Before receiving the Merchant Identity Certificate, it is compulsory to create and validate domains where Apple Pay will be used.

1. Go to the developer's office and go to Certificates, Identifiers & Profiles. [Create](#) a Merchant ID before domain registration.
2. Go to Identifiers.
3. App IDs -> Merchant IDs.
4. Select the needed Merchant ID.
5. Click Add Domain in Merchant Domains.
6. Specify the domain where Apple Pay will be used and click Save.
7. To confirm ownership of a domain, download the file and place it at the specified address, the file at the specified address must be accessible from outside the Connecting Party's network.
8. Click Verify to verify the domain.

Creation of Merchant Identity Certificate

- Step 1. [Create Certificate Signing Request](#)
- Step 2. [Issue Merchant Identity Certificate](#)
- Step 3. [Get the certificate private key and private key password](#)

Create Certificate Signing Request

Certificate Signing Request is a request to issue a certificate, in our case in order to get Payment Processing Certificate.

1. Open Keychain Access on Mac.
2. Certification assistant -> Request a certificate from a Certificate Authority.
3. Specify the email, to which the developer's account was registered, the name (also key name). Select Saved to disk and Specify information about a pair of keys manually.
4. Select the key size 2048 bits and the RSA Algorithm. Click Continue button. Save the .certSigningRequest file.

Issue Merchant Identity Certificate

1. Return to the developer's office. Go to Certificates, Identifiers & Profiles / Identifiers.
2. Go to Identifiers.
3. App IDs -> Merchant IDs.
4. Select the needed Merchant ID.
5. Create Certificate in Apple Pay Merchant Identity Certificate section.
6. Select created in step 1 .certSigningRequest file and click Continue.
7. Download **Merchant Identity Certificate**.

Get the Certificate Private Key and Private Key Password

1. Double click Merchant Identity Certificate on Mac to install.
2. Open Keychain Access and go to Certificates section.
3. Find the certificate. Select Export with the right mouse button.
4. Select the folder to save the key file in .p12 format, click Save.
5. Create password and click OK. **Private key** <name>.p12 and **private key password** are ready to work.

Domain Re-verification and Merchant Identity Certificate Reissue

The Connecting Party must independently monitor the validity of the certificate and domains' verification (25 months). A new Merchant Identity Certificate must be issued before the certificate expires. Certificates can be used concurrently, so the procedure for issuing a new one is as described in the "Creating a Merchant Identity Certificate" section. Before the expiration of the domain verification period, the Verify button is activated next to it. The verification procedure corresponds to that described in the section "Domain registration and verification".

5.27 eCheck Sale

- [Introduction](#)
- [eCheck Flow](#)

5.27.1 Introduction

Doc2.0 allows Connecting Party to accept electronic check payments for goods and services purchased via e-commerce websites. An electronic check works much like a regular check, only instead of using a paper check to provide their bank account, routing number and payment authorization, the Payer provides Connecting Party with the necessary payment information through other methods, allowing the Connecting Party to process the information electronically.

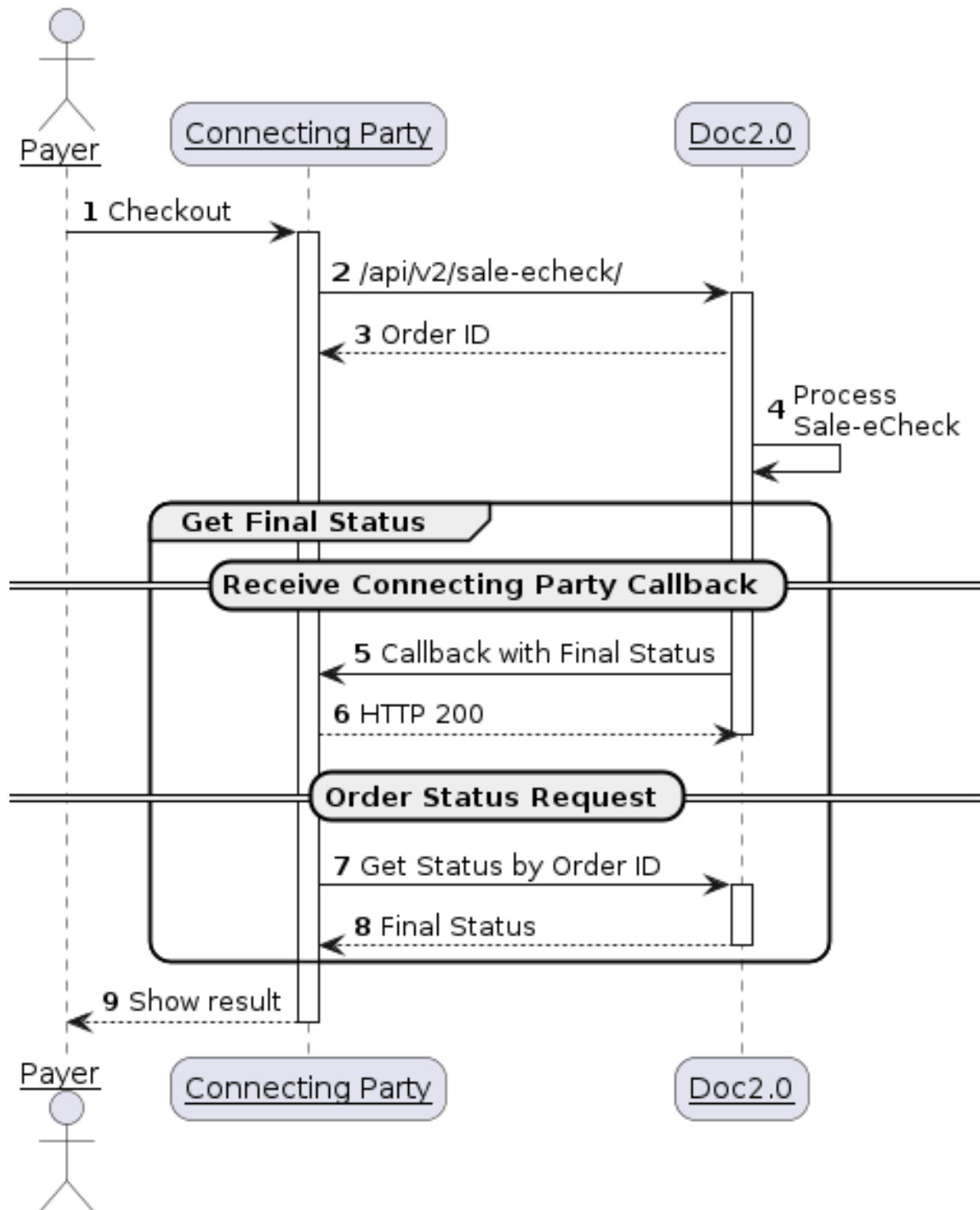
See terms definitions in [Glossary](#).

5.27.2 eCheck Flow

- (2) To implement sale-eCheck request see </api/v2/sale-echeck/>.
- (5) To implement callback with final status handling see [Connecting Party Callback](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

5.28 Buy Now Button

- [Introduction](#)
- [Required Parameters for Buy Now Button Integration](#)
- [Buy Now Button Form Flow](#)



5.28.1 Introduction

Buy Now Button integration is relevant for Connecting Party who has limited portfolio of products to sell. It is the easiest way to integrate with Doc2.0. This way of integration doesn't require much technical effort. Buy Now Button integration also allows Connecting Party to exclude itself from storing, processing, or transmitting Payer's cardholder data or other sensitive payment details. Such data is submitted by Payer on Doc2.0 hosted customer details form and payment form in PCI DSS certified environment.

See terms definitions in [Glossary](#).

1. Payer clicks on Buy Now Button, gets redirected to customer details form provided by Payment Gateway and submits customer data;

Order summary

test

Total: 55.00 USD

Customer information

Last Name

First Name

Address

City

Zip Code

E-Mail

Phone

Customer birth date

Country

State

 **SECURE CONNECTION** [Process payment](#)

2. Payers then gets redirected to payment form and submits payment (cardholder) data.

Order summary

Test Order Description

Total: 10.42 USD

Pay securely with credit card

Credit or Debit card number:

Card expiration date:

01 / January

2022

Card printed name:

Card security code (CVV2/CVC2):






SECURE CONNECTION

Process payment

Cancel

3. Contextual data is gathered by Doc2.0 to process the transaction, Payer stays on wait form until transaction is finished;

Order summary

test

Total: 55.00 USD



Please wait, your payment is being processed...

This page refreshes automatically each 5 seconds...
Next time it will refresh in 4 seconds.

Check result

4. Payer's browser gets redirected to the Connecting Party website to the resultant page.

5.28.2 Required Parameters for Buy Now Button Integration

Connecting Party provides to Payment Gateway support the following information for each product (item): Amount, Description, Final Redirect URL, Destination (purpose of payment) and Payment tool (payment method).

Payment Gateway provides to Connecting Party the Buy Now Button URLs for each product (item). URLs have the following format:

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/pay/ENDPOINTID/ITEMID	https://gateway.doc2.com/paynet/api/v2/pay/ENDPOINTID/ITEMID

5.28.3 Buy Now Button Form Flow

(2) Place the Buy Now Button HTML element on checkout page available for the Payer.

(9) To implement final redirect see [Final Redirect](#).

(11,12) To implement order status request see [/api/v2/status/](#). Connecting Party may request order status after receiving transaction ID in either Connecting Party callback or final redirect.

(13) To implement callback with final status handling see [Connecting Party Callbacks](#).

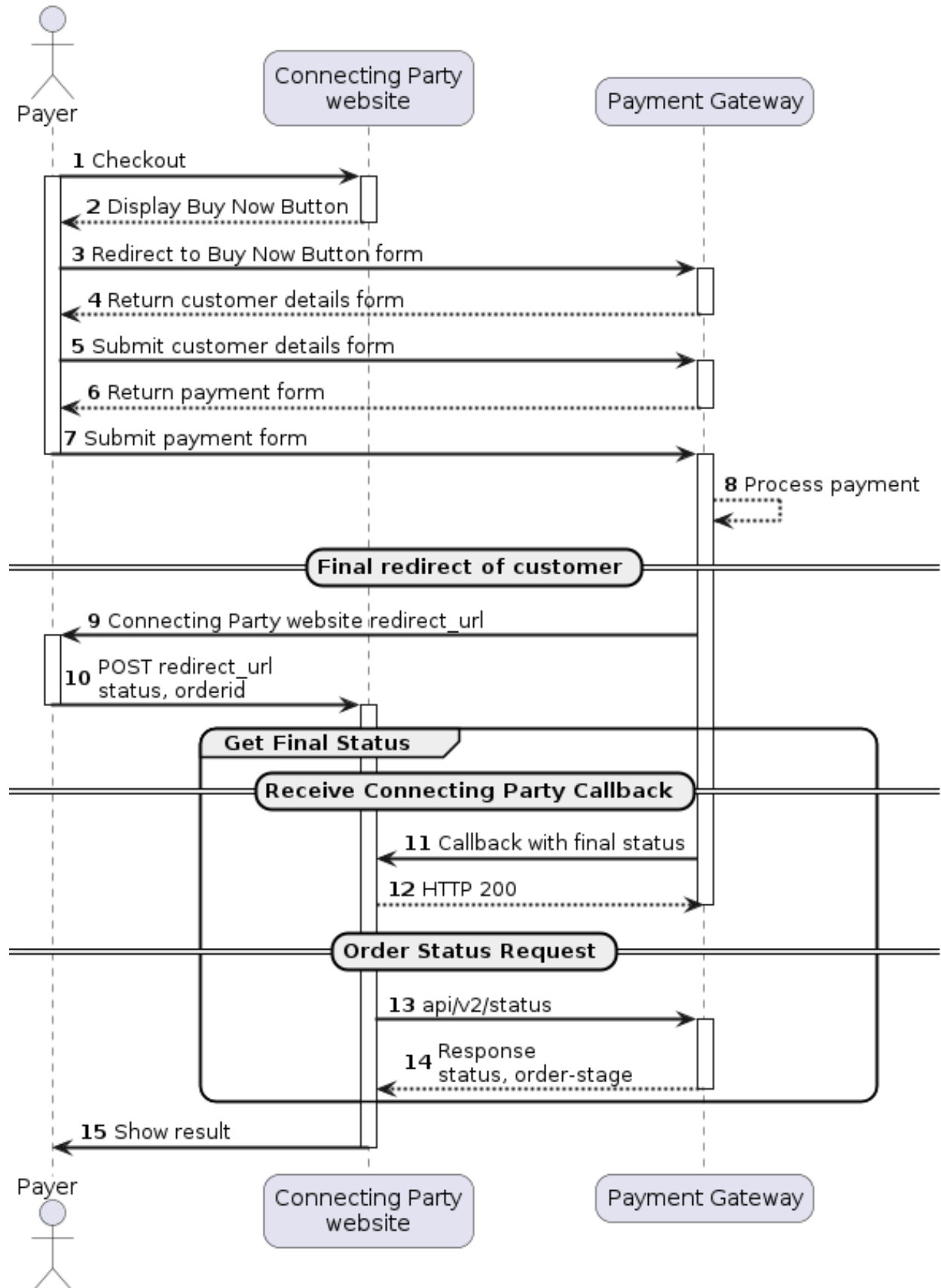
5.29 Direct Bank transfer

- [Introduction](#)
- [Direct Bank Transfer Flow](#)

5.29.1 Introduction

A Direct Bank transfer is an international bank transfer made via the Swift network. This method allows to process bank transfers using the Direct method (with direct transfer of the Payer's bank data).

See terms definitions in [Glossary](#).



5.29.2 Direct Bank Transfer Flow

- (2) To implement bank wire transfer request see </api/v2/bank-wire-transfer/>.
- (5) To implement callback with final status handling see [Connecting Party Callback](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

5.30 Bank transfer with Invoice

- [Introduction](#)
- [Bank Wire Transfer Flow](#)
- [Simple Bank Transfer Flow Statuses](#)
- [Extended Bank Transfer Flow Statuses](#)

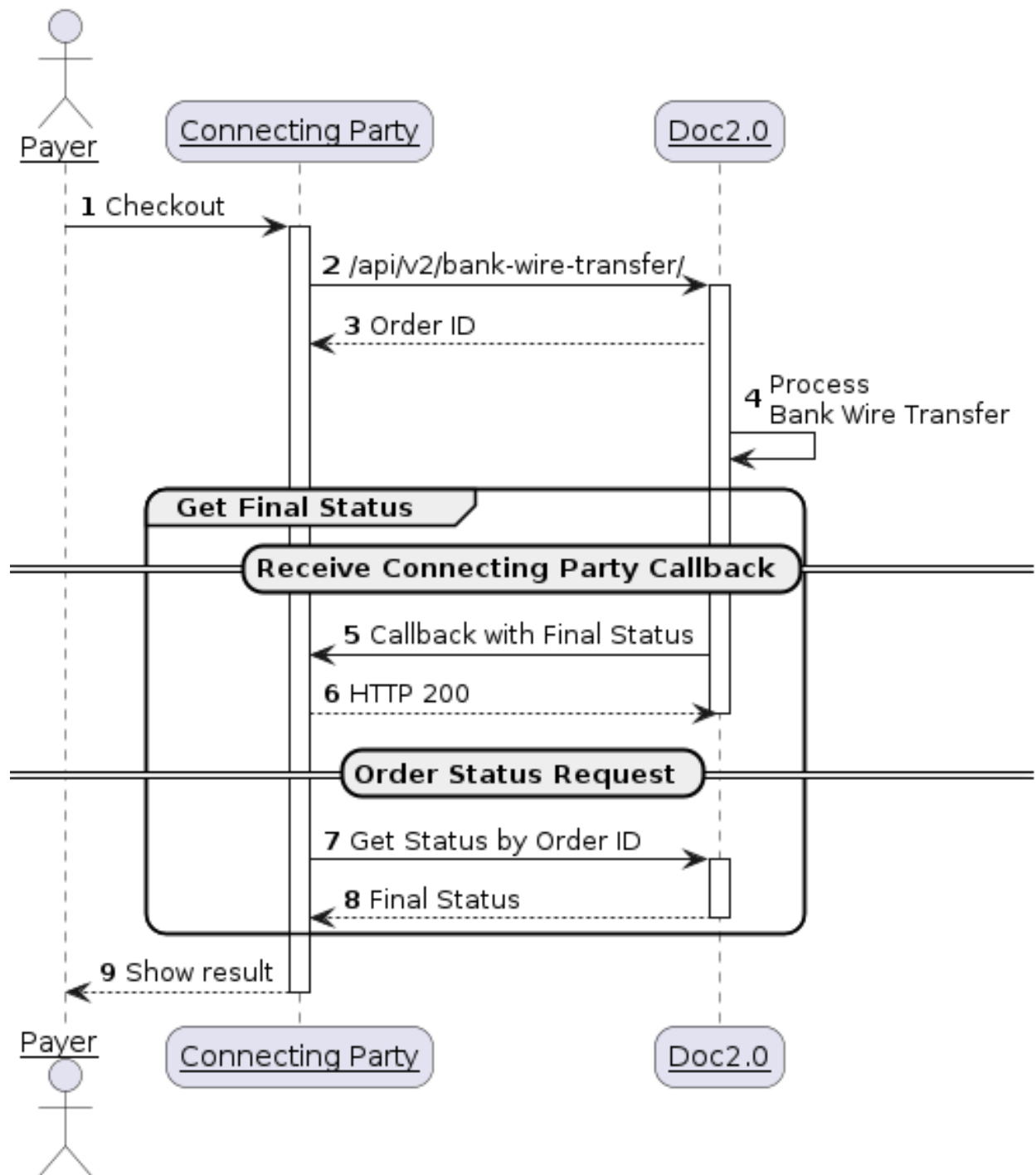
5.30.1 Introduction

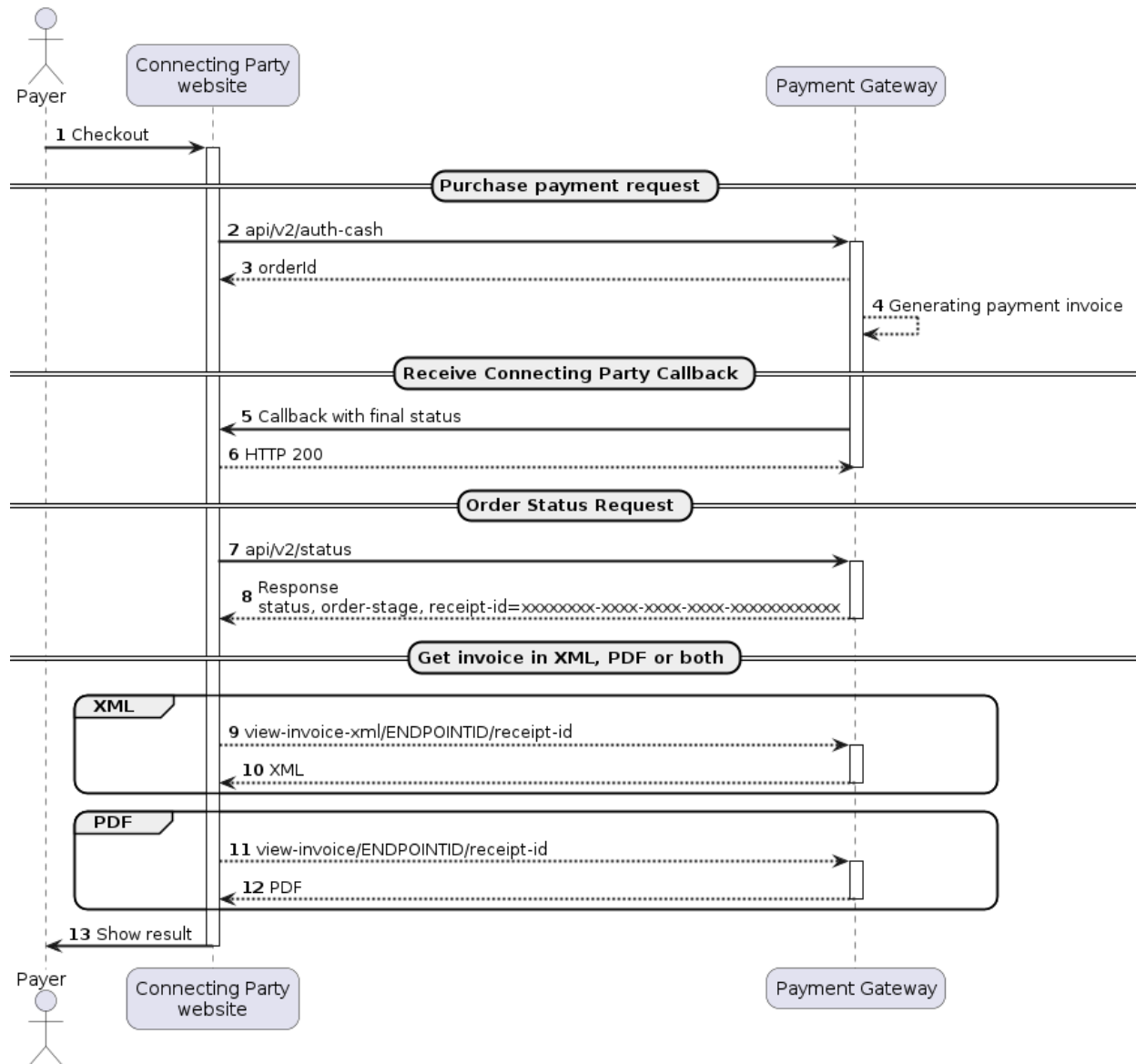
This Use-Case describes how Connecting Party can issue invoices for Swift or similar wire bank transfer systems in Payment Gateway and display these invoices to Payers. Incoming bank transfer statuses might be updated automatically or manually.

See terms definitions in [Glossary](#).

5.30.2 Bank Wire Transfer Flow

- (2) To implement bank transfer request see </api/v2/auth-cash/>.
- (5,6) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response. receipt-id will be received in status response.
- (7) To implement callback with final status handling see [Connecting Party](#).
- (9) To implement view-invoice-xml request see </view-invoice-xml/>. receipt-id will be used to get invoice in XML format.
- (11) To implement view-invoice request see </view-invoice/>. receipt-id will be used to get invoice in PDF format.





5.30.3 Simple Bank Transfer Flow Statuses

Stage name	Description
AUTH_APPROVED	Invoice was issued for the Payer.
CANCEL_APPROVED	Payer didn't complete the transfer. This stage is optional and may be used for better transaction control in back-office.
CAPTURE_APPROVED	Funds were transferred to the Connecting Party account (settled).
VOID_APPROVED	Funds were returned by the Payer (re-called).
REVERSAL_APPROVED	Funds were returned to the Payer by Connecting Party.

5.30.4 Extended Bank Transfer Flow Statuses

Stage name	Description
AUTH_STARTING	Transaction was initiated.
AUTH_INVOICED	Invoice was issued for the Payer. It may be downloaded in back-office, sent to Payer by e-mail or requested via API.
AUTH_SENT	Payer informed the Connecting Party that invoice have been paid. This stage may be skipped.
CANCEL_APPROVED	Payer didn't complete the transfer. This stage is optional and may be used for better transaction control in back-office.
AUTH_APPROVED	Transfer was received by the bank.
CAPTURE_APPROVED	Funds were transferred to the Connecting Party account (settled).
VOID_APPROVED	Funds were returned by the Payer (re-called).
REVERSAL_APPROVED	Funds were returned to the Payer by Connecting Party.
CHARGEBACK_APPROVED	Funds were charged back by the Payer.

5.31 Mobile Device Sale

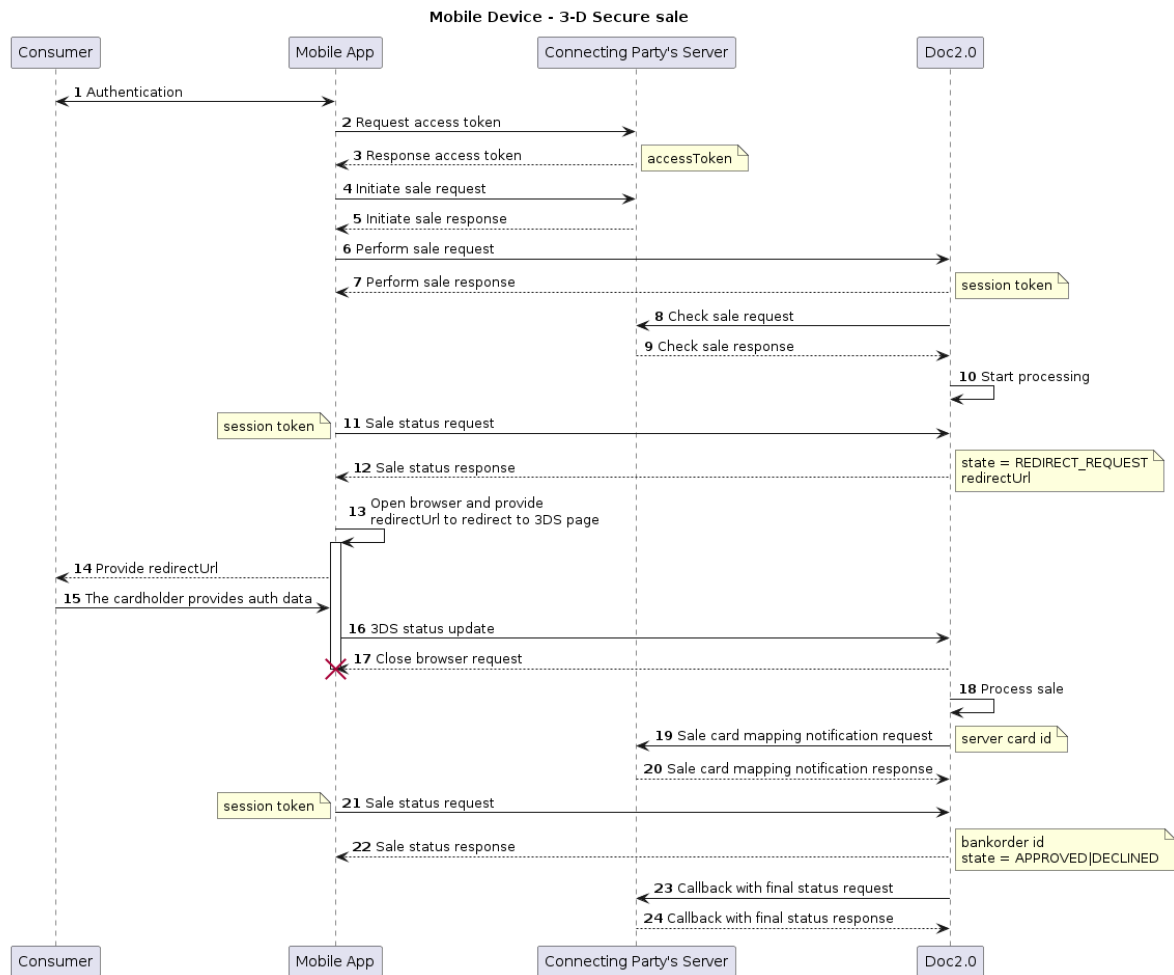
- [Introduction](#)
- [Sale Flow](#)
- [Repeat Sale Flow](#)

5.31.1 Introduction

Mobile device sale transaction can be performed with cardholder data or with card reference, previously made with [card verification](#) process or in previous transfer/sale transactions.

See terms definitions in [Glossary](#).

5.31.2 Sale Flow



(1,2,3) To perform authentication of Consumer in Connecting Party's app, Connecting Party can use any method which fits best to his needs. As a result, Connecting Party's server generates {accessToken} and provides it to Connecting Party's app. This parameter will be used to start and continue session.

(4,5) To initiate sale, Connecting Party's app sends {accessToken} with transaction amount and other device parameters to Connecting Party's server, which are used to start a session with unique random {nonce} and encrypted {signature}. To implement initiate sale request see [Initiate Sale](#).

(6,7) On this stage Connecting Party's app sends cardholder, device, session data and other parameters straight to Doc2.0 to perform sale transaction. To implement perform sale request see [Perform sale](#).

(8,9) Check sale is used for security purposes and allows Doc2.0 to compare the data sent by Connecting Party's app with the data stored on Connecting Party's server. To implement check sale request see [Check Sale](#).

(11,12,21,22) Sale status request is made by Connecting Party's app to Doc2.0 to get the status of sale transaction. To implement sale status request see [Sale status](#).

(19,20) Doc2.0 sends Sale card mapping notification request to Connecting Party's server/proxy with created on its side card reference - {serverCardId}. To implement sale card mapping notification request see [Sale card mapping notification](#).

(23,24) If Connecting Party callback URL is specified on endpoint level, Payment Gateway sends message to this callback URL whenever transaction reaches final status, no matter if the result is approved, declined or has other [final status](#). See more in [Callbacks](#).

5.31.3 Repeat Sale Flow

After successful Card mapping procedure, new sale transactions with the same cardholder data can be done easier for Consumer.

Connecting Party's app makes new sale requests using {clientCardId} instead of cardholder data. Doc2.0 sends this {clientCardId} to Connecting Party's server in "Check sale request" and gets mapped to it {serverCardId} in "Check sale response" from Connecting Party's server. This {serverCardId} is used to continue the processing of sale transaction.

Transaction flow remains the same: [Initiate Sale](#) -> [Perform sale](#) -> [Check Sale](#).

If there is no need to change Consumer's data (such as address, phone, etc.), "Perform sale request" for Repeat sale may be sent without any optional parameters. If Consumer's data has to be changed, new source card reference must be created.

If {clientCardId} is used in "Perform sale request", {serverCardId} must be included in "Check sale response" and signature.

If {clientCardId} was mapped to {serverCardId} in transaction with included {consumer.email} value, new "Perform sale request" and "Check sale response" with this {clientCardId} must also contain the same {consumer.email} value.

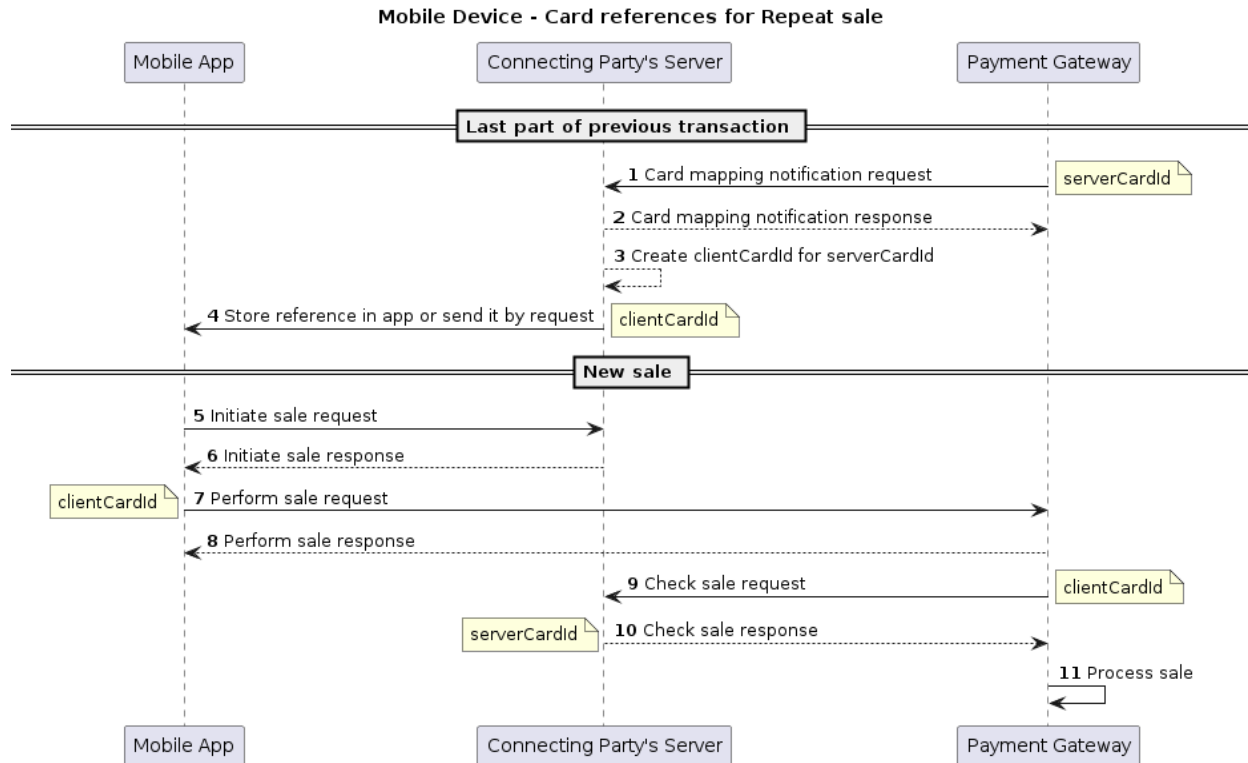
For Repeat Sale Flow use the following parameters in Perform Sale request:

1. {sourceOfFunds.reference.clientCardId}
2. {sourceOfFunds.reference.securityCode}

Instead of:

1. {sourceOfFunds.card.expiry.month}
2. {sourceOfFunds.card.expiry.year}
3. {sourceOfFunds.card.holder}
4. {sourceOfFunds.card.holder.firstName}

5. {sourceOfFunds.card.holder.lastName}
6. {sourceOfFunds.card.number}
7. {sourceOfFunds.card.securityCode}



(1,2) Doc2.0 sends Sale card mapping notification request to Connecting Party's server/proxy with created on its side card reference - {serverCardId}. To implement sale card mapping notification request see [Sale card mapping notification](#).

(5,6) To initiate sale, Connecting Party's app sends {accessToken} with transaction amount and other device parameters to Connecting Party's server, which are used to start a session with unique random {nonce} and encrypted {signature}. To implement initiate sale request see [Initiate Sale](#).

(7,8) On this stage Connecting Party's app sends cardholder, device, session data and other parameters straight to Doc2.0 to perform sale transaction. To implement perform sale request see [Perform sale](#).

(9,10) Check sale is used for security purposes and allows Doc2.0 to compare the data sent by Connecting Party's app with the data stored on Connecting Party's server. To implement check sale request see [Check Sale](#).

5.32 Mobile Device Transfer

- [Introduction](#)
- [Transfer Flow](#)
- [Repeat Transfer Flow](#)
- [Consumer Defined Transfer Rates Flow](#)

5.32.1 Introduction

Mobile device transfer can be performed with cardholder data of sender and receiver of funds or with card reference, previously made with [card verification](#) process or in previous transfer/sale transactions.

See terms definitions in [Glossary](#).

5.32.2 Transfer Flow

(1,2,3) To perform authentication of Consumer in Connecting Party's app, Connecting Party can use any method which fits best to their needs. As a result, Connecting Party's server generates {accessToken} and provides it to Connecting Party's app. This parameter will be used to start and continue session.

(4,5) To initiate funds transfer, Connecting Party's app sends {accessToken} with transaction amount and other device parameters to Connecting Party's server, which are used to start a session with unique random {nonce} and encrypted {signature}. To initiate transfer request see [Initiate Transfer](#).

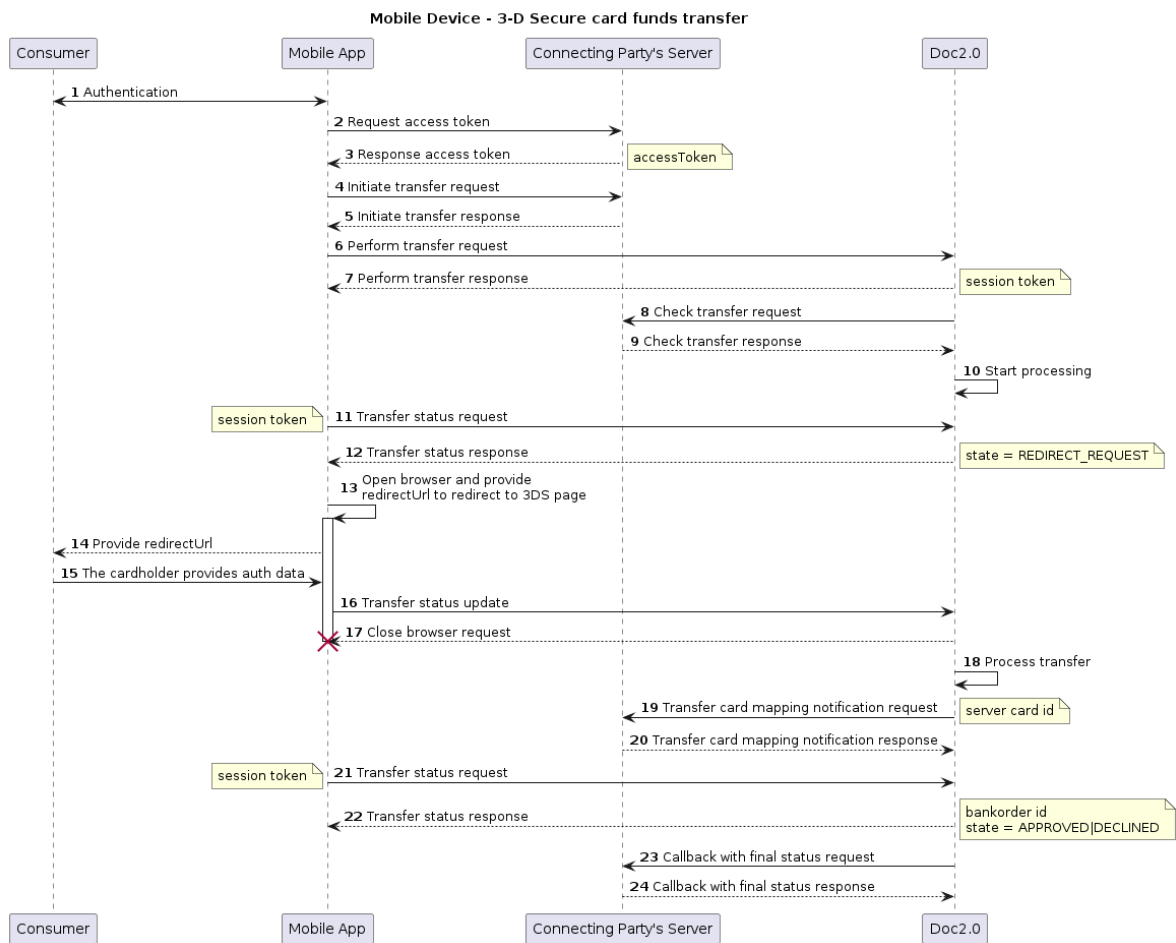
(6,7) On this stage Connecting Party's app sends cardholder, device, session data and other parameters straight to Doc2.0 to perform funds transfer from card to card. To implement perform transfer request see [Perform Transfer](#).

(8,9) Check transfer is used for security purposes and allows Doc2.0 to compare the data sent by Connecting Party's app with the data stored on Connecting Party's server. To implement check transfer request see [Check Transfer](#).

(11,12,21,22) Funds transfer status request is made by Connecting Party's app to Doc2.0 to get the status of transfer transaction. To implement transfer status request see [Status Transfer](#).

(19,20) Doc2.0 sends Transfer card mapping notification request to Connecting Party's server/proxy with created on its side card reference - {serverCardId}. To implement transfer card mapping notification request see [Transfer card mapping notification](#).

(23,24) Connecting Party specified a callback URL, Payment Gateway sends message to the callback URL whenever transaction reaches final status, no matter if the result is approved, declined or other [final status](#). See more [Callbacks](#).



5.32.3 Repeat Transfer Flow

After successful Card mapping procedure, new transfers with the same cardholder data can be done easier for Consumer.

Connecting Party's app makes new transfer requests using {clientCardId} instead of source and/or destination cardholder data. Doc2.0 sends these {clientCardId} to Connecting Party's server in "Check transfer request" and gets mapped to it {serverCardId} in "Check transfer response" from Connecting Party's server. These {serverCardId} are used to continue the processing of transfer transaction.

Transaction flow remains the same: [Initiate Transfer](#) -> [Perform Transfer](#) -> [Check Transfer](#).

If there is no need to change Consumer's data (such as address, phone, etc.), "Perform transfer request" for Repeat transfer may be sent without any optional parameters and with card references for sender and receiver instead of cardholder data. If Consumer's data has to be changed, new source card reference must be created.

If source and/or destination {clientCardId} are used in "Perform transfer request", source and/or destination {serverCardId} must be included in "Transfer check response" and signature.

If source {clientCardId} was mapped to {serverCardId} in transaction with included {consumer.email} value, new "Perform transfer request" and "Check signature response" with this {clientCardId} must also contain the same {consumer.email} value.

For Repeat Transfer Flow use the following source of funds parameters in Perform Transfer request:

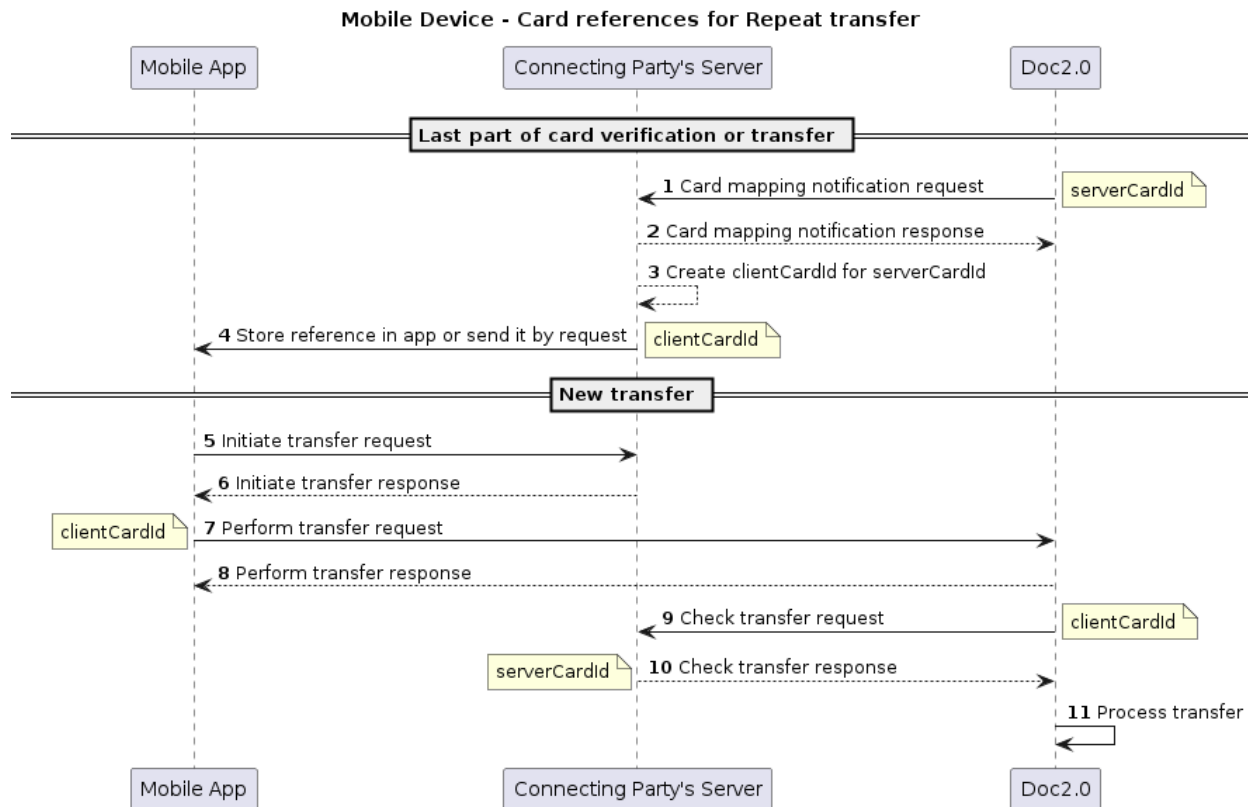
1. {sourceOfFunds.reference.clientCardId}
2. {sourceOfFunds.reference.securityCode}

Instead of:

1. {sourceOfFunds.card.expiry.month}
2. {sourceOfFunds.card.expiry.year}
3. {sourceOfFunds.card.holder}
4. {sourceOfFunds.card.holder.firstName}
5. {sourceOfFunds.card.holder.lastName}
6. {sourceOfFunds.card.number}
7. {sourceOfFunds.card.securityCode}

For destination of funds use {destinationOfFunds.reference.clientCardId} instead of {destinationOfFunds.card.number}.

(1,2) Doc2.0 sends Transfer card mapping notification request to Connecting Party's



server/proxy with created on its side card reference - {serverCardId}. To implement transfer card mapping notification request see [Transfer card mapping notification](#).

(5,6) To initiate funds transfer, Connecting Party's app sends {accessToken} with transaction amount and other device parameters to Connecting Party's server, which are used to start a session with unique random {nonce} and encrypted {signature}. To initiate transfer transfer request see [Initiate Transfer](#).

(7,8) On this stage Connecting Party's app sends cardholder, device, session data and other parameters straight to Doc2.0 to perform funds transfer from card to card. To implement perform transfer request see [Perform Transfer](#).

(9,10) Check transfer is used for security purposes and allows Doc2.0 to compare the data sent by Connecting Party's app with the data stored on Connecting Party's server. To implement check transfer request see [Check Transfer](#).

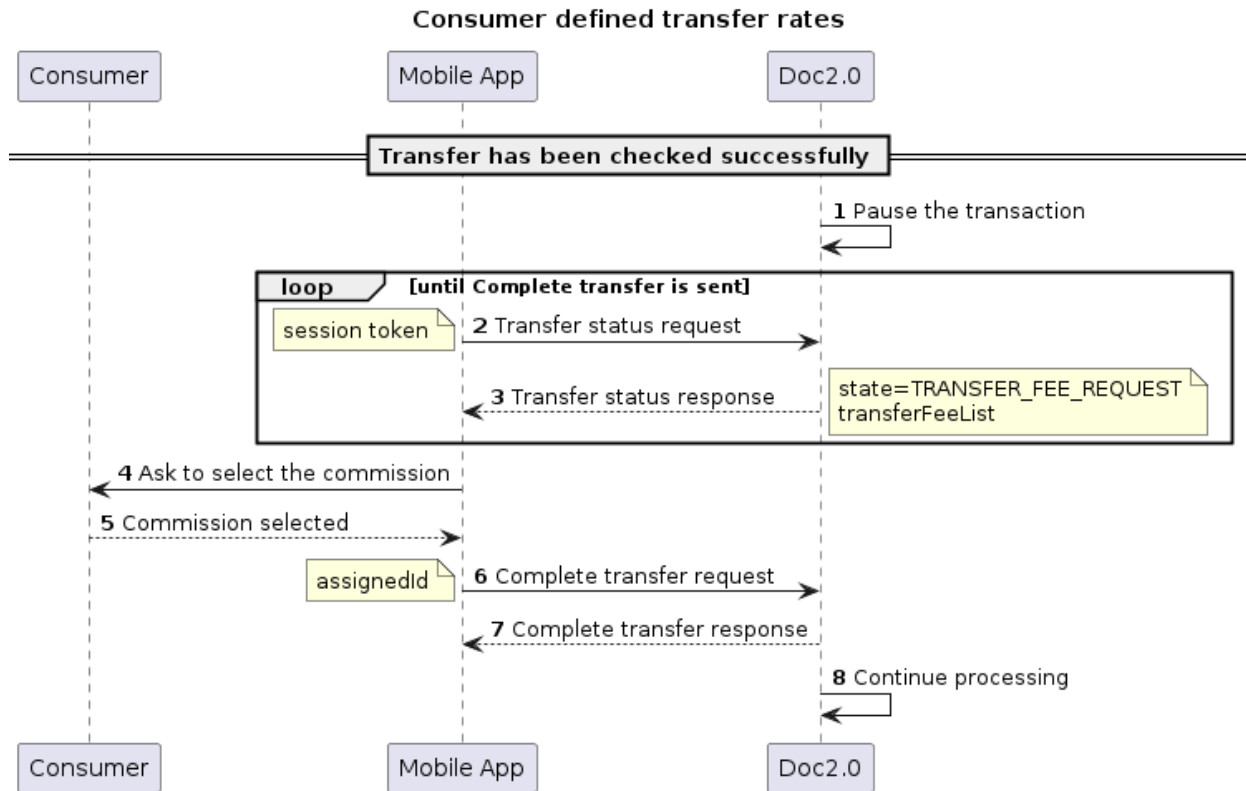
5.32.4 Consumer Defined Transfer Rates Flow

Consumer defined transfer rates is an optional feature which allows the sender to choose the most suitable commission amount for his transfer transaction between several available payment provider gates.

The possibility of such feature depends on the integration. Please ask support manager for details.

When such feature is enabled, the Transfer Flow extends:

1. If Check transfer has been successful, Doc2.0 pauses the processing, includes the list of possible commissions to next Transfer status response and awaits for confirmation from the Consumer.
2. Consumer selects the most suitable commission from the list in the Connecting Party's app.
3. Connecting Party's app sends the selected gate (using its assignedId aliased name) in Complete transfer request to Doc2.0.
4. Doc2.0 continues to process the transaction on the selected gate.



(2,3) Funds transfer status request is made by Connecting Party's app to Doc2.0 to get the status of transfer transaction. To implement transfer status request see [Status Transfer](#). Transfer status response includes the state parameter with TRANSFER_FEE_REQUEST value and the transferFeeList object with a list of commissions for possible processing gates. Each processing gate is represented by its own aliased name in assignedId parameter. TRANSFER_FEE_REQUEST value in state parameter exists only in "Consumer defined transfer rates" flow. The mentioned parameters may not be presented in first Transfer status response, so Connecting Party's app should continue polling the transaction status. When the transaction reaches TRANSFER_FEE_REQUEST state, the processing will remain paused until Complete transfer request is sent.

(6,7) To implement Complete transfer request see [Complete transfer](#).

5.33 Mobile Device Card Verification

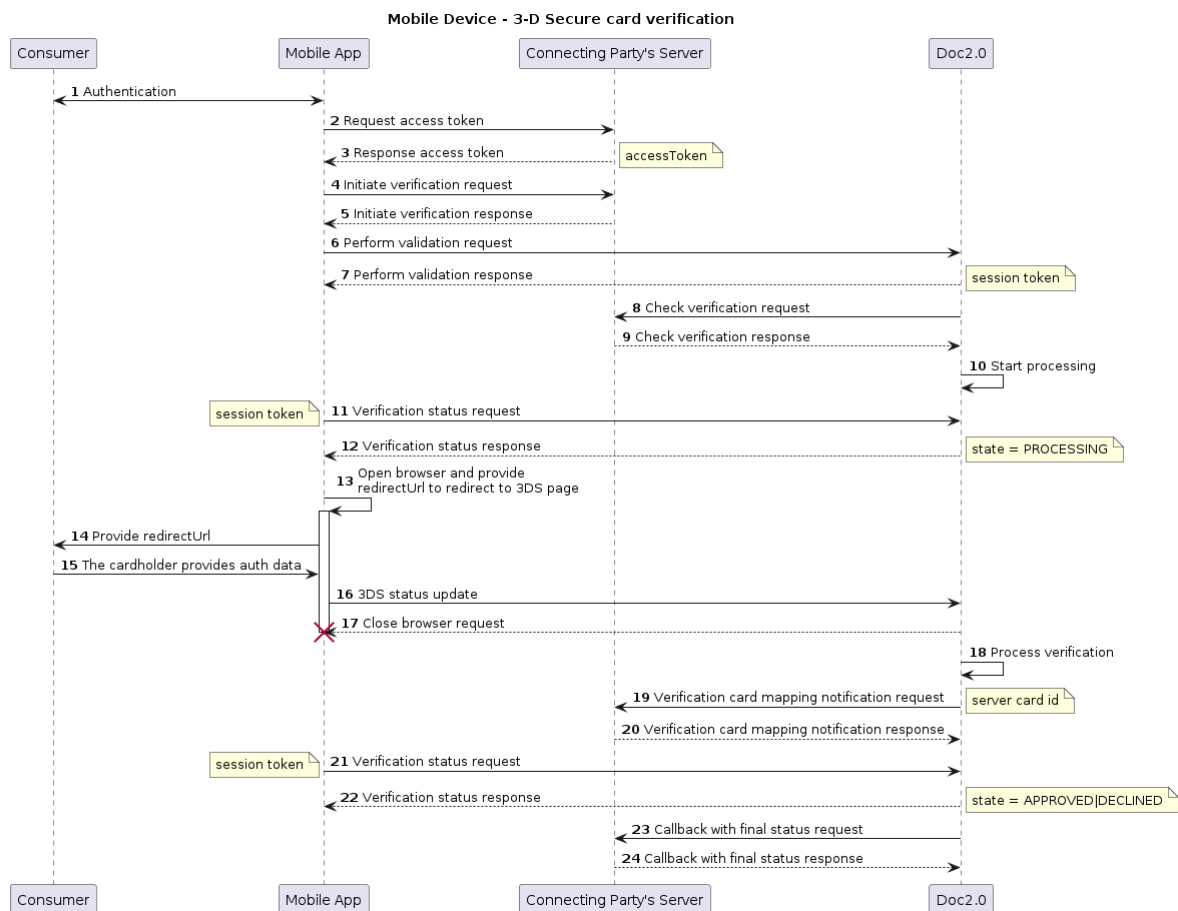
- [Introduction](#)
- [Mobile Device Card Verification Flow](#)

5.33.1 Introduction

Mobile Device Card Verification allows Consumer to make transfer and sale transactions easier with a single verification process. During the process, the card is being validated by Doc2.0, mapped to reference on Connecting Party's server. It allows Consumer to avoid entering card data for each transfer or sale transaction. Integration is based on HTTP POST requests/responses with JSON data.

See terms definitions in [Glossary](#).

5.33.2 Mobile Device Card Verification Flow



(1,2,3) To perform authentication of Consumer in Connecting Party's app, Connecting Party can use any method which fits best to his needs. As a result, Connecting Party's server generates {accessToken} and provides it to Connecting Party's app. This parameter will be used to start and continue session.

(4,5) To initiate card verification, Connecting Party's app sends {accessToken} with other device parameters to Connecting Party's server. To implement initiate verification request see [Initiate verification](#).

(6,7) Connecting Party's app sends cardholder, device, session data and other parameters straight to Doc2.0 to check the card for being valid. To implement perform validation request see [Perform validation](#).

(8,9) Check verification is used for security purposes and allows Doc2.0 to compare the data sent by Connecting Party's app with the data stored on Connecting Party's server. To implement check verification request see [Check verification](#).

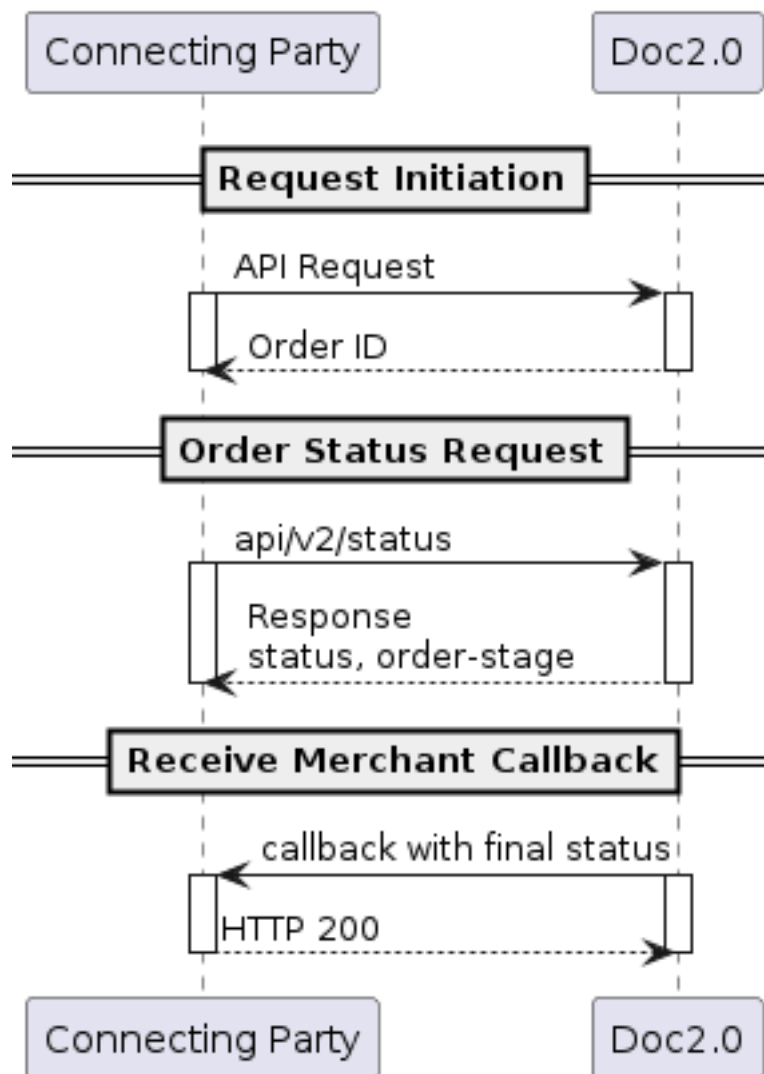
(11,12,21,22) Verification status request is made by Connecting Party's app to Doc2.0 to get the status of verification transaction. To implement verification status request see [Verification status](#).

(19,20) Doc2.0 sends Verification card mapping notification request to Connecting Party's server/proxy with created on its side card reference - {serverCardId}. To implement verification card mapping notification request see [Verification card mapping notification](#).

(23,24) If Connecting Party callback URL is specified on endpoint level, Payment Gateway sends message to this callback URL whenever transaction reaches final status, no matter if the result is approved, declined or has other [final status](#). See more in [Callbacks](#).

GENERAL API USAGE

6.1 Async-Response



Unless explicitly stated otherwise in relevant API command, all Doc2.0 responses are asynchronous, regardless of the request or callback sent.

6.2 Doc2.0 test scenarios

- [Transaction status test scenarios](#)
- [3DS 1.0.2 test scenarios](#)
 - [3DS 1.0.2 test cards](#)
 - [3DS 1.0.2 test amounts](#)
- [3DS 2.x.0 test scenarios](#)
 - [3DS 2.x.0 test cards](#)
 - [3DS 2.x.0 test amounts](#)

6.2.1 Transaction status test scenarios

Doc2.0 system does not provide special card numbers to pass the test transactions. Instead of it you can use any valid credit card number and specify card verification codes from the list to get the desired result.

Please note that specified credit card numbers and CVV2 codes will work only on sandbox environment and can't be used on production environment.

Next CVV2 codes can be used on sandbox environment to perform different answers from the system:

- 123 or 1234 to get an approved transaction;
- 321 for 3DS gates only, shows 3-D Secure validation form (hint is the correct 3-D Secure password to get approved transaction);
- 111 for 3DS gates only, immediately approves the transactions without 3-D Secure validation - emulates non-3DS flow for card that does not support 3-D Secure technology;
- 217 to get the transaction in unknown status (acquirer network problem emulation).
- 101 to get the transaction in decline status (expired card).
- 116 to get the transaction in decline status (not sufficient funds).
- 119 to get the transaction in decline status (Transaction not permitted to cardholder).
- 209 to get the transaction in decline status (Pick-up, stolen card).
- other CVV2 codes could be used to get the decline.

You can use following valid card numbers, if you haven't any or you want to check transactions for different payment systems.

On the sandbox.doc2.com you can use the test cards and amounts listed below.

6.2.2 3DS 1.0.2 test scenarios

3DS 1.0.2 test cards

Card Number	3DS Method
4444 0000 2222 4444	Present on server
4444 0000 1111 8888	Not present on server

3DS 1.0.2 test amounts

Amount	Status	ECI	Description
1.00	Y	00	Request successfully processed
1.01	Y	02	Request successfully processed
1.02	Y	05	Request successfully processed
1.03	Y	07	Request successfully processed
2.00	A	01	User not enrolled
2.01	A	06	Authentication was not available
3.00	N	05	Transaction declined by authorization system
3.01	N	07	User Authentication Failed
4.00	U	05	Validation failed
4.01	U	07	Technical Error in 3D system
Other			You can select behavior on challenge form

6.2.3 3DS 2.x.0 test scenarios

3DS 2.x.0 test cards

Card Number	3DS Method
4444 5555 6666 1111	Successful 3DS Method
4444 2222 3333 9999	Successful 3DS Method with 3 seconds delay
4444 1111 2222 3333	3DS Method with 15 seconds delay
4444 3333 1111 2222	No 3DS Method ACS URL

3DS 2.x.0 test amounts

Amount	Type	Status	Method data	ECI	Component	Description
0.02	Frictionless		Conditional		ARes	Invalid card
0.03	Challenge		Conditional		ARes	Success

continues on next page

Table 4 – continued from previous page

Amount	Type	Status	Method data	ECI	Component	Description
0.04	Frictionless		Conditional		ARes	Access denied, invalid endpoint
0.05	Challenge		Conditional		RReq	User press Cancel on ACS
0.06	Challenge		Conditional		CRes	Wrong threeDSServer-TransID and acsTransID
0.07	Challenge		Conditional		RReq	RReq = Success CRes = Fail
0.08	Challenge		Conditional		RReq	RReq = Fail CRes = Success
1.01	Frictionless	Y	Y	05	ARes	Successful frictionless authentication
1.02	Frictionless	Y	Y	02	ARes	Successful frictionless authentication
1.03	Frictionless	A	Y	06	ARes	Attempt
1.04	Frictionless	A	Y	01	ARes	Attempt
1.05	Frictionless	Y	N	05	ARes	Successful frictionless authentication
1.06	Frictionless	Y	N	02	ARes	Successful frictionless authentication
1.07	Frictionless	A	N	06	ARes	Attempt
1.08	Frictionless	A	N	01	ARes	Attempt
2.03	Frictionless	N	N	01	ARes	User not enrolled
2.04	Error					Card not participating
3.01	Challenge	C	Y		RReq	Challenge
3.02	Challenge	C	N		RReq	Challenge
2.01	Frictionless	U	N		ARes	Technical Error
2.02	Frictionless	N	N	06	ARes	User not enrolled
11.11	Challenge	Y	Conditional	02	RReq	Success
Other	Challenge		Conditional		RReq	You can select behavior on challenge form

Note: Conditional – bases on test cards for 3DS Method

6.3 Request Authentication Methods

6.3.1 SHA-1 Request Authentication Method

- [SHA1 Signature Generation](#)
- [SHA1 Request Generation](#)

To generate request with SHA-1 authentication:

1. [Generate Signature](#),
2. [Generate Request](#).

SHA1 Signature Generation

1. Concatenate all necessary parameters to String (also referred as a “signing string” or “signature base string”). Each API-command might have it’s own list and order of parameters to concatenate in signing string. Check it in the description of control parameter for the relevant API-command with SHA-1 authentication.

Note: Use minimal monetary units for amount (i.e. cent, penny etc. For amount of 0.94 USD value in signing string will be 94, for 10.15 USD value in signing string will be 1015)

For example, parameters:

```
endpointid=1111  
client_orderid=902B4FF5  
amount=10.42  
email=john.smith@gmail.com  
merchant_control=B17F59B4-A7DC-41B4-8FF9-37D986B43D20
```

will be concatenated to the string:

```
1111902B4FF51042john.smith@gmail.comB17F59B4-A7DC-41B4-8FF9-37D986B43D20
```

2. Sign the received string using SHA-1:

Java SHA-1 Generation Example:

```
package com.Doc2.0;  
  
import java.nio.charset.StandardCharsets;  
import java.security.MessageDigest;  
import java.security.NoSuchAlgorithmException;
```

(continues on next page)

(continued from previous page)

```

public class HashTextTest {
    /**
     * @param args
     * @throws NoSuchAlgorithmException
     */
    public static void main(String[] args) throws NoSuchAlgorithmException {
        System.out.println(sha1("test string to sha1"));
    }

    static String sha1(String input) throws NoSuchAlgorithmException {
        MessageDigest mDigest = MessageDigest.getInstance("SHA1");
        byte[] result = mDigest.digest(input.getBytes(StandardCharsets.UTF_8));
        StringBuilder sb = new StringBuilder();
        for (byte b : result) {
            sb.append(Integer.toString((b & 0xff) + 0x100, 16).substring(1));
        }
        return sb.toString();
    }
}

```

JavaScript SHA-1 Generation Example

```

<script src="http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/sha1.js"
  <script type="text/javascript">
    var sha1value = CryptoJS.SHA1("test string to sha1");
  </script>

```

Bash + openssl SHA-1 Generation Example

```
echo -n "test string to sha1" | openssl dgst -sha1
```

After that signature will look like this:

```
c6bdd88a78834ef4b863b088827a459f039e8257
```

SHA1 Request Generation

To make request use generated signature as the value of parameter control in request body.

Example of Request:

```

Request method: POST
Request URI: https://gate.doc2.com/paynet/api/v2/sale/1111
Body: client_orderid=902B4FF5&order_desc=Test Order Description&first_name=John&last_
  ↪ name=Smith&ssn=1267&birthday=19820115&address1=100 Main st&city=Seattle&state=WA&
  ↪ zip_code=98102&country=US&phone=+12063582043&cell_phone=+19023384543&amount=10.42&
  ↪ email=john.smith@gmail.com&currency=USD&ipaddress=65.153.12.232&site_url=https://
  ↪ doc2.codetime.net&credit_card_number=4538977399606732&card_printed_name=CARD HOLDER&
  ↪ expire_month=12&expire_year=2099&cvv2=123&purpose=user_account1&
  ↪ control=c6bdd88a78834ef4b863b088827a459f039e8257

```

Example of CURL Request:

```
curl --data "
  client_orderid=902B4FF5
  &order_desc=Test Order Description
  &first_name=John
  &last_name=Smith
  &ssn=1267
  &birthday=19820115
  &address1=100 Main st
  &city=Seattle
  &state=WA
  &zip_code=98102
  &country=US
  &phone=+12063582043
  &cell_phone=+19023384543
  &amount=10.42
  &email=john.smith@gmail.com
  &currency=USD
  &ipaddress=65.153.12.232
  &site_url=https://doc2.codetime.net
  &credit_card_number=4538977399606732
  &card_printed_name=CARD HOLDER
  &expire_month=12
  &expire_year=2099
  &cvv2=123
  &purpose=user_account1
  &control=c6bdd88a78834ef4b863b088827a459f039e8257
" https://gate.doc2.com/paynet/api/v2/sale/1111
```

6.3.2 OAuth Request Authentication Method

- [OAuth HMAC-SHA1](#)
 - [OAuth HMAC-SHA1 Signature Generation](#)
 - [OAuth HMAC-SHA1 Headers Generation](#)
 - [OAuth HMAC-SHA1 Request Generation](#)
- [OAuth RSA-SHA256](#)
 - [Generating Key Pair](#)
 - [OAuth RSA-SHA256 Signature Generation](#)
 - [OAuth RSA-SHA256 Headers Generation](#)
 - [OAuth RSA-SHA256 Request Generation](#)
 - [OAuth RSA-SHA256 Integration example for PHP](#)

OAuth HMAC-SHA1

To generate request with OAuth HMAC-SHA1 authentication:

1. [Generate Signature](#),
2. [Generate Headers](#),
3. [Generate Request](#).

OAuth HMAC-SHA1 Signature Generation

To prepare the Signature Base String do the following steps:

1. Gather all request body parameters and OAuth parameters (oauth_consumer_key, oauth_nonce, oauth_signature_method, oauth_timestamp, oauth_version):

For oauth_consumer_key use a merchant login provided by Payment Gateway, ex. merchantlogin.

For oauth_nonce generate unique random nonce, ex. y3qlvMPky7g.

For oauth_signature_method set HMAC-SHA1.

For oauth_timestamp header get current timestamp in seconds, ex. 1669966913.

For oauth_version set 1.0.

Parameter oauth_signature is not included in request body parameters (only in OAuth header).

2. Percent-encode value of each gathered on step 1 parameter. See [RFC 3986](#)¹¹ for reference. The unreserved characters can be encoded, but should not be encoded. The reserved characters have to be encoded, for example: % is encoded to %25, / is encoded to %2F, = is encoded to %3D.
3. Sort all gathered on step 1 parameters together with their values in lexicographical order by parameter name.
4. Concatenate all gathered on step 1 parameters and their values using & and = in the following way:
object1=value0f0bject1&object2=value0f0bject2...

Concatenated parameters string example:

```
client-order-id=1234567890&oauth_consumer_key=merchantlogin&oauth_
↪nonce=y3qlvMPky7g&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1669966913&
↪oauth_version=1.0&sending-card-ref-id=7654321
```

5. Percent-encode each element (POST, URL, concatenated parameters) and construct Signature Base String with POST&URL&parameters structure (as documented in OAuth [A.5.1](#)¹²).

¹¹ <https://www.rfc-editor.org/rfc/rfc3986>

¹² https://oauth.net/core/1.0a/#sig_base_example

Signature Base String Example:

```
POST&https%3A%2F%2Fgate.doc2.com%2Fpaynet%2Fapi%2Fv2%2Fpan-eligibility%2Fsend
↳%2FENDPOINTID&client-order-id%3D1234567890%26oauth_consumer_key%3Dmerchantlogin
↳%26oauth_nonce%3Dy3qlvMPky7g%26oauth_signature_method%3DHMAC-SHA1%26oauth_
↳timestamp%3D1669966913%26oauth_version%3D1.0%26sending-card-ref-id%3D7654321
```

6. Sign the Signature Base String with HMAC-SHA1. The signing key is the concatenation of consumer secret (Merchant control key) + & + token secret (empty string).

HMAC-SHA1 Key Example:

```
11111111-1111-1111-1111-111111111111&
```

7. Encode the signed string to Base64 to get signature for API.

Java HMAC-SHA1 Signature Generation Example:

```
package com.Doc2.0;

import org.apache.commons.codec.binary.Base64;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;

public class HMAC_SHA1 {
    private static final String HMAC_SHA1_ALGORITHM = "HmacSHA1";

    public static String calculateHMAC(String data, String key) throws
↳NoSuchAlgorithmException, InvalidKeyException {
        SecretKeySpec signingKey = new SecretKeySpec(key.getBytes(), HMAC_SHA1_
↳ALGORITHM);
        Mac mac = Mac.getInstance(HMAC_SHA1_ALGORITHM);
        mac.init(signingKey);
        return Base64.encodeBase64String(mac.doFinal(data.getBytes()));
    }
}
```

Signature will look like following string:

```
d/IPlITUmPcniwjA7Vckjr6WQeE=
```

OAuth HMAC-SHA1 Headers Generation

A request with OAuth HMAC-SHA1 authentication method must contain next headers:

- content-type=application/x-www-form-urlencoded
- Authorization: OAuth

OAuth header must contain the same OAuth parameters and values which were included in signature base string (oauth_consumer_key, oauth_nonce, oauth_signature_method, oauth_timestamp, oauth_version). Add signature as value of parameter oauth_signature to header Authorization. All values of parameters in OAuth header must be in quotes (See OAuth 5.4.1.¹³) with the following structure: parameter="value" and be separated by mandatory comma (,) and optional space. OAuth realm parameter is optional.

Authorization Header Example (Header is line-separated for display purposes):

```
Authorization: OAuth
oauth_consumer_key="merchantlogin",
oauth_nonce="y3qlvMPky7g",
oauth_signature="d/IpLITUmPcniwjA7Vckjr6WQeE=",
oauth_signature_method="HMAC-SHA1",
oauth_timestamp="1669966913",
oauth_version="1.0"
```

Note: Since headers are not usually encoded, the values of oauth parameters (especially oauth_nonce and oauth_signature) should be percent-encoded separately before being added to Authorization header.

OAuth HMAC-SHA1 Request Generation

1. Use headers from previous section,
2. Add request parameters to body,
3. Percent-encode and send the request.

Example of Request:

```
Request method:      POST
Request URI: https://gate.doc2.com/paynet/api/v2/pan-eligibility/send/1111
Headers: Authorization=OAuth realm="",oauth_version="1.0",oauth_consumer_key=
↪ "merchantlogin",oauth_timestamp="1669966913",oauth_nonce="y3qlvMPky7g",oauth_
↪ signature_method="HMAC-SHA1",oauth_signature="d%2FIpLITUmPcniwjA7Vckjr6WQeE%3D"
Accept= */*
Content-Type=application/x-www-form-urlencoded; charset=ISO-8859-1
Body: client-order-id=1234567890&oauth_consumer_key=merchantlogin&oauth_
↪ nonce=y3qlvMPky7g&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1669966913&oauth_
↪ version=1.0&sending-card-ref-id=7654321
```

Example of CURL Request:

¹³ https://oauth.net/core/1.0a/#auth_header

```
curl -H '
  Authorization: oauth_version="1.0",
  oauth_consumer_key="merchantlogin",
  oauth_timestamp="1669966913",
  oauth_nonce="y3qlvMPky7g",
  oauth_signature_method="HMAC-SHA1",
  oauth_signature="d%2FIPlITUmPcniwjA7Vckjr6WQeE%3D"
' --data '
client-order-id=1234567890
&oauth_consumer_key=merchantlogin
&oauth_nonce=y3qlvMPky7g
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1669966913
&oauth_version=1.0
&sending-card-ref-id=7654321
' 'https://gate.doc2.com/paynet/api/v2/pan-eligibility/send/ENDPOINTID'
```

OAuth RSA-SHA256

To generate request with OAuth RSA-SHA256 authentication:

1. [Generate Key Pair](#),
2. [Generate Signature](#),
3. [Generate Headers](#),
4. [Generate Request](#).

Generating Key Pair

To authorize future requests with RSA-SHA256, generate the key pair:

- A PRIVATE key to sign the request. This private key must be kept secret from everyone.
- A PUBLIC key to verify that the request was signed with the respective private key. Send it to support manager.

To generate keys download latest version of [openssl](#)¹⁴ and run following commands:

```
openssl genpkey -algorithm RSA -out private_key_pkcs_8.pem -pkeyopt rsa_keygen_
bits:4096
openssl rsa -pubout -in private_key_pkcs_8.pem -out public_key.pem
```

Use different keys for production and for testing purposes to avoid compromise.

To use request builders in documentation, get the private key in PKCS#1 container. It should be unencrypted RSA private key in PKCS#1 PEM text format. To get it use the following command:

```
openssl rsa -in private_key_pkcs_8.pem -out private_key_pkcs_1.pem
```

For OpenSSL v3+ (tested on v3.0.5), use the following command instead:

¹⁴ <https://www.openssl.org/>

```
openssl rsa -traditional -in private_key_pkcs_8.pem -out private_key_pkcs_1.pem
```

The resultant PKCS#1 RSA private key will start with `-----BEGIN RSA PRIVATE KEY-----`. For production purposes any format of the key supported by Connecting Party software can be used. Request builders in documentation support key length up to 4096. Example of the PKCS#1 RSA private key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIJKQIBAAKCAgEA16QK2iWgYUbmR2GqSbaS0PQZKF2DkstSj0dakW+hASTz5Ams
R5sDnurfeR4m+Htaxiv69MMdvoDLuCmZE8KQzsEOZovZ9UYSh9CKK4/FzQSZ8ZDP
8cpKLN7/gitWiM14iuC9Pi74TTLeg7PuGjeoc0jUs0WMf7sV6uzfZwvqYgUVRljY
gscwDRiTSGJQumQtanCs/LMIkxouThLztSSEmHhhEz2aW0omqR5hH0+HJ4I1AfET
V7VpKJ34c1+zMesDfpDxZ8VpQpno9iikFG64MigDFmBeskI6q15tBwbR0YSfqNEmG
LwhYQ+SXnojeuazkSJ45CeQRh6dn3GgD7kex2N3lK97qqDcW0Lqcsbe+ZyTGALn
WGzTZWjle0+yrdE6awD34kUYVnzD/9WvdYqpH2pXBDq0IXu6l4m4gLe5pKTRiFEc+
TjgVb34tJGEERkrvqktSEmRQzMgZQnZk/5//7+csUicSPmqdUn5oB6ngVueZkk7v
wtL6dcCxr5isWgXQE0+oYbt72Ns5RjLVWiXwV2ZNFd+iR406+etBxYNz+mg/2B5c
P08NWYvvFlaBUu415GG1XntBGWncKQiZ49WCvLcYebSfUEkWLj6zqJaDS/buT3jU
rWQ0WEI8G1HnTQp0cqmx9WpXDLx4n3ytRjHuHe3ND9AYE28yhFFY5baIU3UCAwEA
AQKCAgAC4QRQD0TFx7c15DzszQY6yfeIBW+bRyGsDgzUgkQJCuBCvCpTrmsm9QXU
zSVCDguRN8ca+3vrLjckF2wWynM6f3NcxSM8lhmRPIqLuFiwuW3/HqrYFJZW8QdC
SqfWHcAtQoDkUqY4CaTU51MXgIS8PU2xsw0EK5BIWa9F5e/ULTMyhD8nx9cJZbmZ
rs5bHrLIgYadvRoxNJlHq5MbaQhoLLtHEXx9EWtAuModI8mPKnrgssJKWn6z7yB9
dYjpXqfdvnyI72bCQkG0FaweyX0bXpVEyZQhPfZj+IuxNWIShADpf83N1P0wvF2V
3Ugp0bgejBZA3o2pXP/S/oSG6ugh8dZHfa8vkw0x5N28393IzIMpzwE2EnsBidN1
ca7NwDpmpUyuULSpi3YoViUYy1i4Mwngv2XQdkbvoGusQwgWoNrppmDKxLL5qEBf
lIPCZAgZSR79KHYw2V0zkm84hu0jDXMthpt9A2gkLkRhGnGgb4n5KzyCpY9iuFK2w
C05Fdlx0XjRLZb7G1JCeU6Qh0kjSE7seh9ltyo+VsW0Lx4UwV0YGCMAF45y00wF
/MJdYotUtlvC5G/DK8iftTtjwb/xPLGDiC441T0ReWwVf6n36+shb6szLI2EmqKBkp
2Sr5xQ5VNZkcg2W/BUf7+n8Rvisu17TyW0HmwDEBdfJQzgzgPQKCAQEA7RpBNVBH
Jx8gR7hQdbyi4/6U5KiYorkzuoK4KkjRWJfqJvp9uGZS2vDw0oIC8kCAXMwu30CI
U0xkDvH7bb/qedn0IG7+72FUCKlxqkMk4lv03zE9yUPcYNT+w573uh+rXQ/mcsFF
+aBtupRZiDqqd4vuvjTjpw5Q4tyk/lxZfbe10S2NyxY4dZsbm8gl0SypLs/rLYjZ
8ZntRpZozIwoenrF3AnvtR114WBDpBVwSJ9Knd8xB5Fufc9TqsZ/EKPjDVRn2Sq2
Lt/xKSopwxPyIhKG1zmAeYhv8Q+GYU0QYCFBj2opDC3A0xANw2j9M8nYjCMDmPaP
5iDCUla35srp/wKCAQEA6NPj8auPGGFen2ZJoydpEPKqU3zAdv05VLKVIvBA9c+Y
oy7zNhnNw0PCckYpB9jPGvpdn6KFh2ZTU/mgmIysKriLcLN4gKho7JUCU3kvglmv
zJiz/5fR0xCCRNPAAH6uJ+CxyssjbUoe9EmyVxKX3l2Zkmy1z0KRc/FbAkql07
ItDReryb64IjsfT4GtU4nBK7zCzI+yya1BjL/McnGBcpKIwp9HCwaTQK7yxa7ThY
TsfTuxoyZM1/xZE0ckRJGVLtkao1Vf0y0SDdCp+RwtBvVmt3Wt6vVcl6qG0LW5Fe
Uz0PN+CebMfhBCaqWXIXeuMUo+RdLnGn113Tl6i6iwKCAQEAz+NzRUGMAXtDHF84
/OJWmD1BY30H0TU9a8ztmPWbyGf6gA6laKcfAqS6nTIdTzbK1ZKZjES6gv65xHjb
ERFyj0BQ0pc/o7fcrH0VG8ofbvFdtMxB9lQvyB84+WBKqMDXyZMFZZyctBC75Rnp
no6BpKvmupM+LZZJyX/YksV6GcaX/j5I0sY63rM08/n7XnogJNfCz0Hu5e0mo/uB
C8ItRKAdER8NM+o0z3t0E3JQrvwrXyzAmngjPuAn5daA1qA7lhwcqMbQUi08D/HO
CCNW7BT+cXsTchv2WpBYLLPGxOhWYF42e10p7R9YUfud9miGG+kfYGDfLtG0UA+E
0zEbFQKCAQEA4lczDnqolpv5394RkiG6+zXTdLYfaM2NUwTfZ0ka9xxEl8cJuztk
lAIogggj1HcKB4EDSTA2vUVVlppjbEm9CZ70N7DRYcnWjr/hTgLOlN04mp6Mxdny
qkwvR/fZLf8bzrs2qcRhIrM5DN/NA0Jn+10f+nMIQUTMSpgFxpDDbDe0SI1WTApV
TaLrTpIGLbfCe7+ef8098qgPMEew7vswXzQM2BVCqBZw+SUVyC0HlXukJZoPlKHI
AcThBNC/eQ3M3miG+YfNZ+yMls9q82viyM/WnN3GXzmCnE37XYb8dp0gZK1EQR8F
BF1fu6hXDLNkbhuZsiZMC92DvFPDYnkuNwKCAQA6/2K8PLl0eK+0p/IGVsgJpgHn
Uh3BehVKHXeG/Buhn5bMXX3cB2hEHg2tz4pw3JxfZlUflhyhKD43XnpXuMmt81Ka
Ja5MeXDg0kfnlXoLVA4ezx2V2EohMExUykk0IfQBDTaNTjsg5PB4HLKFid3kJ6u/
JCXuy0EA07vL/kNl+cDEBLJsVtvtxHLdpdJh01P0i3IIgOpdd0+a/0/GDsdlAWog
hyEb6r7+bWurjw0YjHX+R5ZQ+0XtnzXU20d2NiP/oH2IvQzXRUQ1U17KzZn5PAhs
YC7r9lRV4VjbhEi3Zk2FBPrrzs2ieXo5aHXCnzFywQ99nlrz0Ic8vV16WR1x
```

(continues on next page)

(continued from previous page)

-----END RSA PRIVATE KEY-----

OAuth RSA-SHA256 Signature Generation

To prepare the Signature Base String do the following steps:

1. Gather all request body parameters and OAuth parameters (oauth_consumer_key, oauth_nonce, oauth_signature_method, oauth_timestamp, oauth_version):

For oauth_consumer_key use a merchant login provided by Payment Gateway, ex. paydroid.

For oauth_nonce generate unique random nonce, ex. hoFlZri9c17X1Tvb7yD2fsMEQUIWBQ3m.

For oauth_signature_method set RSA-SHA256.

For oauth_timestamp header get current timestamp in seconds, ex. 1669720957.

For oauth_version set 1.0.

Parameter oauth_signature is not included in request body parameters (only in OAuth header).

2. Percent-encode value of each gathered on step 1 parameter. See [RFC 3986](https://www.rfc-editor.org/rfc/rfc3986)¹⁵ for reference. The unreserved characters can be encoded, but should not be encoded. The reserved characters have to be encoded, for example: % is encoded to %25, / is encoded to %2F, = is encoded to %3D.
3. Sort all gathered on step 1 parameters together with their values in lexicographical order by parameter name.
4. Concatenate all gathered on step 1 parameters and their values using & and = by the following way: object1=valueOfobject1&object2=valueOfobject2...
5. Percent-encode each element (POST, URL, concatenated parameters) and construct Signature Base String with POST&URL&parameters structure (as documented in OAuth [A.5.1](https://oauth.net/core/1.0a/#sig_base_example)¹⁶).

Signature Base String Example:

```
POST&https%3A%2F%2Fgate.doc2.com%2Fpaynet%2Fapi%2Fv4%2Ftransfer%2FENDPOINTID&
↪amount%3D10.42%26card_printed_name%3DJohn%2520Doe%26card_recurring_payment_id
↪%3D42322%26client_orderid%3D1%26credit_card_number%3D4210708776705721%26currency
↪%3DUSD%26cvv2%3D123%26deposit2card%3Dfalse%26destination-card-no
↪%3D4232618181101636%26destination_card_recurring_payment_id%3D61622%26expire_
↪month%3D12%26expire_year%3D2099%26ipaddress%3D1.1.1.1%26oauth_consumer_key
↪%3Dpaydroid%26oauth_nonce%3DhoFlZri9c17X1Tvb7yD2fsMEQUIWBQ3m%26oauth_signature_
↪method%3DRSA-SHA256%26oauth_timestamp%3D1669720957%26oauth_version%3D1.0
↪%26order_desc%3DYour%2520order%2520description%26redirect_url%3Dhttp%253A%252F
↪%252Fwww.example.com
```

6. Sign the Signature Base String with RSA-SHA256 using the private key obtained in previous section.

¹⁵ <https://www.rfc-editor.org/rfc/rfc3986>

¹⁶ https://oauth.net/core/1.0a/#sig_base_example

7. Base64 encode the signed string to get signature for API.

Base64 Encoded Signature Example:

```
K0hLc7GYh65UDTNvJvJbqoD95T7ekVEwIx+AxlBe2rNdPVzCAZMTi58J5pJlXZA1qOrgzUj/
↪uL764NofP6qrqBXHX9Fpg+PdoMBey7zY9nM0mdpHhkwyqA0n8e8oh68x+8RtC1+gmaIsIJDVurpCm2CdaViC2ny90GWP
↪Y9buuyvdKSNsao7qHwD/6bb9Sj2MFFhYP6gHf+p1NipH05224aX9hMZty80vb3+ps/
↪wNwYC8NVfft5bh8ETowaHQ5TU0PFdvU+5IkH0nnfbvz1/+jDeErd0PdQ4xH6c5/
↪gZVQY9j6DFpazsw8d3karwXoBduv7I3mh7L3CSjTweABaRMw=
```

OAuth RSA-SHA256 Headers Generation

A request with OAuth RSA-SHA256 authentication method must contain next headers:

- content-type=application/x-www-form-urlencoded
- Authorization: OAuth

OAuth header must contain the same OAuth parameters and values which were included in signature base string (oauth_consumer_key, oauth_nonce, oauth_signature_method, oauth_timestamp, oauth_version). Add signature as value of parameter oauth_signature to header Authorization. All values of parameters in OAuth header must be in quotes (See OAuth 5.4.1.¹⁷) with the following structure: parameter="value" and be separated by mandatory comma (,) and optional space. OAuth realm parameter is optional.

Authorization Header Example (Header is line-separated for display purposes):

```
Authorization: OAuth
oauth_consumer_key="paydroid",
oauth_nonce="hoFlZri9c17X1Tvb7yD2fsMEQUIWBQ3m",
oauth_signature="K0hLc7GYh65UDTNvJvJbqoD95T7ekVEwIx
↪%2BAxlBe2rNdPVzCAZMTi58J5pJlXZA1qOrgzUj%2FuL764NofP6qrqBXHX9Fpg
↪%2BPdoMBey7zY9nM0mdpHhkwyqA0n8e8oh68x%2B8RtC1
↪%2BgmaIsIJDVurpCm2CdaViC2ny90GWPrrSin9CFwDmIKBt0J7dxNnuFQJkvLxwK9JE9gRfQssG4v0rXrn2f5DvENFvFW
↪%2BtD2
↪%2Bp2rMDDU7HqJUGy764rb4ShuvuiEvzIaNwg3JAxho9fqckJz5LXt4efX1i8oFt33ztYSgZRojsow4HCzuhZcQqMRepx
↪%2B47cJ0KujEHuDXR0blq3o9Uos5K%2BMu9Carjs8jHMiBHo4aS4IagXVY3mEuohCuRhL9
↪%2FY9buuyvdKSNsao7qHwD%2F6bb9Sj2MFFhYP6gHf%2Bp1NipH05224aX9hMZty80vb3%2Bps
↪%2FwNwYC8NVfft5bh8ETowaHQ5TU0PFdvU%2B5IkH0nnfbvz1%2F%2BjDeErd0PdQ4xH6c5
↪%2FgZVQY9j6DFpazsw8d3karwXoBduv7I3mh7L3CSjTweABaRMw%3D",
oauth_signature_method="RSA-SHA256",
oauth_timestamp="1669720957",
oauth_version="1.0"
```

Note: Since headers are not usually encoded, the values of oauth parameters (especially oauth_nonce and oauth_signature) should be percent-encoded separately before being added to Authorization header.

¹⁷ https://oauth.net/core/1.0a/#auth_header

OAuth RSA-SHA256 Request Generation

To make the request:

1. Use headers from previous section,
2. Add request parameters to body,
3. Percent-encode and send the request.

Example of Request:

```
Request method: POST
Request URI: https://gate.doc2.com/paynet/api/v4/transfer/4473
Headers: Authorization= OAuth oauth_nonce="C12mCYiI4RDeVTvyZDHS3f0vUbBNrK9G", oauth_
signature="S4ahUgep8EV3RTIPdcp0
%2FzEwB8V1XwFcGi5zhrYKySPVVA2PAY7AxezPc0CMhMwZQfcf8V0H809v5zZ
%2BmYV2Qsq4kjPe1zEJIjddjhI
%2B2MX9VW8dWn9DyoTD2lk0YUwGsCteXU6mwGtNergN5KwGTJgqYPfzWFLlLSAhGwu0d
%2FgHYVRnA6jd4FvywRaRiSsnzgsasGJjWCGzX%2F1B8L78H
%2FqD1W7dNJHjC2JbonUiHVT4a0s74ppqYE9zcb5rK8PH1l2ems3ArXhqFaSF85%2BYzJ
%2BkSeDswqeIZ1y3NS8XPrmaLeymLNqKp0fAl9Ng47AmIjSYEw3s5fQ7Xi9t4j7Y6fQA4RnIzIdmH4oeMiyzn7dpA87wqnXm5A
%2F04dBGDRw4LP1aEaJktQRmT5xwyoaQfz
%2Bh2MR7zDpRbVZDpUto1iYlQ15Ulgoj0HaLLjW1gqggbvrtVXUBT73KcIboWw00VUcbVUX5Yb
%2FSu7hUZ02fNAh4LYrsVcXRJxCZuwhXCiqvbr3EziEFDGVEZnKUDLCodzHbVA9hbpXHG7TvSLFEANCPDFtapMF9eo
%2F1y0t5Tkxag4g5yNes93lrVabsTqckiFiFKR2R3CcwJ6MQtMfo8ticJcKxfo2v%2BTkC0bEaRkBGk%3D",
oauth_consumer_key="ny_qa_merchant", oauth_signature_method="RSA-SHA256", oauth_
timestamp="1669892964", oauth_version="1.0"
Accept= */*
Content-Type=application/x-www-form-urlencoded; charset=ISO-8859-1
Body: cvv2=123&ipaddress=1.1.1.1&amount=15.42&credit_card_number=4210708776705721&
client_orderid=1&deposit2card=false&destination-card-no=42326181101636&expire_
month=12&order_desc=Your%20order%20description&card_printed_name=Vasia%20Pupkin&
currency=RUB&expire_year=2099&redirect_url=http%3A%2F%2Fwww.example.com
```

Example of CURL Request:

```
curl -H '
  Authorization: OAuth oauth_consumer_key="paydroid",
  oauth_nonce="JOARJWjXJgFRUVqig8HiS0ouFWKfK4N5",
  oauth_signature="k5Z0XCvDvb5h873XG6TDM0854PsuueSnSby4h0%2F3j4TKraY6ebjoFfDmndI09
%2FQp6uEbCgKregNY5N0ccIVIay49l6v7jMdIFefqU7E5eu%2BIJcoqG7kMFcdCu29hweYx7p4ZSk
%2FUtdGLN3wyUUybCAx73XyO0tkZteAleyzQlzdPzQ99vPS8FN2WmbNdzu3H2PLf0X0Zy4DbAPleZGfu3GwXxe9erGsvzBJoz
%2BV3qiG3jc7tdQGRZP%2BLZFw3nihNlW%2F2Nlp%2FId6QG8kPN0Ex2GxAuQa8kufv
%2BbpohU8UftZG1SnNfoa4nDVgNbWoSbTXbgXxwHE0ZlccwT6q
%2BHBtmgggVIdGN9JuuLUL8fDSCxqu3R7YMvcwArzrkd1XEFUFvYwXhc3QpfBu0GyLkL7pnmz
%2BFPTH48C0Z7yXbK6nQDFSiy8lmaEJsnnjMYyMMVgJNkDNzqwNx0QuwprTZ7KzHqWrTj0zHT0Hl44q1pJhIqMwxWDPmsziExY
%2BK%2Bu3M7tvG5Foy7Vwj
%2BDSMPMAhj7j2AmUG4HzaJcpiBWMq0CkGWRjvne0k3NiwkEw0xq5tJ0FjsroditUqJQSX1PTqf
%2FtuiqaE2Gt5EYL19ZzyXbxtMjGZwL%2BzFlct6ftTi4%3D",
  oauth_signature_method="RSA-SHA256",
  oauth_timestamp="1669892687",
  oauth_version="1.0"
' --data '
amount=10.42
&card_printed_name=John%20Doe
&card_recurring_payment_id=42322
&client_orderid=1
&credit_card_number=4210708776705721
```

(continues on next page)

(continued from previous page)

```

&currency=USD
&cvv2=123
&deposit2card=false
&destination-card-no=4232618181101636
&destination_card_recurring_payment_id=61622
&expire_month=12
&expire_year=2099
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
' https://gate.doc2.com/paynet/api/v4/transfer/ENDPOINTID '

```

OAuth RSA-SHA256 Integration example for PHP

```

<?php
/**
 * OAuth RSA-SHA256 Integration Example for PaynetEasy API
 * This example demonstrates a simplified single-file implementation
 */

// Include the OAuth library (tmhOAuth)
class tmhOAuth {
    const VERSION = '0.8.5';
    var $response = array();

    /**
     * Creates a new tmhOAuth object
     *
     * @param string $config, the configuration to use for this request
     * @return void
     */
    public function __construct($config=array()) {
        $this->buffer = null;
        $this->reconfigure($config);
        $this->reset_request_settings();
        $this->set_user_agent();
    }

    public function reconfigure($config=array()) {
        // default configuration options
        $this->config = array_merge(
            array(
                // leave 'user_agent' blank for default, otherwise set this to
                // something that clearly identifies your app
                'user_agent'          => '',
                'host'                => 'api.twitter.com',
                'method'              => 'GET',

                'consumer_key'        => '',
                'consumer_secret'     => '',
                'token'               => '',
                'secret'              => '',

                // RSA private key (for RSA-SHA1 and RSA-SHA256 methods)

```

(continues on next page)

(continued from previous page)

```

// Please note that this is expected to be a string representing
// the PEM-formatted key itself and NOT the file name
'private_key_pem'          => '',

// OAuth2 bearer token. This should already be URL encoded
'bearer'                   => '',

// oauth signing variables that are not dynamic
'oauth_version'            => '1.0',
'oauth_signature_method'   => 'HMAC-SHA1',

// you probably don't want to change any of these curl values
'curl_http_version'        => CURL_HTTP_VERSION_1_1,
'curl_connecttimeout'      => 30,
'curl_timeout'            => 10,

// for security this should always be set to 2.
'curl_ssl_verifyhost'     => 2,
// for security this should always be set to true.
'curl_ssl_verifypeer'     => true,
// for security this should always be set to true.
'use_ssl'                  => true,

// you can get the latest cacert.pem from here http://curl.haxx.se/ca/cacert.
↪pem
// if you're getting HTTP 0 responses, check cacert.pem exists and is readable
// without it curl won't be able to create an SSL connection
'curl_cainfo'              => __DIR__ . DIRECTORY_SEPARATOR . 'cacert.pem',
'curl_capath'              => __DIR__,

// in some cases (very very odd ones) the SSL version must be set manually.
// unless you know why your are changing this, you should leave it as false
// to allow PHP to determine the value for this setting itself.
'curl_sslversion'         => false,

'curl_followlocation'     => false, // whether to follow redirects or not

// support for proxy servers
'curl_proxy'              => false, // really you don't want to use this.
↪if you are using streaming
'curl_proxyuserpwd'       => false, // format username:password for proxy,
↪if required
'curl_encoding'           => '', // leave blank for all supported
↪formats, else use gzip, deflate, identity etc

// streaming API configuration
'is_streaming'            => false,
'streaming_eol'           => "\r\n",
'streaming_metrics_interval' => 10,

// header or querystring. You should always use header!
// this is just to help me debug other developers implementations
'as_header'               => true,
'force_nonce'             => false, // used for checking signatures. leave
↪as false for auto
'force_timestamp'         => false, // used for checking signatures. leave

```

(continues on next page)

(continued from previous page)

```

↪as false for auto
    ),
    $config
);
}

private function reset_request_settings($options=array()) {
    $this->request_settings = array(
        'params' => array(),
        'headers' => array(),
        'with_user' => true,
        'multipart' => false,
    );

    if (!empty($options))
        $this->request_settings = array_merge($this->request_settings, $options);
}

/**
 * Sets the useragent for PHP to use
 * If '$this->config['user_agent']' already has a value it is used instead of one
 * being generated.
 *
 * @return void value is stored to the config array class variable
 */
private function set_user_agent() {
    if (!empty($this->config['user_agent']))
        return;

    $ssl = ($this->config['curl_ssl_verifyhost'] && $this->config['curl_ssl_verifypeer']
↪) && $this->config['use_ssl']) ? '+' : '-';
    $ua = 'tmhOAuth ' . self::VERSION . $ssl . 'SSL - //github.com/thematttharris/
↪tmhOAuth';
    $this->config['user_agent'] = $ua;
}

/**
 * Generates a random OAuth nonce.
 * If 'force_nonce' is false a nonce will be generated, otherwise the value of '
↪$this->config['force_nonce']' will be used.
 *
 * @param string $length how many characters the nonce should be before MD5 hashing.
↪ default 12
 * @param string $include_time whether to include time at the beginning of the nonce.
↪ default true
 * @return $nonce as a string
 */
private function nonce($length=12, $include_time=true) {
    if ($this->config['force_nonce'] === false) {
        $prefix = $include_time ? microtime() : '';
        return md5(substr($prefix . uniqid(), 0, $length));
    } else {
        return $this->config['force_nonce'];
    }
}

```

(continues on next page)

(continued from previous page)

```

/**
 * Generates a timestamp.
 * If 'force_timestamp' is false a timestamp will be generated, otherwise the value
of '$this->config['force_timestamp']' will be used.
 *
 * @return $time as a string
 */
private function timestamp() {
    if ($this->config['force_timestamp'] === false) {
        $time = time();
    } else {
        $time = $this->config['force_timestamp'];
    }
    return (string) $time;
}

/**
 * Encodes the string or array passed in a way compatible with OAuth.
 * If an array is passed each array value will will be encoded.
 *
 * @param mixed $data the scalar or array to encode
 * @return $data encoded in a way compatible with OAuth
 */
private function safe_encode($data) {
    if (is_array($data)) {
        return array_map(array($this, 'safe_encode'), $data);
    } else if (is_scalar($data)) {
        return str_replace(
            array('+', '%7E'),
            array(' ', '~'),
            rawurlencode($data)
        );
    } else {
        return '';
    }
}

/**
 * Decodes the string or array from it's URL encoded form
 * If an array is passed each array value will will be decoded.
 *
 * @param mixed $data the scalar or array to decode
 * @return string $data decoded from the URL encoded form
 */
private function safe_decode($data) {
    if (is_array($data)) {
        return array_map(array($this, 'safe_decode'), $data);
    } else if (is_scalar($data)) {
        return rawurldecode($data);
    } else {
        return '';
    }
}

/**
 * Prepares OAuth1 signing parameters.

```

(continues on next page)

(continued from previous page)

```

*
* @return void all required OAuth parameters, safely encoded, are stored to the
class variable '$this->request_settings['oauth1_params']'
*/
private function prepare_oauth1_params() {
    $defaults = array(
        'oauth_nonce'           => $this->nonce(),
        'oauth_timestamp'       => $this->timestamp(),
        'oauth_version'         => $this->config['oauth_version'],
        'oauth_consumer_key'    => $this->config['consumer_key'],
        'oauth_signature_method' => $this->config['oauth_signature_method'],
    );

    // include the user token if it exists
    if ( $oauth_token = $this->token() )
        $defaults['oauth_token'] = $oauth_token;

    $this->request_settings['oauth1_params'] = array();

    // safely encode
    foreach ($defaults as $k => $v) {
        $this->request_settings['oauth1_params'][$this->safe_encode($k)] = $this->safe_
        encode($v);
    }
}

private function token() {
    if ( $this->request_settings['with_user'] ) {
        if (isset($this->config['token']) && !empty($this->config['token'])) return
        $this->config['token'];
        elseif (isset($this->config['user_token'])) return $this->config['user_token'];
    }
    return '';
}

private function secret() {
    if ( $this->request_settings['with_user'] ) {
        if (isset($this->config['secret']) && !empty($this->config['secret'])) return
        $this->config['secret'];
        elseif (isset($this->config['user_secret'])) return $this->config['user_secret
        '];
    }
    return '';
}

/**
 * Extracts and decodes OAuth parameters from the passed string
 *
 * @param string $body the response body from an OAuth flow method
 * @return array the response body safely decoded to an array of key => values
 */
public function extract_params($body) {
    $kvs = explode('&', $body);
    $decoded = array();
    foreach ($kvs as $kv) {
        $kv = explode('=', $kv, 2);

```

(continues on next page)

(continued from previous page)

```

        $kv[0] = $this->safe_decode($kv[0]);
        $kv[1] = $this->safe_decode($kv[1]);
        $decoded[$kv[0]] = $kv[1];
    }
    return $decoded;
}

/**
 * Prepares the HTTP method for use in the base string by converting it to
 * uppercase.
 *
 * @return void value is stored to the class variable '$this->request_settings[
↳ 'method']'
 */
private function prepare_method() {
    $this->request_settings['method'] = strtoupper($this->request_settings['method']);
}

/**
 * Prepares the URL for use in the base string by ripping it apart and
 * reconstructing it.
 *
 * Ref: 3.4.1.2
 *
 * @return void value is stored to the class array variable '$this->request_
↳ settings['url']'
 */
private function prepare_url() {
    $parts = parse_url($this->request_settings['url']);

    $port = isset($parts['port']) ? $parts['port'] : false;
    $scheme = $parts['scheme'];
    $host = $parts['host'];
    $path = isset($parts['path']) ? $parts['path'] : false;

    $port or $port = ($scheme == 'https') ? '443' : '80';

    if (($scheme == 'https' && $port != '443') || ($scheme == 'http' && $port != '80
↳ ')) {
        $host = "$host:$port";
    }

    // the scheme and host MUST be lowercase
    $this->request_settings['url'] = strtolower("$scheme://$host");
    // but not the path
    $this->request_settings['url'] .= $path;
}

/**
 * If the request uses multipart, and the parameter isn't a file path, prepend a
↳ space
 * otherwise return the original value. we chose a space here as twitter whitespace
↳ trims from
 * the beginning of the tweet. we don't use \0 here because it's the character for
↳ string
 * termination.

```

(continues on next page)

(continued from previous page)

```

*
* @param the parameter value
* @return string the original or modified string, depending on the request and the
↪input parameter
*/
private function multipart_escape($value) {
    if (!$this->request_settings['multipart'] || strpos($value, '@') !== 0)
        return $value;

    // see if the parameter is a file.
    // we split on the semi-colon as it's the delimiter used on media uploads
    // for fields with semi-colons this will return the original string
    list($file) = explode(';', substr($value, 1), 2);
    if (file_exists($file))
        return $value;

    return " $value";
}

/**
* Prepares all parameters for the base string and request.
* Multipart parameters are ignored as they are not defined in the specification,
* all other types of parameter are encoded for compatibility with OAuth.
*
* @param array $params the parameters for the request
* @return void prepared values are stored in the class array variable '$this->
↪request_settings'
*/
private function prepare_params() {
    $doing_oauth1 = false;
    $this->request_settings['prepared_params'] = array();
    $prepared = &$this->request_settings['prepared_params'];
    $prepared_pairs = array();
    $prepared_pairs_with_oauth = array();

    if (isset($this->request_settings['oauth1_params'])) {
        $oauth1 = &$this->request_settings['oauth1_params'];
        $doing_oauth1 = true;
        $params = array_merge($oauth1, $this->request_settings['params']);

        // Remove oauth_signature if present
        // Ref: Spec: 9.1.1 ("The oauth_signature parameter MUST be excluded.")
        unset($params['oauth_signature']);

        // empty the oauth1 array. we reset these values later in this method
        $oauth1 = array();
    } else {
        $params = $this->request_settings['params'];
    }

    // Parameters are sorted by name, using lexicographical byte value ordering.
    // Ref: Spec: 9.1.1 (1)
    uksort($params, 'strcmp');

    // set this now so we're not doing it on every parameter

```

(continues on next page)

(continued from previous page)

```

$supports_curl_file = class_exists('CurlFile', false);

// encode params unless we're doing multipart
foreach ($params as $k => $v) {
    $k = $this->request_settings['multipart'] ? $k : $this->safe_encode($k);

    if (is_array($v))
        $v = implode(',', $v);

    // we don't need to do the multipart escaping if we support curlfile
    if ($supports_curl_file && ($v instanceof CurlFile)) {
        // leave $v alone
    } elseif ($this->request_settings['multipart']) {
        $v = $this->multipart_escape($v);
    } else {
        $v = $this->safe_encode($v);
    }

    // split parameters for the basestring and authorization header, and recreate
    the oauth1 array
    if ($doing_oauth1) {
        // if we're doing multipart, only store the oauth_* params, ignore the users
        request params
        if ((strpos($k, 'oauth') === 0) || !$this->request_settings['multipart'])
            $prepared_pairs_with_oauth[] = "{$k}={$v}";

        if (strpos($k, 'oauth') === 0) {
            $oauth1[$k] = $v;
            continue;
        }
    }
    $prepared[$k] = $v;

    if (!$this->request_settings['multipart'])
        $prepared_pairs[] = "{$k}={$v}";
}

if ($doing_oauth1) {
    $this->request_settings['basestring_params'] = implode('&', $prepared_pairs_
    with_oauth);
}

// setup params for GET/POST/PUT method handling
if (!empty($prepared)) {
    $content = implode('&', $prepared_pairs);

    switch ($this->request_settings['method']) {
        case 'PUT':
            // fall through to POST as PUT should be treated the same
        case 'POST':
            $this->request_settings['postfields'] = $this->request_settings['multipart
            ''] ? $prepared : $content;
            break;
        default:
            $this->request_settings['querystring'] = $content;
            break;
    }
}

```

(continues on next page)

(continued from previous page)

```

    }
  }
}

/**
 * Prepares the OAuth signing key
 *
 * @return void prepared signing key is stored in the class variable 'signing_key'
 */
private function prepare_signing_key() {
    $left = $this->safe_encode($this->config['consumer_secret']);
    $right = $this->safe_encode($this->secret());
    $this->request_settings['signing_key'] = $left . '&' . $right;
}

/**
 * Prepare the base string.
 * Ref: Spec: 9.1.3 ("Concatenate Request Elements")
 *
 * @return void prepared base string is stored in the class variable 'base_string'
 */
private function prepare_base_string() {
    $url = $this->request_settings['url'];

    # if the host header is set we need to rewrite the basestring to use
    # that, instead of the request host. otherwise the signature won't match
    # on the server side
    if (!empty($this->request_settings['headers']['Host'])) {
        $url = str_ireplace(
            $this->config['host'],
            $this->request_settings['headers']['Host'],
            $url
        );
    }

    $base = array(
        $this->request_settings['method'],
        $url,
        $this->request_settings['basestring_params']
    );
    $this->request_settings['basestring'] = implode('&', $this->safe_encode($base));
}

/**
 * Signs the OAuth 1 request
 *
 * @return void oauth_signature is added to the parameters in the class array
 * variable '$this->request_settings'
 */
private function prepare_oauth_signature() {
    switch ($this->config['oauth_signature_method']) {
        case 'HMAC-SHA1':
            $signature = $this->sign_with_hmac('sha1');
            break;
        case 'HMAC-SHA256':
            $signature = $this->sign_with_hmac('sha256');

```

(continues on next page)

(continued from previous page)

```

        break;
    case 'RSA-SHA1':
        $signature = $this->sign_with_rsa(OPENSSL_ALGO_SHA1);
        break;
    case 'RSA-SHA256':
        $signature = $this->sign_with_rsa(OPENSSL_ALGO_SHA256);
        break;
    default:
        throw new Exception("Unsupported oauth_signature_method: '" . $this->config[
↪ 'oauth_signature_method'] . "'");
    }
    $this->request_settings['oauth1_params']['oauth_signature'] = $this->safe_
↪ encode(base64_encode($signature));
}

/**
 * Signs the OAuth 1 request using HMAC-based signature algorithm
 *
 * @param string $algorithm algorithm name (like sha1 or sha256)
 * @return binary signature
 */
private function sign_with_hmac($algorithm) {
    return hash_hmac(
        $algorithm, $this->request_settings['basestring'], $this->request_settings[
↪ 'signing_key'], true
    );
}

/**
 * Signs the OAuth 1 request using RSA-based signature algorithm
 *
 * @param mixed $algorithm ID or name of hash algorithm that will be
 * used to compute base string hash before encrypting it with RSA;
 * values understood by openssl_sign()'s $signature_alg parameter
 * are accepted here (like 'sha1' or OPENSSL_ALGO_SHA256)
 * @return binary signature
 */
private function sign_with_rsa($algorithm) {
    if (!function_exists('openssl_sign')) {
        throw new Exception("openssl_sign function does not exist. Please make sure
↪ OpenSSL extension is installed");
    }
    if ($this->config['private_key_pem'] == '') {
        throw new Exception("No private key PEM is configured, cannot sign");
    }
    $ok = openssl_sign($this->request_settings['basestring'], $signature, $this->
↪ config['private_key_pem'], $algorithm);
    if (!$ok) {
        throw new Exception("Cannot sign: " . openssl_error_string());
    }
    return $signature;
}

/**
 * Prepares the Authorization header
 *

```

(continues on next page)

(continued from previous page)

```

    * @return void prepared authorization header is stored in the class variable.
    headers['Authorization']
    */
    private function prepare_auth_header() {
        if (!$this->config['as_header'])
            return;

        // oauth1
        if (isset($this->request_settings['oauth1_params'])) {
            // sort again as oauth_signature was added post param preparation
            usort($this->request_settings['oauth1_params'], 'strcmp');
            $encoded_quoted_pairs = array();
            foreach ($this->request_settings['oauth1_params'] as $k => $v) {
                $encoded_quoted_pairs[] = "{$k}=\"{$v}\"";
            }
            $header = 'OAuth ' . implode(' ', $encoded_quoted_pairs);
        } elseif (!empty($this->config['bearer'])) {
            $header = 'Bearer ' . $this->config['bearer'];
        }

        if (isset($header))
            $this->request_settings['headers']['Authorization'] = $header;
    }

    /**
     * Create the bearer token for OAuth2 requests from the consumer_key and consumer_
     * secret.
     */
    * @return string the bearer token
    */
    public function bearer_token_credentials() {
        $credentials = implode(':', array(
            $this->safe_encode($this->config['consumer_key']),
            $this->safe_encode($this->config['consumer_secret'])
        ));
        return base64_encode($credentials);
    }

    /**
     * Make an HTTP request using this library. This method doesn't return anything.
     * Instead the response should be inspected directly.
     *
     * @param string $method the HTTP method being used. e.g. POST, GET, HEAD etc
     * @param string $url the request URL without query string parameters
     * @param array $params the request parameters as an array of key=value pairs.
     * @param array $useauth whether to use authentication when making the request.
     * @param string $multipart whether this request contains multipart data. Default
     * @param array $headers any custom headers to send with the request. Default empty
     * @return int the http response code for the request. 0 is returned if a
     * connection could not be made
     */
    public function request($method, $url, $params=array(), $useauth=true,

```

(continues on next page)

(continued from previous page)

```

↪$multipart=false, $headers=array()) {
    $options = array(
        'method'    => $method,
        'url'       => $url,
        'params'    => $params,
        'with_user' => true,
        'multipart' => $multipart,
        'headers'   => $headers
    );
    $options = array_merge($this->default_options(), $options);

    if ($useauth) {
        return $this->user_request($options);
    } else {
        return $this->unauthenticated_request($options);
    }
}

public function aponly_request($options=array()) {
    $options = array_merge($this->default_options(), $options, array(
        'with_user' => false,
    ));
    $this->reset_request_settings($options);
    if ($options['without_bearer']) {
        return $this->oauth1_request();
    } else {
        $this->prepare_method();
        $this->prepare_url();
        $this->prepare_params();
        $this->prepare_auth_header();
        return $this->curlit();
    }
}

public function user_request($options=array()) {
    $options = array_merge($this->default_options(), $options, array(
        'with_user' => true,
    ));
    $this->reset_request_settings($options);
    return $this->oauth1_request();
}

public function unauthenticated_request($options=array()) {
    $options = array_merge($this->default_options(), $options, array(
        'with_user' => false,
    ));
    $this->reset_request_settings($options);
    $this->prepare_method();
    $this->prepare_url();
    $this->prepare_params();
    return $this->curlit();
}

/**
 * Signs the request and adds the OAuth signature. This runs all the request
 * parameter preparation methods.

```

(continues on next page)

(continued from previous page)

```

*
* @param string $method the HTTP method being used. e.g. POST, GET, HEAD etc
* @param string $url the request URL without query string parameters
* @param array $params the request parameters as an array of key=value pairs
* @param boolean $with_user whether to include the user credentials when making
↳ the request.
* @return void
*/
private function oauth1_request() {
    $this->prepare_oauth1_params();
    $this->prepare_method();
    $this->prepare_url();
    $this->prepare_params();
    $this->prepare_base_string();
    $this->prepare_signing_key();
    $this->prepare_oauth_signature();
    $this->prepare_auth_header();
    return $this->curlit();
}

private function default_options() {
    return array(
        'method'          => 'GET',
        'params'           => array(),
        'with_user'        => true,
        'multipart'        => false,
        'headers'          => array(),
        'without_bearer'   => false,
    );
}

/**
* Make a long poll HTTP request using this library. This method is
* different to the other request methods as it isn't supposed to disconnect
*
* Using this method expects a callback which will receive the streaming
* responses.
*
* @param string $method the HTTP method being used. e.g. POST, GET, HEAD etc
* @param string $url the request URL without query string parameters
* @param array $params the request parameters as an array of key=value pairs
* @param string $callback the callback function to stream the buffer to.
* @return void
*/
public function streaming_request($method, $url, $params=array(), $callback='') {
    if ( ! empty($callback) ) {
        if ( ! is_callable($callback) ) {
            return false;
        }
        $this->config['streaming_callback'] = $callback;
    }
    $this->metrics['start']          = time();
    $this->metrics['interval_start'] = $this->metrics['start'];
    $this->metrics['messages']       = 0;
    $this->metrics['last_messages']  = 0;
    $this->metrics['bytes']          = 0;

```

(continues on next page)

(continued from previous page)

```

$this->metrics['last_bytes']    = 0;
$this->config['is_streaming']    = true;
$this->request($method, $url, $params);
}

/**
 * Handles the updating of the current Streaming API metrics.
 *
 * @return array the metrics for the streaming api connection
 */
private function update_metrics() {
    $now = time();
    if (($this->metrics['interval_start'] + $this->config['streaming_metrics_interval']
↪ ']) > $now)
        return null;

    $this->metrics['mps'] = round( ($this->metrics['messages'] - $this->metrics['last_
↪ messages']) / $this->config['streaming_metrics_interval'], 2);
    $this->metrics['bps'] = round( ($this->metrics['bytes'] - $this->metrics['last_
↪ bytes']) / $this->config['streaming_metrics_interval'], 2);

    $this->metrics['last_bytes'] = $this->metrics['bytes'];
    $this->metrics['last_messages'] = $this->metrics['messages'];
    $this->metrics['interval_start'] = $now;
    return $this->metrics;
}

/**
 * Utility function to create the request URL in the requested format.
 * If a fully-qualified URI is provided, it will be returned.
 * Any multi-slashes (except for the protocol) will be replaced with a single slash.
 *
 * @param string $request the API method without extension
 * @param string $extension the format of the response. Default json. Set to an
↪ empty string to exclude the format
 * @return string the concatenation of the host, API version, API method and format,
↪ or $request if it begins with http
 */
public function url($request, $extension='json') {
    // remove multi-slashes
    $request = preg_replace('$(^:)]//+$', '$1/', $request);

    if (stripos($request, 'http') === 0 || stripos($request, '//') === 0) {
        return $request;
    }

    $extension = strlen($extension) > 0 ? ".$extension" : '';
    $proto = $this->config['use_ssl'] ? 'https://' : 'http://';

    // trim trailing slash
    $request = ltrim($request, '/');

    $pos = strlen($request) - strlen($extension);
    if (substr($request, $pos) === $extension)
        $request = substr_replace($request, '', $pos);

```

(continues on next page)

(continued from previous page)

```

    return implode('/', array(
        $proto,
        $this->config['host'],
        $request . $extension
    ));
}

/**
 * Public access to the private safe decode/encode methods
 *
 * @param string $text the text to transform
 * @param string $mode the transformation mode. either encode or decode
 * @return string $text transformed by the given $mode
 */
public function transformText($text, $mode='encode') {
    return $this->{"safe_$mode"}($text);
}

/**
 * Utility function to parse the returned curl headers and store them in the
 * class array variable.
 *
 * @param object $ch curl handle
 * @param string $header the response headers
 * @return string the length of the header
 */
private function curlHeader($ch, $header) {
    $this->response['raw'] .= $header;

    list($key, $value) = array_pad(explode(':', $header, 2), 2, null);

    $key = trim($key);
    $value = trim($value);

    if ( ! isset($this->response['headers'][$key])) {
        $this->response['headers'][$key] = $value;
    } else {
        if (!is_array($this->response['headers'][$key])) {
            $this->response['headers'][$key] = array($this->response['headers'][$key]);
        }
        $this->response['headers'][$key][] = $value;
    }

    return strlen($header);
}

/**
 * Utility function to parse the returned curl buffer and store them until
 * an EOL is found. The buffer for curl is an undefined size so we need
 * to collect the content until an EOL is found.
 *
 * This function calls the previously defined streaming callback method.
 *
 * @param object $ch curl handle
 * @param string $data the current curl buffer

```

(continues on next page)

(continued from previous page)

```

    * @return int the length of the data string processed in this function
    */
private function curlWrite($ch, $data) {
    $l = strlen($data);
    if (strpos($data, $this->config['streaming_eol']) === false) {
        $this->buffer .= $data;
        return $l;
    }

    $buffered = explode($this->config['streaming_eol'], $data);
    $content = $this->buffer . $buffered[0];

    $this->metrics['messages']++;
    $this->metrics['bytes'] += strlen($content);

    if ( ! is_callable($this->config['streaming_callback']))
        return 0;

    $metrics = $this->update_metrics();
    $stop = call_user_func(
        $this->config['streaming_callback'],
        $content,
        strlen($content),
        $metrics
    );
    $this->buffer = $buffered[1];
    if ($stop)
        return 0;

    return $l;
}

/**
 * Makes a curl request. Takes no parameters as all should have been prepared
 * by the request method
 *
 * the response data is stored in the class variable 'response'
 *
 * @return int the http response code for the request. 0 is returned if a
↳ connection could not be made
 */
private function curlit() {
    $this->response = array(
        'raw' => ''
    );

    // configure curl
    $c = curl_init();

    if ($this->request_settings['method'] == 'GET' && isset($this->request_settings[
↳ 'querystring'])) {
        $this->request_settings['url'] = $this->request_settings['url'] . '?' . $this->
↳ request_settings['querystring'];
    } elseif ($this->request_settings['method'] == 'POST' || $this->request_settings[
↳ 'method'] == 'PUT') {
        $postfields = array();

```

(continues on next page)

(continued from previous page)

```

    if (isset($this->request_settings['postfields']))
        $postfields = $this->request_settings['postfields'];

    curl_setopt($c, CURLOPT_POSTFIELDS, $postfields);
}

curl_setopt($c, CURLOPT_CUSTOMREQUEST, $this->request_settings['method']);

curl_setopt_array($c, array(
    CURLOPT_HTTP_VERSION => $this->config['curl_http_version'],
    CURLOPT_USERAGENT => $this->config['user_agent'],
    CURLOPT_CONNECTTIMEOUT => $this->config['curl_connecttimeout'],
    CURLOPT_TIMEOUT => $this->config['curl_timeout'],
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_SSL_VERIFYPEER => $this->config['curl_ssl_verifypeer'],
    CURLOPT_SSL_VERIFYHOST => $this->config['curl_ssl_verifyhost'],

    CURLOPT_FOLLOWLOCATION => $this->config['curl_followlocation'],
    CURLOPT_PROXY => $this->config['curl_proxy'],
    CURLOPT_ENCODING => $this->config['curl_encoding'],
    CURLOPT_URL => $this->request_settings['url'],
    // process the headers
    CURLOPT_HEADERFUNCTION => array($this, 'curlHeader'),
    CURLOPT_HEADER => false,
    CURLINFO_HEADER_OUT => true,
));

if ($this->config['curl_cainfo'] !== false)
    curl_setopt($c, CURLOPT_CAINFO, $this->config['curl_cainfo']);

if ($this->config['curl_capath'] !== false)
    curl_setopt($c, CURLOPT_CAPATH, $this->config['curl_capath']);

if ($this->config['curl_proxyuserpwd'] !== false)
    curl_setopt($c, CURLOPT_PROXYUSERPWD, $this->config['curl_proxyuserpwd']);

if ($this->config['curl_sslversion'] !== false)
    curl_setopt($c, CURLOPT_SSLVERSION, $this->config['curl_sslversion']);

if ($this->config['is_streaming']) {
    // process the body
    $this->response['content-length'] = 0;
    curl_setopt($c, CURLOPT_TIMEOUT, 0);
    curl_setopt($c, CURLOPT_WRITEFUNCTION, array($this, 'curlWrite'));
}

if ( ! empty($this->request_settings['headers'])) {
    foreach ($this->request_settings['headers'] as $k => $v) {
        $headers[] = trim($k . ': ' . $v);
    }
    curl_setopt($c, CURLOPT_HTTPHEADER, $headers);
}

if (isset($this->config['block']) && (true === $this->config['block']))
    return 0;

```

(continues on next page)

(continued from previous page)

```

// do it!
$response = curl_exec($c);
$code = curl_getinfo($c, CURLINFO_HTTP_CODE);
$info = curl_getinfo($c);
$error = curl_error($c);
$errno = curl_errno($c);
curl_close($c);

// store the response
$this->response['code'] = $code;
$this->response['response'] = $response;
$this->response['info'] = $info;
$this->response['error'] = $error;
$this->response['errno'] = $errno;

if (!isset($this->response['raw'])) {
    $this->response['raw'] = '';
}
$this->response['raw'] .= $response;

return $code;
}
}

/*
 * Configuration - Replace these values with your actual credentials
 */
$config = [
    'server_host' => 'sandbox.doc2.com',
    'endpoint_id' => 1, // Your endpoint ID
    'merchant_login' => 'your-merchant-login',
    'private_key_file' => 'test-private-key.pem', // Your private key file
    'test_mode' => true // Set to false for production
];

/*
 * API Request Function
 */
function makeApiRequest($endpoint, $params, $config) {
    // Load private key
    $private_key_pem = file_get_contents($config['private_key_file']);
    if (!$private_key_pem) {
        throw new Exception("Could not load private key file");
    }

    // Initialize OAuth
    $oauth = new tmhOAuth();
    $oauth->reconfigure([
        'consumer_key' => $config['merchant_login'],
        'oauth_signature_method' => 'RSA-SHA256',
        'private_key_pem' => $private_key_pem,
        'curl_ssl_verifypeer' => $config['test_mode'] ? false : true
    ]);

    // Make request
    $url = "https://{ $config['server_host'] }/paynet/api/v4/{ $endpoint }/{ $config["

```

(continues on next page)

(continued from previous page)

```

    'endpoint_id']}";
    $status = $oauth->request('POST', $url, $params);

    // Handle response
    if ($status == 0) {
        throw new Exception('HTTP request failed');
    } elseif ($status != 200) {
        throw new Exception("HTTP error: {$status}");
    }

    return [
        'status' => $status,
        'response' => $oauth->response['response'],
        'headers' => $oauth->response['headers']
    ];
}

/*
 * Example Usage - Transfer by Reference
 */
if (isset($_POST['action']) && $_POST['action'] == 'transfer') {
    try {
        $params = [
            'client-order-id' => 'transfer-' . uniqid(),
            'amount' => $_POST['amount'] ?? '100',
            'currency' => $_POST['currency'] ?? 'RUB',
            'order_desc' => $_POST['description'] ?? 'Consumer credit',
            'destination-card-no' => $_POST['card_number'] ?? '4444333322221111'
        ];

        $result = makeApiRequest('transfer-by-ref', $params, $config);
        $response = json_decode($result['response'], true);

        echo "<div class='alert alert-success'>Success! Transaction ID: " .
            ($response['transaction-id'] ?? 'N/A') . "</div>";

    } catch (Exception $e) {
        echo "<div class='alert alert-danger'>Error: " . $e->getMessage() . "</div>";
    }
}

/*
 * Example Usage - Bank Payout
 */
if (isset($_POST['action']) && $_POST['action'] == 'payout') {
    try {
        $params = [
            'client-order-id' => 'payout-' . uniqid(),
            'amount' => $_POST['amount'] ?? '100',
            'currency' => $_POST['currency'] ?? 'USD',
            'order_desc' => $_POST['description'] ?? 'Bank payout',
            'account_number' => $_POST['account_number'] ?? '123412341234',
            'bank_name' => $_POST['bank_name'] ?? 'Test Bank'
        ];

        $result = makeApiRequest('payout', $params, $config);
    }
}

```

(continues on next page)

(continued from previous page)

```
$response = json_decode($result['response'], true);

echo "<div class='alert alert-success'>Success! Payout ID: " .
    ($response['payout-id'] ?? 'N/A') . "</div>";

} catch (Exception $e) {
    echo "<div class='alert alert-danger'>Error: " . $e->getMessage() . "</div>";
}
?>
```

API COMMANDS, CALLBACKS & REDIRECTS

7.1 /api/v2/account-verification

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
 - [SBP Sale Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.1.1 Introduction

Account Verification is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.1.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/account-verification/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/account-verification/ENDPOINTID

continues on next page

Table 1 – continued from previous page

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/account-verification/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v2/account-verification/group/ENDPOINTGROUPID

7.1.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.
 Acquirer can redefine the necessity of some fields so they become required instead of optional.
 Leading and trailing whitespace in input parameters will be omitted.

Warning: The following characters must be escaped in the parameter values: & + “.

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10

continues on next page

Table 2 – continued from previous page

Parameter Name	Description	Value
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50
cvv2	Payer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card.	Necessity: Required Type: Numeric Length: 3-4
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL) 2. Request parameter: client_orderid 3. Request parameter: email 4. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 128
first_name	Payer's first name.	Necessity: Optional Type: String Length: 50
last_name	Payer's last name.	Necessity: Optional Type: String Length: 50
cardrefid	Card reference ID obtained at Card Registration Stage .	Necessity: Optional Type: String Length: 50

continues on next page

Table 2 – continued from previous page

Parameter Name	Description	Value
ssn	Last four digits of the Payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYYMMDD.	Necessity: Optional Type: Numeric Length: 8
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Conditional Type: String Length: 2-3
phone	Payer's full international phone number, including country code.	Necessity: Optional Type: String Length: 15
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
credit_card_number	Payer's credit card number. Send either combination of credit_card_number, card_printed_name, expire_month and expire_year or cardrefid, not all.	Necessity: Conditional Type: Numeric Length: 20
card_printed_name	Cardholder name, printed on the bank card.	Necessity: Conditional Type: String Length: 128
expire_month	Bank card expiration month.	Necessity: Conditional Type: Numeric Length: 2

continues on next page

Table 2 – continued from previous page

Parameter Name	Description	Value
expire_year	Bank card expiration year.	Necessity: Conditional Type: Numeric Length: 4
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128
purpose	Destination to where the payment goes. It is useful for the Connecting Partys who let their Payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 128

continues on next page

Table 2 – continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 128

SBP Sale Request Parameters

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10

continues on next page

Table 3 – continued from previous page

Parameter Name	Description	Value
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45

continues on next page

Table 3 – continued from previous page

Parameter Name	Description	Value
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL) 2. Request parameter: client_orderid 3. Request parameter: amount (in minor units) 4. Request parameter: email 5. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status . Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command . Pass http://https://doc2.codetime.net if you have no need to return payer anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128

continues on next page

Table 3 – continued from previous page

Parameter Name	Description	Value
purpose	Destination to where the payment goes. It is useful for the Connecting Party's who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 3 – continued from previous page

Parameter Name	Description	Value
merchant_data	Custom Connecting Party details, which can be attached to the transaction and passed back in the status response , Connecting Party callback parameters or server_callback_url. Additional information of transaction which may be useful in Connecting Party's external system, e.g. VIP customer, TV promo campaign lead. Information returns in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
dapi_imei	Unique device identifier.	Necessity: Optional Type: String Length: 32

7.1.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. May be async-response, validation-error, error etc. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.1.5 Request Example

```
POST /paynet/api/v2/account-verification/39915 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 527
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=34T43R77N
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&birthday=19820115
&address1=100 Main st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&email=john.smith@gmail.com
&ipaddress=65.153.12.232
&site_url=www.google.com
&credit_card_number=4666034752545240
&card_printed_name=CARD HOLDER
&expire_month=12
&expire_year=2099
&cvv2=123
&purpose=user_account1
&server_callback_url=https://httpstat.us/200
&merchant_data=VIP customer
&control=5112b51084be33fa331d995d17b9d82af3bfbbc9
```

7.1.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 27 Apr 2023 14:03:58 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 145

type=async-response
&serial-number=000000000-0000-0000-0000-000002e34518
&merchant-order-id=34T43R77N
&paynet-order-id=6994016
&end-point-id=39915
```

7.1.7 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Mon, 22 May 2023 11:14:25 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 152

type=validation-error
&serial-number=00000000-0000-0000-0000-000002e5c583
&merchant-order-id=34T43R77N
&error-message=INVALID_CONTROL_CODE
&error-code=2
```

7.1.8 Postman Collection

7.1.9 Request Builder

7.2 /api/v2/account-verification-form

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.2.1 Introduction

Account Verification form is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.2.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/account-verification-form/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/account-verification-form/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/account-verification-form/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v2/account-verification-form/group/ENDPOINTGROUPID

7.2.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Leading and trailing whitespace in input parameters will be omitted.

Acquirer can redefine the necessity of some fields so they become required instead of optional.

Request parameter name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256

continues on next page

Table 6 – continued from previous page

Request name	parameter	Description	Value
city		Payer's city.	Necessity: Required Type: String Length: 50
zip_code		Payer's ZIP code.	Necessity: Required Type: String Length: 10
country		Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
email		Payer's e-mail address.	Necessity: Required Type: String Length: 50
ipaddress		Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45
control		Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL) 2. Request parameter: client_orderid 3. Request parameter: email 4. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 128
first_name		Payer's first name.	Necessity: Optional Type: String Length: 50

continues on next page

Table 6 – continued from previous page

Request name	parameter	Description	Value
last_name		Payer's last name.	Necessity: Optional Type: String Length: 50
ssn		Last four digits of the Payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday		Payer's date of birth, in the format YYYYMMDD.	Necessity: Optional Type: Numeric Length: 8
state		Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Conditional Type: String Length: 2-3
phone		Payer's full international phone number, including country code.	Necessity: Optional Type: String Length: 15
cell_phone		Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
currency		Currency the transaction is charged in (three-letter currency code). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Optional Type: String Length: 3
site_url		The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128

continues on next page

Table 6 – continued from previous page

Request name	parameter	Description	Value
redirect_url		URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined or in any other final status. Connecting Party must not use the parameters which come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either redirect_url or a combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024
redirect_success_url		URL, where the Payer is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters which come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either redirect_url or a combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 6 – continued from previous page

Request parameter name	Description	Value
redirect_fail_url	URL, where the Payer is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters which come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either redirect_url or a combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters. This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 128

continues on next page

Table 6 – continued from previous page

Request name	parameter	Description	Value
notify_url		URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 128
preferred_language		Payer's two-letter language code for multi-language payment-forms.	Necessity: Optional Type: String Length: 2
merchant_form_data		Parameters sent as this parameter value are parsed into macros with the same name, the parameter is url-encoded, example: test-param%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters[a-zA-Z0-9], parameter value characters[a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128

7.2.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. May be async-form-response, validation-error, error etc. If type equals validation-error or error, error-message and error-code parameters contain error details.
payet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.
redirect-url	The URL to the page where the Connecting Party should redirect the Payer's browser. Connecting Party should send HTTP 302 redirect, see Account Verification Form Flow .

7.2.5 Request Example

```
POST /paynet/api/v2/account-verification-form/39915 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 565
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=34T43R77N
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100 Main st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&email=john.smith@gmail.com
&ipaddress=65.153.12.232
&site_url=www.google.com
```

(continues on next page)

(continued from previous page)

```
&currency=USD
&purpose=user_account1
&redirect_url=http://connectingparty.com/result
&server_callback_url=https://httpstat.us/200
&merchant_data=VIP customer
&merchant_form_data=testparam%3Dtest1%26mynewparam%3Dtest2
&control=5112b51084be33fa331d995d17b9d82af3bfbbc9
```

7.2.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 10 May 2023 09:07:27 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 285

type=async-form-response
&serial-number=000000000-0000-0000-0000-000002e378e8
&merchant-order-id=34T43R77N
&paynet-order-id=6998044
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
%2FBB587546567A31587163597A684535634A775969614A5538374E4535734B44743364784335716A7979316B673D
```

7.2.7 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Mon, 22 May 2023 11:36:59 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 152

type=validation-error
&serial-number=000000000-0000-0000-0000-000002e5c587
&merchant-order-id=34T43R77N
&error-message=INVALID_CONTROL_CODE
&error-code=2
```

7.2.8 Postman Collection

7.2.9 Request Builder

7.3 /api/v2/auth-cash

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.3.1 Introduction

Auth-cash is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.3.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/auth-cash/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/auth-cash/ENDPOINTID

7.3.3 Request Parameters

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15

continues on next page

Table 9 – continued from previous page

Parameter Name	Description	Value
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL). 2. Request parameter: client_orderid. 3. Request parameter: amount (in minor units). 4. Request parameter: email. 5. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
first_name	Payer's first name.	Necessity: Required Type: String Length: 50
last_name	Payer's last name.	Necessity: Required Type: String Length: 50
state	Payer's state. (two-letter state code). Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Conditional Type: String Length: 2

continues on next page

Table 9 – continued from previous page

Parameter Name	Description	Value
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined or in any other final status. Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if you have no need to return payer anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024
redirect_success_url	URL, where the Payer is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 9 – continued from previous page

Parameter Name	Description	Value
redirect_fail_url	URL, where the Payer is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
ssn	Last four digits of the Payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYYMMDD.	Necessity: Optional Type: Numeric Length: 8
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128

continues on next page

Table 9 – continued from previous page

Parameter Name	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
preferred_language	Payer's two-letter language code for multi-language invoices.	Necessity: Optional Type: String Length: 2

7.3.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
end-point-id	Endpoint id used for the transaction.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.3.5 Request Example

```
POST /paynet/api/v2/auth-cash/46760 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 592
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=inv3982750
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100 Main st
&city=Seattle&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&amount=55
&email=john.smith@gmail.com
&currency=USD
&ipaddress=65.153.12.232
&site_url=https://doc2.codetime.net
&purpose=user_account1
&redirect_url=http://sandbox.doc2.com/doc/dummy.htm
&server_callback_url=https://httpstat.us/200
```

(continues on next page)

(continued from previous page)

```
&merchant_data=VIP customer
&control=ce2dcc2b307c123242b9867cb0ce15617b8ee7f9
&preferred_language=en
```

7.3.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 14 Jun 2023 08:43:52 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 146

type=async-response
&serial-number=00000000-0000-0000-0000-000002eb1adf
&merchant-order-id=inv3982750
&paynet-order-id=7174243
&end-point-id=46760
```

7.3.7 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 14 Jun 2023 08:43:52 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 146

type=validation-error
&serial-number=00000000-0000-0000-0000-000002ddb0d4
&error-message=Project+with+currency+RUB+does+not+apply+request+with+currency+USD
&error-code=16
```

7.3.8 Postman Collection

7.3.9 Request Builder

7.4 /api/v2/documents/add-document

- [Introduction](#)
- [API URLs](#)
- [Request Headers](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.4.1 Introduction

Add document is initiated through HTTPS POST request by using [URL](#) and the [parameters](#) specified below. [OAuth HMAC-SHA1](#) is used for authentication (two-legged mode).

7.4.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration Base URL	Production Base URL
https://sandbox.doc2.com/paynet/api/v2/documents/add-document/ENDPOINTID/ORDERID	https://gate.doc2.com/paynet/api/v2/documents/add-document/ENDPOINTID/ORDERID

7.4.3 Request Headers

Content-Type	The request must contain this header with the following value: binary/octet-stream.
Content-Length	This header specifies correct file size in bytes.
Authorization	OAuth HMAC-SHA1 is used for authentication (two-legged mode).

7.4.4 Request Parameters

Request parameters	Description	Value
endpoint id	Endpoint of the order. This parameter is passed in request URL.	Necessity: Required Type: Numeric Length: 10
order id	Transaction order identifier assigned by Payment Gateway. This parameter is passed in request URL.	Necessity: Required Type: String Length: 128
consumer key	Connecting Party's login name. This parameter is used to calculate signature and is passed in Authorization request header as oauth_consumer_key.	Necessity: Required Type: String Length: 20
consumer secret	Connecting Party's control key. This parameter is used to calculate signature as a part of Encryption key for OAuth HMAC-SHA1.	Necessity: Required Type: String Length: 40
file name	Name of the file to upload, e.g. filename.txt. This parameter is passed in the request body as file and in a query part of request URL as fileName.	Necessity: Required Type: String Length: 255
comment.	Adds a comment to attached file, accessible via pointing cursor on file in order details page. This parameter is passed in a query part of request URL.	Necessity: Optional Type: String Length: 255
file content	Field for the contents of the file. Used to calculate only txt files via bytes per symbol. Use either file content or a combination of file sha1 and file size, not both. Mandatory, if file sha1 and file size fields are empty. This parameter is passed in the request body as a string.	Necessity: Conditional Type: String Length: 255

continues on next page

Table 13 - continued from previous page

Request parameters	Description	Value
file sha1	Calculate sha1 of the file and insert here. Use either file content or a combination of file sha1 and file size, not both. Mandatory, if file content field is empty. This parameter is passed in base part of request URL, after order id parameter.	Necessity: Conditional Type: String Length: 255
file size	Field for file size in bytes. Use either file content or a combination of file sha1 and file size, not both. Mandatory, if file content field is empty. This parameter is passed in Content-Length request header.	Necessity: Conditional Type: String Length: 255

7.4.5 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
status	May have values such as: success or error. Success means that document attached to order successfully. Error provides error-message.
error-message	If status is error this parameter contains the reason for decline or error details.

7.4.6 Request Example

```
POST /paynet/api/v2/documents/add-document/6655/1852755/
↪561d839dde5ae8deeced73c14902e04dc746bfc2?fileName=test5.txt&comment=no%20comments_
↪HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Content-Type: binary/octet-stream
Content-Length: 7
Authorization: OAuth realm="",oauth_signature_method="HMAC-SHA1",oauth_signature=
↪"ItDxdBQtPRqLiKTA0dy%2BAtdB5cI%3D"
Connection: close
```

7.4.7 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Tue, 18 Jul 2023 06:10:27 GMT
Content-Type: text/plain;charset=utf-8
Content-Length: 14
Connection: close
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000

status=success
```

7.4.8 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Tue, 18 Jul 2023 07:37:34 GMT
Content-Type: text/plain;charset=utf-8
Content-Length: 46
Connection: close
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000

status=error
&error-message=File+hash+mismatch
```

7.4.9 Request Builder

The file can be uploaded in 2 different ways:

1. Upload file attached to request - to perform this operation the curl request should be executed in the folder with the target file that must be attached. file sha1 and file size parameters must be filled.
2. Upload file with one-line content - to perform this operation file content parameter must be filled.

7.5 /api/v2/bank-wire-transfer

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.5.1 Introduction

Bank wire transfer is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.5.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/bank-wire-transfer/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/bank-wire-transfer/ENDPOINTID

7.5.3 Request Parameters

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128

continues on next page

Table 16 – continued from previous page

Parameter Name	Description	Value
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
payer-fullname	Payer's full name.	Necessity: Required Type: String Length: 128
recipient-account-number	Receiver's account number.	Necessity: Required Type: Numeric Length: 20
recipient-bank-bic	Receiver's Bank Identifier Code.	Necessity: Required Type: Numeric Length: 9
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
payer-email	Payer's e-mail address.	Necessity: Optional Type: String Length: 128
payer-phone	Payer's full international phone number, including country code.	Necessity: Optional Type: String Length: 128
recipient-name	Receiver's name.	Necessity: Required Type: String Length: 255
recipient-inn	Receiver's Taxpayer Identification Number.	Necessity: Required Type: String Length: 255

continues on next page

Table 16 – continued from previous page

Parameter Name	Description	Value
payment-details	Payment purpose.	Necessity: Required Type: String Length: 120
vat-amount	Value-added tax amount.	Necessity: Required Type: Numeric Length: 10
payer-identity-document	Type, series and number of the identification card.	Necessity: Required Type: String Length: 64
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID> (See: Request URL) 2. Request parameter: client_orderid 4. Request parameter: payer-fullname 8. Request parameter: recipient-name 5. Request parameter: recipient-account-number 6. Request parameter: recipient-bank-bic 7. Request parameter: amount (in minor units) 9. Request parameter: vat-amount (in minor units) 3. Request parameter: currency 10. merchant_control (Control key assigned to Connecting Party in Payment Gateway.)	Necessity: Required Type: String Length: 40

continues on next page

Table 16 - continued from previous page

Parameter Name	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 128
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 128

7.5.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter name	Description
type	The type of response. May be async-form-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is declined or error this parameter contains the reason for decline or error details.
error-code	The error code in case of declined or error status.

7.5.5 Request Example

```
POST /paynet/api/v2/bank-wire-transfer/40265 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 436
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=902B4FF5
&payer-fullname=Test full name
&recipient-name=Test name
&recipient-account-number=40963810202157880527
&recipient-bank-bic=44030861
&amount=55.55
&vat-amount=4.54
&currency=EUR
&payer-email=john.smith@gmail.com
&payer-phone=+74951129876
&recipient-inn=7812013775
&payment-details=Test payment
&payer-identity-document=01,9207123123
&server_callback_url=https://httpstat.us/200
&control=bc7dae1fe9389aa02471c7c93d08d6b1ba0f2ea8
```


7.5.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Fri, 26 May 2023 11:44:12 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 145

type=async-response
&serial-number=00000000-0000-0000-0000-000003c3994f
&merchant-order-id=inv9157586
&paynet-order-id=1824084
&end-point-id=6655
```

7.5.7 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Fri, 26 May 2023 10:11:58 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 163

type=validation-error
&serial-number=00000000-0000-0000-0000-000002e73881
&merchant-order-id=902B4FF5
&error-message=Project+with+currency+USD+does+not+apply+request+with+currency+EUR
&error-code=20
```

7.5.8 Postman Collection

7.5.9 Request Builder

7.6 /api/v2/capture

- [Introduction](#)

- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.6.1 Introduction

Capture is initiated through HTTPS POST request to the [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. Capture may only be initiated when Preauth have final success status. See [Statuses](#).

7.6.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/capture/	https://api.doc2.com/paynet/api/v2/capture/
https://sandbox.doc2.com/paynet/api/v2/capture/	https://api.doc2.com/paynet/api/v2/capture/

7.6.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Request Parameter	Description
login	Connecting Party login name.
client_orderid	Connecting Party order identifier of the transaction for which the status is requested.
orderid	Order id assigned to the order by Doc2.0
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro. This parameter may or may not be sent.

continues on next page

Table 19 – continued from previous page

Request Parameter	Description
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents. This parameter may or may not be sent.
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Request parameter: login. 2. Request parameter: merchant-order-id. 3. Request parameter: paynet-order-id. 4. Request parameter: amount in minor units (if sent). 5. request parameter: currency (if amount sent) 6. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).

7.6.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameter	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.
end-point-id	Endpoint id used for the transaction.

7.6.5 Request Example

```
POST /paynet/api/v2/capture/39914 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 119
Content-Type: application/x-www-form-urlencoded
Connection: close

login=TestMerchant
```

(continues on next page)

(continued from previous page)

```
&client_orderid=902B4FF5
&orderid=6863088
&amount=345
&currency=USD
&control=da832c0fcff0105728e26465d5f99e6f83179a45
```

7.6.6 Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 13:15:19 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 140

type=async-response
&serial-number=00000000-0000-0000-0000-000002ddb06a
&merchant-order-id=902B4FF5
&paynet-order-id=6863088
&end-point-id=37211
```

7.6.7 Fail Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 12:59:09 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 154

pe=validation-error
&serial-number=00000000-0000-0000-0000-000002ddb06b
&error-message=Can+not+make+capture+for+order+state+AUTH_3D_VALIDATING
&error-code=27
```

7.6.8 Postman Collection

7.6.9 Request Builder

7.7 /api/v2/create-card-ref

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.7.1 Introduction

Card reference ID creation is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication.

7.7.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/create-card-ref/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/create-card-ref/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/create-card-ref/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v2/create-card-ref/group/ENDPOINTGROUPID

7.7.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Get Card Ref Request Parameters	Description
login	Connecting Party's login name.
client_orderid	Connecting Party's order identifier of the transaction.
orderid	Transaction order identifier assigned by Payment Gateway.
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Request parameter: login 2. Request parameter: client_orderid 3. Request parameter: orderid 4. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).

Connecting Party has to supply orderid and client_orderid associated with the first payment transaction. It emphasizes that the first payment is a mandatory step to process recurring payments. To authorize the credit card the information about it must be sufficient and an initial payment must be in [final status](#).

The returned Recurring Payment ID and/or Card Reference ID is a token that refers to cardholder data stored on Payment Gateway side. See the required API command to find which type of token is supported in request parameters. This ID has the following usage: 1. In direct API commands it can be passed instead of cardholder data to initiate new transaction. 2. In form API commands it can be passed to prefill the payment form with previously saved cardholder data.

7.7.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Registration Response Parameters	Description
type	The type of response. May be async-response, validation-error, error etc. If type equals validation-error or error, error-message and error-code parameters contain error details.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
card-ref-id	Card reference ID to used in subsequent recurring payments.
unq-card-ref-id	Unique card reference ID to each PAN. It can be used by Connecting Party for loyalty programs or fraud control.
status	See Status List for details.
error-message	If status is declined or error this parameter contains the reason for decline.
error-code	The error code is case of declined or error status.

7.7.5 Request Example

```
POST /paynet/api/v2/create-card-ref/46750 HTTP/1.1
Host: https://sandbox.doc2.com
User-Agent: curl/7.85.0
Accept: */*
Content-Length: 100
Content-Type: application/x-www-form-urlencoded
Connection: close

login=login
&client_orderid=902B4FF5
&orderid=6937242
&control=717349cab3fcab2c13c93ff5770ae07a7b8e0727
```

7.7.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Tue, 07 Feb 2023 08:46:24 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 144
```

(continues on next page)

(continued from previous page)

```
type=create-card-ref-response
&serial-number=00000000-0000-0000-0000-000002e110a2
&card-ref-id=1461670
&unq-card-ref-id=2463777
&status=approved
```

7.7.7 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Tue, 07 Feb 2023 08:55:26 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 123

type=validation-error
&serial-number=00000000-0000-0000-0000-000002e110ad
&error-message=INVALID_CONTROL_CODE
&error-code=2
```

7.7.8 Postman Collection

7.7.9 Request Builder

7.8 /api/v2/close-day

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.8.1 Introduction

Gate Close Day is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#).

7.8.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/close-day/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/close-day/ENDPOINTID

7.8.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Parameter Name	Description	Value
day_close_request_id	Unique request identifier. Can contain letters and numbers.	Necessity: Required Type: String Length: 8-32
sale_approved_count	The number of confirmed sale transactions.	Necessity: Optional Type: String Length: -
sale_approved_sum	The amount of confirmed sale transactions.	Necessity: Optional Type: String Length: -

continues on next page

Table 25 – continued from previous page

Parameter Name	Description	Value
reversal_approved_count	The number of confirmed reversal transactions.	Necessity: Optional Type: String Length: -
reversal_approved_sum	The amount of confirmed reversal transactions.	Necessity: Optional Type: String Length: -

7.8.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter Name	Description	Value
status	Close Day transaction status.	Type: String
gateId	Close Day Gate ID.	Type: Long
isDayClosing	Close Day mark. If true - Gate is blocked.	Type: Boolean
gateCloseDate	Close Day Gate day.	Type: String
epntId	Property assigned by the bank to the End-point when it is configured in the bank's system.	Type: Long
saleApprovedCount	The number of confirmed sale transactions.	Type: String
saleAmount	The amount of confirmed sale transactions.	Type: String
reversalApprovedCount	The number of confirmed reversal transactions.	Type: String

continues on next page

Table 26 – continued from previous page

Parameter Name	Description	Value
reversalAmount	The amount of confirmed reversal transactions.	Type: String
code39	The result of Close day. The bank's response to the making procedure of Gate close day.	Type: String
RRN	The Close Day number grouped by Endpoint ID.	Type: String
theRequestSerialNumber	Serial number of the Close Day request.	Type: String

7.8.5 Request Example

```
POST /paynet/api/v2/close-day/39914 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 119
Content-Type: application/x-www-form-urlencoded
Connection: close
```

```
"day_close_request_id": {string},
"sale_approved_count": {string},
"sale_approved_sum": {string},
"reversal_approved_count": {string},
"reversal_approved_sum": {string}
```

7.8.6 Success Response Example

Information about the closed day will appear in the response from Payment Gateway as it is received from the processors (in case of simultaneous closure of several Gates). After the Gate day closes, the response status will be set to 'finished', regardless of whether the closing of the day on all Gates was completed successfully, or part of the Gates (or all) closed with an error.

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 13:15:19 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
```

(continues on next page)

(continued from previous page)

Content-Length: 1412

```

"response": {
  "status": "finished",
  "gates": [ {
    "gateId": 193,
    "isDayClosing": false,
    "gateCloseDate": "Mon Dec 10 14:44:21 MSK 2018",
    "epntId": 1,
    "saleApprovedCount": "2334",
    "saleAmount": "87328428482.00",
    "reversalApprovedCount": "123",
    "reversalAmount": "12393423.00",
    "code39": "000",
    "RRN": "2034829434"
  }, {
    "gateId": 82,
    "isDayClosing": false,
    "gateCloseDate": "Mon Dec 10 14:44:21 MSK 2018",
    "epntId": 1,
    "saleApprovedCount": "2334",
    "saleAmount": "87328428482.00",
    "reversalApprovedCount": "123",
    "reversalAmount": "12393423.00",
    "code39": "000",
    "RRN": "2034829434"
  }, {
    "gateId": 317,
    "isDayClosing": false,
    "gateCloseDate": "Mon Dec 10 14:44:21 MSK 2018",
    "epntId": 1,
    "saleApprovedCount": "2334",
    "saleAmount": "87328428482.00",
    "reversalApprovedCount": "123",
    "reversalAmount": "12393423.00",
    "code39": "000",
    "RRN": "2034829434"
  }
  ]
},
"theRequestSerialNumber": "00000000-0000-0000-0000-00000000002c"

```

Warning: If via Close Day API the Gates remain in a locked state (in the response body for "status": "finished" for Gate - "isDayClosing": true), use the lock removal tool in Payment Gateway UI.

7.8.7 Fail Response Example

If during the Close Day procedure, Connecting Party sends a repeat request for Close Day with a new request ID, the Payment Gateway will return a response with the status 'error' and a description of the error. It is pointless to reuse the identifier of this request to further initiate the closing of the day, since the error status is the final status and the request identifier will always be associated with this request.

If Connecting Party initiates a request with a new Endpoint request ID, for which it was Close Day was already performed, and some of the Gates associated with the Endpoint closed with an error (and are in a closed state), the Payment Gateway will send a response with the status 'error' and the corresponding error description.

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 12:59:09 GMT
Content-Type: text/html;charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 154

    "response": {
      "status": "error",
      "errorCode": "99",
      "errorMessage": "gate is closing already"
    },
    "theRequestSerialNumber": "00000000-0000-0000-0000-000000000020"
```

Warning: To resolve this problem, perform the Close Day operation manually in Payment Gateway UI.

7.8.8 Request Builder

7.9 /api/v2/get-balance

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)

- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.9.1 Introduction

Get Balance is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#). Also can be viewed on UI. Please contact support managers to enable this feature.

7.9.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/get-balance/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/get-balance/ENDPOINTID

7.9.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
balance-provider	Input your balance provider.	Necessity: Required Type: String Length: 128

7.9.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
balance-amount	Current balance.

7.9.5 Request Example

```
POST /paynet/api/v2/get-balance/ HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="balance_test_
↪merchant14",oauth_signature_method="HMAC-SHA1",oauth_signature="hSXMSUPP
↪%2FIGY00WpU4LVx0mu7SA%3D"
Content-Length: 108
Content-Type: application/x-www-form-urlencoded
Connection: close

balance-provider=Test
&oauth_consumer_key=balance_test_merchant14
&oauth_nonce=2xGTFuAgetE
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1686923599
&oauth_version=1.0
```

7.9.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Tue, 16 May 2023 08:06:05 GMT
Content-Length: 192
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000

[
  {
    "balance-amount":29.99,
  }
]
```

7.9.7 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Fri, 16 Jun 2023 13:53:43 GMT
Content-Length: 42
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000

error-message=Unknown balance provider saa
```

7.9.8 Postman Collection

7.9.9 Request Builder

7.10 /api/v2/get-balance/manager

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.10.1 Introduction

Get Balance by Manager is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#). Also can be viewed on UI. Please contact technical support to enable this feature.

7.10.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/get-balance/manager/MANAGERLOGIN	https://gate.doc2.com/paynet/api/v2/get-balance/manager/MANAGERLOGIN

7.10.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
balance-name	Balance name. If this parameter is omitted, the result will give all available balances.	Necessity: Optional Type: String Length: 128

7.10.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
name	Balance name.
amount	Current balance including STH and Rolling reserve.
online-balance-live-amount	Current balance calculated from configuration excluding STH and Rolling reserve. Live = current - STH - RR.
total-short-term-hold-amount	Current amount of hold based on Date bumping function.
total-rolling-reserve-amount	Current amount calculated from rate plan hold.
currency	Currency used for selected balance.
merchantLogin	Merchant login for the balance.

continues on next page

Table 32 – continued from previous page

Response Parameters	Description
amount-buffer-hold	Calculated amount for OUT operations without final status.

7.10.5 Request Example

```
POST /paynet/api/v2/get-balance/manager/BalanceTestManager HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key=
↪ "BalanceTestManager",oauth_timestamp="1686924196",oauth_nonce="sT33IbJ4QHs",oauth_
↪ signature_method="HMAC-SHA1",oauth_signature="LyKy7M8P%2F%2FZxyLnh7ULW4050DV%3D"
Content-Length: 156
Content-Type: application/x-www-form-urlencoded
Connection: close

balance-name=saa
&oauth_consumer_key=BalanceTestManager
&oauth_nonce=sT33IbJ4QHs
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1686924196
&oauth_version=1.0
```

7.10.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Tue, 16 May 2023 08:06:05 GMT
Content-Length: 192
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000

[
  {
    "name": "saa",
    "amount": 219.500,
    "online-balance-live-amount": 219.500,
    "total-short-term-hold-amount": 0.000,
    "total-rolling-reserve-amount": 0.000,
    "currency": "USD",
    "merchantLogin": "balance_test_merchant14",
    "externalMerchantIdentifier": null,
    "amount-buffer-hold": 0.000
  }
]
```

7.10.7 Fail Response Example

```
HTTP/1.1 403
Server: server
Date: Fri, 16 Jun 2023 13:23:37 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

...
<body>
  <p>Access is denied</p>
</body>
```

7.10.8 Postman Collection

7.10.9 Request Builder

7.11 /api/v2/get-balance/merchant

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.11.1 Introduction

Get Balance by Merchant is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#). Also can be viewed on UI. Please contact support managers to enable this feature.

7.11.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/get-balance/merchant/MERCHANTLOGIN	https://gate.doc2.com/paynet/api/v2/get-balance/merchant/MERCHANTLOGIN

7.11.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
balance-name	Balance name. If this parameter is omitted, the result will give all available balances.	Necessity: Optional Type: String Length: 128

7.11.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
name	Balance name.
amount	Current balance including STH and Rolling reserve.

continues on next page

Table 35 – continued from previous page

Response Parameters	Description
online-balance-live-amount	Current balance calculated from configuration excluding STH and Rolling reserve. Live = current - STH - RR..
total-short-term-hold-amount	Current amount of hold based on Date bumping function.
total-rolling-reserve-amount	Current amount calculated from rate plan hold.
currency	Currency used for selected balance.
amount-buffer-hold	Calculated amount for OUT operations without final status.

7.11.5 Request Example

```
POST /paynet/api/v2/get-balance/merchant/balance_test_merchant14 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="balance_test_
↪merchant14",oauth_timestamp="1686924613",oauth_nonce="qoJPcins4Aw",oauth_signature_
↪method="HMAC-SHA1",oauth_signature="M6fTXbI0tFz4Dy7YEL3SBzCN1I0%3D"
Content-Length: 161
Content-Type: application/x-www-form-urlencoded
Connection: close

balance-name=saa
&oauth_consumer_key=balance_test_merchant14
&oauth_nonce=qoJPcins4Aw
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1686924613
&oauth_version=1.0
```

7.11.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Tue, 16 May 2023 08:06:05 GMT
Content-Length: 192
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000

[
  {
    "name": "Test1",
    "amount": 8854.000,
    "online-balance-live-amount": 8854.000,
    "total-short-term-hold-amount": 0.000,
    "total-rolling-reserve-amount": 0.000,
    "currency": "AUD",
```

(continues on next page)

(continued from previous page)

```
    "amount-buffer-hold":0.000  
  }  
]
```

7.11.7 Fail Response Example

```
HTTP/1.1 200  
Server: server  
Date: Tue, 16 May 2023 08:06:05 GMT  
Content-Length: 192  
Connection: close  
X-XSS-Protection: 1  
X-Content-Type-Options: nosniff  
Strict-Transport-Security: max-age=31536000  
Strict-Transport-Security: max-age=31536000  
  
...  
<body>  
  <p>Access is denied</p>  
</body>
```

7.11.8 Postman Collection

7.11.9 Request Builder

7.12 /api/v2/get-exchange

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.12.1 Introduction

Get exchange rate is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#).

7.12.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/get-exchange/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/get-exchange/ENDPOINTID

7.12.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
exchange-provider	Exchange provider.	Necessity: Required Type: String Length: 128
exchange-currency-code-from	Currency code that need to exchange (See: Currency codes).	Necessity: Required Type: String Length: 128
exchange-currency-code-to	Currency code that need to get (See: Currency codes).	Necessity: Required Type: String Length: 128

7.12.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter Name	Description
exchange-low-rate	Provider's buy rate.
exchange-high-rate	Provider's sell rate.

7.12.5 Request Example

```
POST /paynet/api/v2/get-exchange/39790 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="test_merch",
  ↳oauth_timestamp="1684995901",oauth_nonce="1kKy3hMlAe1",oauth_signature_method="HMAC-
  ↳SHA1",oauth_signature="jGUaNFqLvB0lgGgWag%2FSU3wMnUc%3D"
Content-Length: 222
Content-Type: application/x-www-form-urlencoded
Connection: close

exchange-currency-code-from=USD
&exchange-currency-code-to=RUB
&exchange-provider=test
&oauth_consumer_key=test_merch
&oauth_nonce=1kKy3hMlAe1
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1684995901
&oauth_version=1.0
```

7.12.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 25 May 2023 06:27:46 GMT
Content-Length: 48
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000

exchange-low-rate=10.00
&exchange-high-rate=20.00
```


7.12.7 Fail Response Example

```
HTTP/1.1 403
Server: server
Date: Thu, 25 May 2023 07:02:59 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

error-message=Book TCurrencyBook was not found by code USB
```

7.12.8 Postman Collection

7.12.9 Request Builder

7.13 /api/v2/get-card-info

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.13.1 Introduction

For integration purposes use staging environment HTTPS POST request to the [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.13.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/get-card-info/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/get-card-info/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/get-card-info/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v2/get-card-info/group/ENDPOINTGROUPID

7.13.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Parameter Name	Description	Value
login	Connecting Party's login name.	Necessity: Mandatory Type: String Length: 20
cardrefid	Card reference ID obtained at Card Registration Stage .	Necessity: Mandatory Type: String Length: 20
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Request parameter: login 2. Request parameter: cardrefid 3. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Mandatory Type: String Length: 128

7.13.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Information Re- sponse Parameters	Description
type	The type of response. May be get-card-info-response, validation-error, error. If type equals error, error-message and error-code parameters contain error details.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
card-printed-name	Card holder's name.
bin	Bank Identification Number.
last-four-digits	The last four digits of PAN.
expire-year	Card expiration year.
expire-month	Card expiration month.
result-3d	3DS completion result.
error-message	If status is validation-error or error this parameter contains the reason for decline or error details.
error-code	The error code is case of validation-error or error status.

7.13.5 Request Example

```
POST /paynet/api/v2/get-card-info/46750 HTTP/1.1
Host: https://sandbox.doc2.com
User-Agent: curl/7.85.0
Accept: */
Content-Length: 77
Content-Type: application/x-www-form-urlencoded
Connection: close

&login=Test
&cardrefid=1461665
&control=f8ffd7dac9e334799862da0e9190387510d43ff7
```

7.13.6 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Thu, 02 Feb 2023 11:21:39 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 193

type=get-card-info-response
&serial-number=00000000-0000-0000-0000-000002e0d218
&card-printed-name=CARD+HOLDER
&bin=453897
&last-four-digits=6732
&expire-year=2099
&expire-month=12
&result-3d=F

```

7.13.7 Fail Response Example

```

HTTP/1.1 200
Server: server
Date: Thu, 02 Feb 2023 11:26:22 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 136

type=validation-error
&serial-number=00000000-0000-0000-0000-000002e0d21f
&error-message=End+point+with+id+99999+not+found
&error-code=3

```

7.13.8 Postman Collection

7.13.9 Request Builder

7.14 /api/v2/make-rebill-preauth

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Request Parameters for Insurance](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.14.1 Introduction

Preauth recurring transactions are initiated through HTTPS POST request to the [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.14.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/make-rebill-preauth/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/make-rebill-preauth/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/make-rebill-preauth/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v2/make-rebill-preauth/group/ENDPOINTGROUPID

7.14.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Parameter Name	Description	Value
login	Connecting Party's login name.	Necessity: Required Type: String Length: 20
client_orderid	Connecting Party's order identifier of the transaction.	Necessity: Required Type: String Length: 128
cardrefid	Card reference ID obtained at Card Registration Stage /api/v2/create-card-ref/ .	Necessity: Required Type: String Length: 20
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
enumerate_amounts	This parameter may comprise multiple amounts, separated with ,. Doc2.0 will cycle through the amounts of the list, try to make a payment on that amount, until there are no more amounts from the list, or not get approved.	Necessity: Optional Type: String Length: 128

continues on next page

Table 43 - continued from previous page

Parameter Name	Description	Value
recurrent_scenario	Type of the rebill transaction. Possible values: REGULAR or IRREGULAR. If this parameter is sent in the request, it will have priority over the same parameter specified on gate level. Only used for specific acquirers.	Necessity: Optional Type: String Length: 50
recurrent_initiator	Initiator of the rebill transaction. Possible values: CARDHOLDER or MERCHANT. If this parameter is sent in the request, it will have priority over the same parameter specified on gate level. Only used for specific acquirers.	Necessity: Optional Type: String Length: 50
cvv2	Customer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card. May be empty or absent if bank gateway supports processing without CVV2.	Necessity: Optional Type: Numeric Length: 3-4
ipaddress	Customer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45
comment	A short comment.	Necessity: Optional Type: String Length: 50
order_desc	Order description.	Necessity: Required Type: String Length: 64k

continues on next page

Table 43 – continued from previous page

Parameter Name	Description	Value
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Request parameter: login 2. Request parameter: client_orderid 3. Request parameter: cardrefid 4. Request parameter: amount in minor units (if sent). 5. request parameter: currency 6. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
purpose	Destination to where the payment goes. It is useful for the Connecting Party who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999; mail@example.com etc. This value will be used by fraud monitoring system.	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. Parameter is required for 3DS flow and optional for non-3DS. https://doc2.codetime.net can be used for testing purposes if it's unknown whether 3DS is used or not.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 43 - continued from previous page

Parameter Name	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 43 - continued from previous page

Parameter Name	Description	Value
merchant_data	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64

7.14.4 Request Parameters for Insurance

Note:

The acquirer can override the mandatory nature of some fields to make them mandatory instead of optional.

Leading and trailing whitespace in input parameters will be trimmed.

Parameter Name	Description	Value
insurance_amount	The insurance amount. The amount must be specified in minimum units with a . delimiter. For example, 100.5 in USD means 100 US dollars and 50 cents.	Type: Numeric Length: 10
insured_person_first_name	Insured person's first name.	Type: String Length: 256
insured_person_last_name	Insured person's last name.	Type: String Length: 256
insured_person_middle_name	Insured person's middle name.	Type: String Length: 256
insured_person_birthday	Insured person's date of birth.	Type: String Length: 256

continues on next page

Table 44 – continued from previous page

Parameter Name	Description	Value
insured_person_document_series	Insured person's document series.	Type: String Length: 256
insured_person_document_number	Insured person's document number.	Type: String Length: 256
insured_person_document_issue_date	Date the insured person's document was issued.	Type: String Length: 256
insured_person_document_issuer_name	Name of the authority that issued the insured person's document.	Type: String Length: 256
insured_person_document_issuer_code	Code of the authority that issued the insured person's document.	Type: String Length: 256
insured_person_registration_address	Insured person's registration address.	Type: String Length: 256
insured_person_phone	Full international phone number of the insured person.	Type: String Length: 256
insured_person_email	Insured person's email address.	Type: String Length: 256

continues on next page

Table 44 – continued from previous page

Parameter Name	Description	Value
card_insurance_agreement_number	Agreement number.	Type: String Length: 256
card_insurance_agreement_sell_date	Sale date of the agreement.	Type: String Length: 256
card_insurance_agreement_start_date	Agreement start date.	Type: String Length: 256
card_insurance_agreement_end_date	Agreement end date.	Type: String Length: 256
card_insurance_agreement_amount	Insurance amount.	Type: String Length: 256
card_insurance_agreement_bonus	Insurance premium.	Type: String Length: 256

7.14.5 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Sale Response Parameters	Description
type	The type of response. May be async-response, validation-error, error etc. If type equals validation-error or error, error-message and error-code parameters contain error details.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
merchant-order-id	Connecting Party's order id.
paynet-order-id	Order id assigned to the order by Doc2.0.
end-point-id	ID of the used End Point.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.14.6 Request Example

```
POST /paynet/api/v2/make-rebill-preauth/46750 HTTP/1.1
Host: https://sandbox.doc2.com
User-Agent: curl/7.85.0
Accept: */
Content-Length: 229
Content-Type: application/x-www-form-urlencoded
Connection: close

&login=login
&client_orderid=902B4FF5
&cardrefid=1461665
&amount=5.00
&currency=USD
&cvv2=321
&ipaddress=34.129.65.12
&comment=Information about Rebill
&order_desc=Rebill order description
&control=a37f4972233b4a5dbfb4dcaae149ce7feed01ef9
```

7.14.7 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 02 Feb 2023 13:10:44 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 144
```

(continues on next page)

(continued from previous page)

```
&type=async-response
&serial-number=00000000-0000-0000-0000-000002e0d498
&merchant-order-id=902B4FF5
&paynet-order-id=6937030
&end-point-id=46750
```

7.14.8 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 02 Feb 2023 13:18:50 GMT
Content-Type: text/html;charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 164

type=validation-error
&serial-number=00000000-0000-0000-0000-000002e0d614
&merchant-order-id=902B4FF5
&error-message=End+point+with+id+99999+not+found
&error-code=3
```

7.14.9 Postman Collection

7.14.10 Request Builder

7.15 /api/v2/make-rebill-sale

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Request Parameters for Insurance](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)

- [Request Builder](#)

7.15.1 Introduction

Recurring transactions are initiated through HTTPS POST request to the [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.15.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/make-rebill-sale/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/make-rebill-sale/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/make-rebill-sale/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v2/make-rebill-sale/group/ENDPOINTGROUPID

7.15.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Parameter Name	Description	Value
login	Connecting Party's login name.	Necessity: Required Type: String Length: 20
client_orderid	Connecting Party's order identifier of the transaction.	Necessity: Required Type: String Length: 128

continues on next page

Table 47 – continued from previous page

Parameter Name	Description	Value
cardrefid	Card reference ID obtained at Card Registration Stage /api/v2/create-card-ref/ .	Necessity: Required Type: String Length: 20
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
enumerate_amounts	This parameter may comprise multiple amounts, separated with ,. Doc2.0 will cycle through the amounts of the list, try to make a payment on that amount, until there are no more amounts from the list, or not get approved.	Necessity: Optional Type: String Length: 128
recurrent_scenario	Type of the rebill transaction. Possible values: REGULAR or IRREGULAR. If this parameter is sent in the request, it will have priority over the same parameter specified on gate level. Only used for specific acquirers.	Necessity: Optional Type: String Length: 50
recurrent_initiator	Initiator of the rebill transaction. Possible values: CARDHOLDER or MERCHANT. If this parameter is sent in the request, it will have priority over the same parameter specified on gate level. Only used for specific acquirers.	Necessity: Optional Type: String Length: 50
cvv2	Customer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card. May be empty or absent if bank gateway supports processing without CVV2.	Necessity: Optional Type: Numeric Length: 3-4
ipaddress	Customer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45

continues on next page

Table 47 – continued from previous page

Parameter Name	Description	Value
comment	A short comment.	Necessity: Optional Type: String Length: 50
order_desc	Order description.	Necessity: Required Type: String Length: 64k
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Request parameter: login 2. Request parameter: client_orderid 3. Request parameter: cardrefid 4. Request parameter: amount in minor units (if sent). 5. request parameter: currency 6. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
purpose	Destination to where the payment goes. It is useful for the Connecting Party who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999; mail@example.com etc. This value will be used by fraud monitoring system.	Necessity: Optional Type: String Length: 128

continues on next page

Table 47 - continued from previous page

Parameter Name	Description	Value
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. Parameter is required for 3DS flow and optional for non-3DS. https://doc2.codetime.net can be used for testing purposes if it's unknown whether 3DS is used or not.	Necessity: Optional Type: String Length: 1024
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 47 - continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
merchant_data	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64

7.15.4 Request Parameters for Insurance

Note:

The acquirer can override the mandatory nature of some fields to make them mandatory instead of optional.

Leading and trailing whitespace in input parameters will be trimmed.

Parameter Name	Description	Value
insurance_amount	The insurance amount. The amount must be specified in minimum units with a . delimiter. For example, 100.5 in USD means 100 US dollars and 50 cents.	Type: Numeric Length: 10

continues on next page

Table 48 – continued from previous page

Parameter Name	Description	Value
insured_person_first_name	Insured person's first name.	Type: String Length: 256
insured_person_last_name	Insured person's last name.	Type: String Length: 256
insured_person_middle_name	Insured person's middle name.	Type: String Length: 256
insured_person_birthday	Insured person's date of birth.	Type: String Length: 256
insured_person_document_series	Insured person's document series.	Type: String Length: 256
insured_person_document_number	Insured person's document number.	Type: String Length: 256
insured_person_document_issue_date	Date the insured person's document was issued.	Type: String Length: 256
insured_person_document_issuer_name	Name of the authority that issued the insured person's document.	Type: String Length: 256

continues on next page

Table 48 – continued from previous page

Parameter Name	Description	Value
insured_person_document_issuer_code	Code of the authority that issued the insured person's document.	Type: String Length: 256
insured_person_registration_address	Insured person's registration address.	Type: String Length: 256
insured_person_phone	Full international phone number of the insured person.	Type: String Length: 256
insured_person_email	Insured person's email address.	Type: String Length: 256
card_insurance_agreement_number	Agreement number.	Type: String Length: 256
card_insurance_agreement_sell_date	Sale date of the agreement.	Type: String Length: 256
card_insurance_agreement_start_date	Agreement start date.	Type: String Length: 256
card_insurance_agreement_end_date	Agreement end date.	Type: String Length: 256

continues on next page

Table 48 – continued from previous page

Parameter Name	Description	Value
card_insurance_agreement_amount	Insurance amount.	Type: String Length: 256
card_insurance_agreement_bonus	Insurance premium.	Type: String Length: 256

7.15.5 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Sale Response Parameters	Description
type	The type of response. May be async-response, validation-error, error etc. If type equals validation-error or error, error-message and error-code parameters contain error details.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
merchant-order-id	Connecting Party's order ID.
paynet-order-id	Order id assigned to the order by Doc2.0.
end-point-id	ID of the used End Point.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.15.6 Request Example

```
POST /paynet/api/v2/make-rebill-sale/46750 HTTP/1.1
Host: https://sandbox.doc2.com
User-Agent: curl/7.85.0
Accept: */
Content-Length: 229
Content-Type: application/x-www-form-urlencoded
Connection: close
```

(continues on next page)

(continued from previous page)

```
&login=login
&client_orderid=902B4FF5
&cardrefid=1461665
&amount=5.00
&currency=USD
&cvv2=123
&ipaddress=34.129.65.12
&comment=Information about Rebill
&order_desc=Rebill order description
&control=a37f4972233b4a5dbfb4dcaae149ce7feed01ef9
```

7.15.7 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 02 Feb 2023 13:22:04 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 144

type=async-response
&serial-number=00000000-0000-0000-0000-000002e0d6b9
&merchant-order-id=902B4FF5
&paynet-order-id=6937242
&end-point-id=46750
```

7.15.8 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 02 Feb 2023 13:24:47 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 164

type=validation-error
&serial-number=00000000-0000-0000-0000-000002e0d74c
&merchant-order-id=902B4FF5
&error-message=End+point+with+id+99999+not+found
```

(continues on next page)

(continued from previous page)

`&error-code=3`

7.15.9 Postman Collection

7.15.10 Request Builder

7.16 /api/v2/pan-eligibility

- [Introduction](#)
- [PAN Eligibility API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Send Request Example](#)
- [Send Success Response Example](#)
- [Send Fail Response Example](#)
- [Receive Request Example](#)
- [Receive Success Response Example](#)
- [Receive Fail Response Example](#)
- [Both Request Example](#)
- [Both Success Response Example](#)
- [Both Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.16.1 Introduction

PAN Eligibility is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#).

7.16.2 PAN Eligibility API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Method	Integration	Production
Sending-only	https://sandbox.doc2.com/paynet/api/v2/pan-eligibility/send/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/pan-eligibility/send/ENDPOINTID
Receiving-only	https://sandbox.doc2.com/paynet/api/v2/pan-eligibility/receive/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/pan-eligibility/receive/ENDPOINTID
Both	https://sandbox.doc2.com/paynet/api/v2/pan-eligibility/both/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/pan-eligibility/both/ENDPOINTID

7.16.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Sending-only

Parameter	Description	Value
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
sending-card-number	Sender`s credit card number. (if sending-card-ref-id is not specified)	Necessity: Required Type: Numeric Length: 19
card-printed-name	Sender`s card printed name. (if sending-card-ref-id is not specified)	Necessity: Required Type: String Length: 128

continues on next page

Table 51 – continued from previous page

Parameter	Description	Value
expire_month	Sender credit card's month of expiration. (if sending-card-ref-id is not specified)	Necessity: Required Type: Numeric Length: 2
expire_year	Sender credit card's year of expiration. (if sending-card-ref-id is not specified)	Necessity: Required Type: Numeric Length: 4
sending-card-ref-id	Sender's card ref ID. (if sending-card-number is not specified)	Necessity: Required Type: Numeric Length: 20
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 51 – continued from previous page

Parameter	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

Receiving-only

Parameter	Description	Value
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
receiving-card-number	Receiver`s credit card number. (if receiving-card-ref-id is not specified)	Necessity: Required Type: Numeric Length: 19
receiving-card-ref-id	Receiver`s card ref ID. (if receiving-card-number is not specified)	Necessity: Required Type: Numeric Length:

continues on next page

Table 52 – continued from previous page

Parameter	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters. This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters. This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

Both

Parameter	Description	Value
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
sending-card-number	Sender`s card number. (if sending-card-ref-id is not specified)	Necessity: Required Type: Numeric Length: 19
card-printed-name	Sender`s card printed name. (if sending-card-ref-id is not specified)	Necessity: Required Type: String Length: 128
expire_month	Sender credit card`s month of expiration. (if sending-card-ref-id is not specified)	Necessity: Required Type: Numeric Length: 2
expire_year	Sender credit card`s year of expiration. (if sending-card-ref-id is not specified)	Necessity: Required Type: Numeric Length: 4
sending-card-ref-id	Sender`s card ref ID. (if sending-card-ref-id is not specified)	Necessity: Required Type: Numeric Length: 20
receiving-card-number	Receiver`s card number. (if receiving-card-ref-id is not specified)	Necessity: Required Type: Numeric Length: 19
receiving-card-ref-id	Receiver`s card ref ID. (if receiving-card-number is not specified)	Necessity: Required Type: Numeric Length:

continues on next page

Table 53 - continued from previous page

Parameter	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters. This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters. This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

7.16.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. May be async-response, validation-error, error etc. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.16.5 Send Request Example

```
POST /paynet/api/v2/pan-eligibility/send/39907 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="test_merch",
→oauth_timestamp="1687507082",oauth_nonce="XQkXQgLxhCE",oauth_signature_method="HMAC-
→SHA1",oauth_signature="lLTi0EyUY%2FwibmGVA%2F5rY6qIS9I%3D"
Content-Length: 192
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=1234567890
&oauth_consumer_key=test_merch
&oauth_nonce=XQkXQgLxhCE
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1687507082
&oauth_version=1.0
&sending-card-ref-id=1461790
```

7.16.6 Send Success Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 21 Jun 2023 11:19:20 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 147

type=async-response
&serial-number=00000000-0000-0000-0000-000002ecbcb
&merchant-order-id=1234567890
&paynet-order-id=7216063
&end-point-id=39907
```

7.16.7 Send Fail Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 11:19:20 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 154

...
<body>
  <p>Access is denied</p>
</body>
```

7.16.8 Receive Request Example

```
POST /paynet/api/v2/pan-eligibility/receive/39907 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="test_merch",
  ↳oauth_timestamp="1687507082",oauth_nonce="XQkXQgLxhCE",oauth_signature_method="HMAC-
  ↳SHA1",oauth_signature="lLTi0EyUY%2FwibmGVA%2F5rY6qIS9I%3D"
Content-Length: 192
Content-Type: application/x-www-form-urlencoded
```

(continues on next page)

(continued from previous page)

```

Connection: close

client_orderid=1234567890
&oauth_consumer_key=test_merch
&oauth_nonce=XQkXQgLxhCE
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1687507082
&oauth_version=1.0
&receiving-card-number=4444555566661111

```

7.16.9 Receive Success Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 21 Jun 2023 12:19:20 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 147

type=async-response
&serial-number=00000000-0000-0000-0000-000002ecbcb
&merchant-order-id=1234567890
&paynet-order-id=7216063
&end-point-id=39907

```

7.16.10 Receive Fail Response Example

```

HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 12:19:29 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 154

...
<body>
  <p>Access is denied</p>
</body>

```

7.16.11 Both Request Example

```
POST /paynet/api/v2/pan-eligibility/both/39907 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="test_merch",
  ↳oauth_timestamp="1687513856",oauth_nonce="T0z9qml2rDZ",oauth_signature_method="HMAC-
  ↳SHA1",oauth_signature="rsHewN7TRIxZPQAtd99z5eoSyJY%3D"
Content-Length: 231
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=1234567890
&oauth_consumer_key=test_merch
&oauth_nonce=T0z9qml2rDZ
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1687513856
&oauth_version=1.0
&receiving-card-number=4444555566661111
&sending-card-ref-id=1461790
```

7.16.12 Both Success Response Example

```
HTTP/1.1 200
Server: server
Date: Fri, 23 Jun 2023 09:51:15 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 146

type=async-response
&serial-number=000000000-0000-0000-0000-000002ecbcc2
&merchant-order-id=1234567890
&paynet-order-id=7216068
&end-point-id=39907
```

7.16.13 Both Fail Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 13:19:29 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
```

(continues on next page)

(continued from previous page)

```
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 154

...
<body>
  <p>Access is denied</p>
</body>
```

7.16.14 Postman Collection

7.16.15 Request Builder

7.17 /api/v2/pan-eligibility/form

- [Introduction](#)
- [PAN Eligibility API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Send-form Request Example](#)
- [Send-form Success Response Example](#)
- [Send-form Fail Response Example](#)
- [Receive-form Request Example](#)
- [Receive-form Success Response Example](#)
- [Receive-form Fail Response Example](#)
- [Both-form Request Example](#)
- [Both-form Success Response Example](#)
- [Both-form Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.17.1 Introduction

PAN Eligibility form is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#).

7.17.2 PAN Eligibility API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Method	Integration	Production
Sending-only	https://sandbox.doc2.com/paynet/api/v2/pan-eligibility/send-form/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/pan-eligibility/send-form/ENDPOINTID
Receiving-only	https://sandbox.doc2.com/paynet/api/v2/pan-eligibility/receive-form/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/pan-eligibility/receive-form/ENDPOINTID
Both	https://sandbox.doc2.com/paynet/api/v2/pan-eligibility/both-form/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/pan-eligibility/both-form/ENDPOINTID

7.17.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128

continues on next page

Table 56 – continued from previous page

Parameter Name	Description	Value
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status. Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if you have no need to return payer anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters. This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 128
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters. This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 128

7.17.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. May be async-response, validation-error, error etc. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.17.5 Send-form Request Example

```
POST /paynet/api/v2/pan-eligibility/send-form/39907 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="javid_test_merch
→",oauth_timestamp="1687507473",oauth_nonce="Gs2vcwIXv6H",oauth_signature_method=
→"HMAC-SHA1",oauth_signature="sMQWXIUY00TgBzt20avSFhT3Q3s%3D"
Content-Length: 192
Content-Type: application/x-www-form-urlencoded
Connection: close

client-order-id=1234567890
&oauth_consumer_key=test_merch
&oauth_nonce=Gs2vcwIXv6H
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1687507473
&oauth_version=1.0
&redirect-url=http%3A%2F%2Fsandbox.doc2.com%2Fdoc%2Fdummy.htm
```

7.17.6 Send-form Success Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 21 Jun 2023 11:19:20 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 147

type=async-response
&serial-number=00000000-0000-0000-0000-000002ec863e
&merchant-order-id=1234567890
&paynet-order-id=7211136
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Fpan-eligibility%2Finit
%2FBB3757784F4A53585143535A554E2F4D6A3264715A616E744C6C31667043414962654534575039376E636F343D

```

7.17.7 Send-form Fail Response Example

```

HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 12:59:09 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 154

...
<body>
  <p>Access is denied</p>
</body>

```

7.17.8 Receive-form Request Example

```

POST /paynet/api/v2/pan-eligibility/receive-form/39907 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="javid_test_merch
",oauth_timestamp="1687511712",oauth_nonce="cRmRev3FdaP",oauth_signature_method=
"HMAC-SHA1",oauth_signature="6zWG0Sli0dehdFikhoGdoEqhcQA%3D"
Content-Length: 226

```

(continues on next page)

(continued from previous page)

```

Content-Type: application/x-www-form-urlencoded
Connection: close

client-order-id=1234567890
&oauth_consumer_key=test_merch
&oauth_nonce=cRmRev3FdaP
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1687511712
&oauth_version=1.0
&redirect-url=http%3A%2F%2Fsandbox.doc2.com%2Fdoc%2Fdummy.htm

```

7.17.9 Receive-form Success Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 21 Jun 2023 12:19:20 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 147

type=async-response
&serial-number=00000000-0000-0000-0000-000002ec863e
&merchant-order-id=1234567890
&paynet-order-id=7211136
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Fpan-eligibility%2Finit
↪%2FBB3757784F4A53585143535A554E2F4D6A3264715A616E744C6C316670434149626545436159574254497A4D3D

```

7.17.10 Receive-form Fail Response Example

```

HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 12:59:09 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 154

...
<body>
  <p>Access is denied</p>
</body>

```


7.17.11 Both-form Request Example

```
POST /paynet/api/v2/pan-eligibility/both-form/39907 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="javid_test_merch
↪",oauth_timestamp="1687512444",oauth_nonce="Nu8SL8l70H2",oauth_signature_method=
↪"HMAC-SHA1",oauth_signature="wypHgZYxmKrdshhj1CabwPGnnA%3D"
Content-Length: 226
Content-Type: application/x-www-form-urlencoded
Connection: close

client-order-id=1234567890
&oauth_consumer_key=test_merch
&oauth_nonce=G52vcwIXv6H
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1687507765
&oauth_version=1.0
&redirect-url=http%3A%2F%2Fsandbox.doc2.com%2Fdoc%2Fdummy.htm
```

7.17.12 Both-form Success Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 21 Jun 2023 13:19:20 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 147

type=async-response
&serial-number=00000000-0000-0000-0000-000002ec863e
&merchant-order-id=1234567890
&paynet-order-id=7211136
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Fpan-eligibility%2Finit
↪%2FBB3757784F4A53585143535A554E2F4D6A3264715A616E744C6C31667043414962654534575039376E636F343D
```

7.17.13 Both-form Fail Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 22 Aug 2022 13:19:09 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
```

(continues on next page)

(continued from previous page)

```
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 154

...
<body>
  <p>Access is denied</p>
</body>
```

7.17.14 Postman Collection

7.17.15 Request Builder

7.18 /api/v2/preauth

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Additional Fields for Preauth Transactions](#)
 - [For Connecting Party](#)
 - [For Payment Institutions](#)
- [Response Parameters](#)
- [Request with Cardholder Data Example](#)
- [Request with Card Recurring Payment ID Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.18.1 Introduction

Preauth is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.18.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/preauth/ENDPOINTID	https://doc2.com/paynet/api/v2/preauth/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/preauth/group/ENDPOINTID	https://doc2.com/paynet/api/v2/preauth/group/ENDPOINTID

7.18.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128
order_desc	Brief order description.	Necessity: Required Type: String Length: 125
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3

continues on next page

Table 59 - continued from previous page

Parameter Name	Description	Value
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45

continues on next page

Table 59 - continued from previous page

Parameter Name	Description	Value
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL). 2. Request parameter: client_orderid. 3. Request parameter: amount in minor units (if sent). 4. Request parameter: email. 5. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
cvv2	Payer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number printed on the back of the card in the signature area.	Necessity: Required Type: Numeric Length: 3-4
credit_card_number	Payer's bank card number (also known as PAN - Primary Account Number).	Necessity: Required Type: Numeric Length: 20
card_recurring_payment_id	Payer's tokenized cardholder's data ID. Send either card_recurring_payment_id or combination of credit_card_number, card_printed_name, expire_month and expire_year, not all. To create card_recurring_payment_id see /api/v2/create-card-ref .	Necessity: Required Type: Long Length: 20
card_printed_name	Cardholder name, printed on the bank card.	Necessity: Required Type: String Length: 64k
expire_month	Bank card expiration month.	Necessity: Required Type: Numeric Length: 2
expire_year	Bank card expiration year.	Necessity: Required Type: Numeric Length: 4

continues on next page

Table 59 - continued from previous page

Parameter Name	Description	Value
first_name	Payer's first name.	Necessity: Required Type: String Length: 50
last_name	Payer's last name.	Necessity: Required Type: String Length: 50
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Conditional Type: String Length: 2-3
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status. Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize <code>server_callback_url</code> or status API command. Pass <code>https://doc2.codetime.net</code> if you have no need to return payer anywhere. Use either <code>redirect_url</code> or combination of <code>redirect_success_url</code> and <code>redirect_fail_url</code>, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 59 - continued from previous page

Parameter Name	Description	Value
redirect_success_url	URL, where the Payer is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
redirect_fail_url	URL, where the Payer is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
ssn	Last four digits of the payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYYMMDD.	Necessity: Optional Type: Numeric Length: 8
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15

continues on next page

Table 59 - continued from previous page

Parameter Name	Description	Value
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128
purpose	Destination to where the payment goes. It is useful for the Connecting Partys who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 59 - continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters. This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
merchant_data	Custom Connecting Party details, which can be attached to the transaction and passed back in the status response, Connecting Party callback parameters or server_callback_url. Additional information of transaction which may be useful in Connecting Party's external system, e.g. VIP customer, TV promo campaign lead. Information returns in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
mini-mum_transaction_amount	This parameter can be used to limit the minimum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10
maxi-mum_transaction_amount	This parameter can be used to limit the maximum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10

continues on next page

Table 59 - continued from previous page

Parameter Name	Description	Value
customer_level	Customer level in CMS system	Necessity: Optional Type: Varchar Length: 32
customer_id	Customer ID in CMS system. Required if transaction goes via CMS (Payment Gateway mode)	Necessity: Optional Type: Int Length: 10
merchant_customer_id	Merchant Customer ID in CMS system. Required if transaction goes via CMS (CRM mode)	Necessity: Optional Type: Varchar Length: 64
recurring-payment-id	Recurring Payment ID can be sent instead of cardholder data. CVV for native is not needed. Customer Data can be updated via /api/v4/update-recurring-payment/ . Recurring Payment ID creation is initiated through HTTPS POST request by using URLs and the parameters specified below. Use OAuth RSA-SHA256 for authentication.	Necessity: Conditional Type: Long

7.18.4 Additional Fields for Preauth Transactions

For Connecting Party

Note:

Browser data for 3DS 2.X is gathered by Doc2.0 system on 3DS authentication stage. For some processing channels, however, the browser data and/or Connecting Party URL for 3DS challenge results must be provided in initial transaction request. Please contact Support manager to clarify if these parameters should be included in request parameters.

The Connecting Party's site needs to accurately populate the browser information for each transaction. This data can be obtained by Connecting Party's servers. Ensure that the data is not altered or hard-coded, and that it is unique to each transaction.

Parameter Name	Description	Value
ipaddress	IP address of the browser as returned by the HTTP headers to the 3DS Requestor.	Necessity: Required Type: String Length: 45
customer_browser_accept_header	Exact content of the HTTP accept headers as sent to the 3DS Requestor from the Cardholder's browser.	Necessity: Required Type: String Length: 2048
customer_browser_javascript_enabled	Boolean that represents the ability of the cardholder browser to execute JavaScript.	Necessity: Required Type: Boolean Length: -
customer_browser_accept_language	Value representing the browser language as defined in IETF BCP47.	Necessity: Required Type: String Length: 8
customer_browser_user_agent	Exact content of the HTTP user-agent header.	Necessity: Required Type: String Length: 2048
tds_areq_notification_url, alias tds_cres_notification_url	Fully qualified URL of Connecting Party system that will receive the CRes message or Error Message. This CRes message must be sent to Doc2.0. See details here Upload CRes Result .	Necessity: Optional Type: String Length: 256
customer_browser_info	If true, then the fields below must be present.	Necessity: Optional Type: Boolean Length: -
customer_browser_color_depth	Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: String Length: 2

continues on next page

Table 60 – continued from previous page

Parameter Name	Description	Value
cus-tomer_browser_java_enabled	Boolean that represents the ability of the cardholder browser to execute Java. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: Boolean Length: -
cus-tomer_browser_screen_height	Total height of the Cardholder's screen in pixels. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: Numeric Length: 6
cus-tomer_browser_screen_width	Total width of the cardholder's screen in pixels. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: Numeric Length: 6
cus-tomer_browser_time_zone	Time-zone offset in minutes between UTC and the Cardholder browser local time. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: String Length: 5

For Payment Institutions

The PSP or Acquirer can fill the 3DS results for each transaction, if 3DS authentication is performed on their side.

Parameter Name	Description	Value
tds_authentication_result_type	Type of result. Possible values are: - SIMPLE	Type: String Length: 6

continues on next page

Table 61 – continued from previous page

Parameter Name	Description	Value
tds_authentication_result_authentication_type	Authentication Type. Indicates the type of authentication method the Issuer will use to challenge the Cardholder, whether in the ARes message or what was used by the ACS when in the RReq message. Possible values are: - 01 = Static - 02 = Dynamic - 03 = OOB - 04 = Decoupled - 05-79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) - 80-99 = Reserved for DS use	Type: String Length: 2
tds_authentication_result_authentication_value	Authentication Value. Payment System-specific value provided by the ACS or the DS using an algorithm defined by Payment System. Authentication Value may be used to provide proof of authentication. A 20-byte value that has been Base64 encoded, giving a 28-byte result	Type: String Length: 19-28
tds_authentication_result_transaction_id	transaction_id.0.2 or dsTransID for 2.1.0/2.2.0	Type: String Length: 19-36

continues on next page

Table 61 – continued from previous page

Parameter Name	Description	Value
tds_authentication_result_transaction_status	Transaction Status. Indicates whether a transaction qualifies as an authenticated transaction or account verification. Possible values are: - Y = Authentication Verification Successful - N = Not Authenticated/Account Not Verified, Transaction denied - U = Authentication/Account Verification Could Not Be Performed, Technical or other problem, as indicated in ARes or RReq - A = Attempts Processing Performed, Not Authenticated/Verified, but a proof of attempted authentication/verification is provided - C = Challenge Required, Additional authentication is required using the CReq/CRes - D = Challenge Required, Decoupled Authentication confirmed - R = Authentication/ Account Verification Rejected, Issuer is rejecting	Type: String Length: 1
tds_authentication_result_message_version	Message Version Number. Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction. Possible values are: - 1.0.2 - 2.1.0 - 2.2.0	Type: String Length: 5

7.18.5 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. May be async-response, validation-error, error etc. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.
end-point-id	Endpoint id used for the transaction.

7.18.6 Request with Cardholder Data Example

```
POST /paynet/api/v2/preauth/39549 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 314
Content-Type: application/x-www-form-urlencoded
Connection: close
```

```
credit_card_number=4538977399606732
&card_printed_name=CARD HOLDER
&expire_month=12
&expire_year=2099
&cvv2=123
&client_orderid=902B4FF5
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100 Main st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=%2B12063582043
&cell_phone=%2B19023384543
```

(continues on next page)

(continued from previous page)

```

&email=john.smith@gmail.com
&currency=USD
&amount=10.42
&ipaddress=65.153.12.232
&site_url=https://doc2.codetime.net
&purpose=user_account1
&redirect_url=https://doc2.codetime.net/doc/dummy.htm
&server_callback_url=https://httpstat.us/200
&merchant_data=VIP customer
&control=768eb8162fc361a3e14150ec46e9a6dd8fbfa483

```

7.18.7 Request with Card Recurring Payment ID Example

```

POST /paynet/api/v2/preauth/39549 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 314
Content-Type: application/x-www-form-urlencoded
Connection: close

card_recurring_payment_id=1491927
&cvv2=123
&client_orderid=34T43R77N
&order_desc=Test Order Description
&amount=777
&currency=USD
&ipaddress=65.153.12.232
&redirect_url=https://doc2.codetime.net/doc/dummy.htm
&server_callback_url=https://httpstat.us/200
&control=218d377897ce25c2ac69d99de42bc6902eb5bcd8

```

7.18.8 Success Response Example

```

HTTP/1.1 200 OK
Server: server
Date: Mon, 05 Sep 2022 10:43:57 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 139

type=async-response
&serial-number=00000000-0000-0000-0000-000002ddb018
&merchant-order-id=123
&paynet-order-id=6863073
&end-point-id=39914

```


7.18.9 Fail Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 05 Sep 2022 10:51:14 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 208

type=validation-error
&serial-number=00000000-0000-0000-0000-000002ddb019
&merchant-order-id=123
&error-message=Validate+card+number+failed.
↪+Card+Number+length+must+be+between+16+and+19+digits..
&error-code=8
```

7.18.10 Postman Collection

7.18.11 Request Builder

7.19 /api/v2/preauth-form

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Request Parameters for Insurance](#)
- [Response Parameters](#)
- [Request Example](#)
- [Request Example for Insurance](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.19.1 Introduction

Preauth-form is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.19.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/preauth-form/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/preauth-form/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/preauth-form/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v2/preauth-form/group/ENDPOINTGROUPID

7.19.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Acquirer can redefine the necessity of some fields so they become required instead of optional.

Leading and trailing whitespace in input parameters will be omitted.

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10

continues on next page

Table 64 – continued from previous page

Parameter Name	Description	Value
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45

continues on next page

Table 64 – continued from previous page

Parameter Name	Description	Value
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL). 2. Request parameter: client_orderid. 4. Request parameter: amount in minor units (if sent). 4. Request parameter: email. 5. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
first_name	Payer's first name.	Necessity: Required Type: String Length: 50
last_name	Payer's last name.	Necessity: Required Type: String Length: 50
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Conditional Type: String Length: 2-3

continues on next page

Table 64 - continued from previous page

Parameter Name	Description	Value
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status. Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass http://https://doc2.codetime.net if you have no need to return payer anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024
redirect_success_url	URL, where the Payer is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put http://https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 64 - continued from previous page

Parameter Name	Description	Value
redirect_fail_url	URL, where the Payer is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass http://https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
ssn	Last four digits of the Payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYYMMDD.	Necessity: Optional Type: Numeric Length: 8
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128

continues on next page

Table 64 - continued from previous page

Parameter Name	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters. This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters. This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
preferred_language	Payer's two-letter language code for multi-language payment-forms.	Necessity: Optional Type: String Length: 2

continues on next page

Table 64 - continued from previous page

Parameter Name	Description	Value
merchant_form_data	Parameters sent in MERCHANT_FORM_DATA API parameter are parsed into macros with the same name, the parameter is url-encoded, example: test-param%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters[a-zA-Z0-9], parameter value characters[a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128
minimum_transaction_amount	This parameter can be used to limit the minimum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10
maximum_transaction_amount	This parameter can be used to limit the maximum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10
customer_level	Customer level in CMS system	Necessity: Optional Type: Varchar Length: 32
customer_id	Customer ID in CMS system. Required if transaction goes via CMS (Payment Gateway mode)	Necessity: Optional Type: Int Length: 10
merchant_customer_id	Merchant Customer ID in CMS system. Required if transaction goes via CMS (CRM mode)	Necessity: Optional Type: Varchar Length: 64

continues on next page

Table 64 – continued from previous page

Parameter Name	Description	Value
preferred_language	Payer's two-letter language code for multi-language payment forms.	Necessity: Optional Type: String Length: 2
card_recurring_payment_id	Payer's tokenized cardholder's data ID, referred as Recurring Payment ID (RPI). Recurring Payment ID can be created with v4 tokenization request .	Necessity: Conditional Type: Long
cardrefid	Card reference ID used in subsequent recurring payments. Card reference ID can be created with v4 tokenization request or v2 tokenization request .	Necessity: Conditional Type: Long

7.19.4 Request Parameters for Insurance

Note:

The acquirer can override the mandatory nature of some fields to make them mandatory instead of optional.

Leading and trailing whitespace in input parameters will be trimmed.

Parameter Name	Description	Value
insurance_amount	The insurance amount. The amount must be specified in minimum units with a . delimiter. For example, 100.5 in USD means 100 US dollars and 50 cents.	Type: Numeric Length: 10
insured_person_first_name	Insured person's first name.	Type: String Length: 256
insured_person_last_name	Insured person's last name.	Type: String Length: 256

continues on next page

Table 65 – continued from previous page

Parameter Name	Description	Value
insured_person_middle_name	Insured person's middle name.	Type: String Length: 256
insured_person_birthday	Insured person's date of birth.	Type: String Length: 256
insured_person_document_series	Insured person's document series.	Type: String Length: 256
insured_person_document_number	Insured person's document number.	Type: String Length: 256
insured_person_document_issue_date	Date the insured person's document was issued.	Type: String Length: 256
insured_person_document_issuer_name	Name of the authority that issued the insured person's document.	Type: String Length: 256
insured_person_document_issuer_code	Code of the authority that issued the insured person's document.	Type: String Length: 256
insured_person_registration_address	Insured person's registration address.	Type: String Length: 256

continues on next page

Table 65 – continued from previous page

Parameter Name	Description	Value
insured_person_phone	Full international phone number of the insured person.	Type: String Length: 256
insured_person_email	Insured person's email address.	Type: String Length: 256
card_insurance_agreement_number	Agreement number.	Type: String Length: 256
card_insurance_agreement_sell_date	Sale date of the agreement.	Type: String Length: 256
card_insurance_agreement_start_date	Agreement start date.	Type: String Length: 256
card_insurance_agreement_end_date	Agreement end date.	Type: String Length: 256
card_insurance_agreement_amount	Insurance amount.	Type: String Length: 256
card_insurance_agreement_bonus	Insurance premium.	Type: String Length: 256

7.19.5 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.
redirect-url	The URL to the page where the Connecting Party should redirect the client's browser. Connecting Party should send HTTP 302 redirect, see General Payment-form Process Flow .

7.19.6 Request Example

```
POST /paynet/api/v2/preauth-form/39539 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 314
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=902B4FF5
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267&birthday=19820115
&address1=100 Main st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&amount=69
&email=john.smith@gmail.com
&currency=USD
&ipaddress=65.153.12.232
&site_url=https://doc2.codetime.net
&credit_card_number=4538977399606732
&card_printed_name=CARD HOLDER
```

(continues on next page)

(continued from previous page)

```
&expire_month=12
&expire_year=2099
&cvv2=123
&purpose=user_account1
&redirect_url=http://sandbox.doc2.com/doc/dummy.htm
&server_callback_url=https://httpstat.us/200
&merchant_data=VIP customer
&merchant_form_data=testparam%3Dtest1%26mynewparam%3Dtest2
&control=b7ba0b0ce36fda192c3772e045520c7a9cb5e442
&preferred_language=en
```

7.19.7 Request Example for Insurance

```
client_orderid=902B4FF5
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100 Main st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&amount=10.42
&insurance_amount=10
&email=john.smith@gmail.com
&currency=AED
&ipaddress=65.153.12.232
&site_url=www.google.com
&purpose=user_account1
&redirect_url=https://doc2.codetime.net/doc/dummy.htm
&server_callback_url=https://httpstat.us/200
&merchant_data=VIP customer
&merchant_form_data=testparam%3Dtest1%26mynewparam%3Dtest2
&insured_person_first_name=John
&insured_person_last_name=Doe
&insured_person_middle_name=J
&insured_person_birthday=19820115
&insured_person_document_series=4111
&insured_person_document_number=562297
&insured_person_document_issue_date=19970117
&insured_person_document_issuer_name=First document department
&insured_person_registration_address=Seattle 100 Main st
&insured_person_phone=+12063582043
&insured_person_email=john.smith@gmail.com
&card_insurance_agreement_number=210H3FIM2426726391
&card_insurance_agreement_sell_date=20210115
&card_insurance_agreement_start_date=20210116
&card_insurance_agreement_end_date=20250115
&card_insurance_agreement_amount=20000
&card_insurance_agreement_bonus=4000
```

(continues on next page)

(continued from previous page)

```
&control=c1ac1d532c96ddde35ad87be8b80a3d5aa0f2f48
```

7.19.8 Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Thu, 13 Oct 2022 09:54:53 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 280

type=async-form-response
&serial-number=00000000-0000-0000-0000-000002ddb0d3
&merchant-order-id=Test
&paynet-order-id=6863103
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
↳%2FBB587546567A31587163597A68633370432F78675258396E6F78367975715973596936522B594B4F646168553D
```

7.19.9 Fail Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Thu, 13 Oct 2022 09:58:28 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 170

type=validation-error
&serial-number=00000000-0000-0000-0000-000002ddb0d4
&error-message=Project+with+currency+RUB+does+not+apply+request+with+currency+USD
&error-code=16
```

7.19.10 Postman Collection

7.19.11 Request Builder

7.20 /api/v2/return

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.20.1 Introduction

Return transactions may only be initiated when the transaction have final successful status. See [Statuses](#). For Preauth it makes Cancel transaction, for Capture and Sale - Reversal. Return transactions are initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication.

7.20.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/return/ENDPOINTID	https://api.doc2.com/paynet/api/v2/return/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/return/group/ENDPOINTID	https://api.doc2.com/paynet/api/v2/return/group/ENDPOINTID

7.20.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Warning: The following characters must be escaped in the parameter values: & + “.

Parameter Name	Description	Value
login	Connecting Party login name.	Necessity: Mandatory Type: String Length: 20
orderid	Unique order identifier assigned by Doc2.0 system.	Necessity: Mandatory Type: String Length: 20
client_orderid	Unique Connecting Party identifier.	Necessity: Mandatory Type: String Length: 128
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Request parameter: login. 2. Request parameter: client_orderid. 3. Request parameter: orderid. 4. Request parameter: amount (in minor units, if sent). 5. Request parameter: currency (if sent). 6. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Conditional Type: String Length: 128

continues on next page

Table 68 – continued from previous page

Parameter Name	Description	Value
amount	The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents. If this parameter is omitted, initial transaction is reversed as a whole. Amount only makes sense for reversals, not cancels. Amount cannot exceed initial transaction amount. Please note that Currency MUST be specified if amount is specified!	Necessity: Mandatory Type: Numeric Length: 10
currency	Currency in which amount is specified. Provided in three-letter currency code format. Example: USD for US Dollar. Reference: Currency Codes .	Necessity: Mandatory Type: String Length: 50
comment	A brief description of reason.	Necessity: Mandatory Type: String Length: 50

7.20.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter Name	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
end-point-id	Entry point used for processing this transaction.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.20.5 Request Example

```
POST /paynet/api/v2/return/39914 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 162
Content-Type: application/x-www-form-urlencoded
Connection: close

login=TestMerchant
&client_orderid=Test
&orderid=6862958
&amount=5.00
&currency=RUB
&comment=Service not provided
&control=2dfdb99c4eff5b31c978ecb8bc4b4d094e24c1d4
```

7.20.6 Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 08 Aug 2022 07:50:08 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 141

type=async-response
&serial-number=000000000-0000-0000-0000-000002ddad4a
&merchant-order-id=Test
&paynet-order-id=6862958
&end-point-id=39914
```

7.20.7 Fail Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 08 Aug 2022 10:45:32 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
```

(continues on next page)

(continued from previous page)

Content-Length: 153

type=validation-error

&serial-number=000000000-0000-0000-0000-000002ddad5c

&error-message=Reversal+currency+does+not+match+project+currency

&error-code=16

7.20.8 Postman Collection

7.20.9 Request Builder

7.21 /api/v2/sale

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [SBP Sale Request Parameters](#)
- [Additional Parameters](#)
 - [For Connecting Party](#)
 - [For Payment Institutions](#)
- [Response Parameters](#)
- [Request with Cardholder Data Example](#)
- [Request with Card Recurring Payment ID Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.21.1 Introduction

Sale is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.21.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/sale/ENDPOINTID	https://doc2.com/paynet/api/v2/sale/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/sale/group/ENDPOINTID	https://doc2.com/paynet/api/v2/sale/group/ENDPOINTID

7.21.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.
 Acquirer can redefine the necessity of some fields so they become required instead of optional.
 Leading and trailing whitespace in input parameters will be omitted.

Warning: The following characters must be escaped in the parameter values: & + “.

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k

continues on next page

Table 71 – continued from previous page

Parameter Name	Description	Value
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50

continues on next page

Table 71 – continued from previous page

Parameter Name	Description	Value
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL) 2. Request parameter: client_orderid 3. Request parameter: amount (in minor units) 4. Request parameter: email 5. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
cvv2	Payer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card.	Necessity: Required Type: Numeric Length: 3-4
credit_card_number	Payer's credit card number. Send either combination of credit_card_number, card_printed_name, expire_month and expire_year or card_recurring_payment_id, not all.	Necessity: Conditional Type: Numeric Length: 20
card_recurring_payment_id	Payer's tokenized cardholder's data ID. Send either card_recurring_payment_id or combination of credit_card_number, card_printed_name, expire_month and expire_year, not all. To create card_recurring_payment_id see /api/v4/create-card-ref .	Necessity: Conditional Type: Long Length: 20
card_printed_name	Cardholder name, printed on the bank card.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 71 - continued from previous page

Parameter Name	Description	Value
expire_month	Bank card expiration month.	Necessity: Conditional Type: Numeric Length: 2
expire_year	Bank card expiration year.	Necessity: Conditional Type: Numeric Length: 4
first_name	Payer's first name. Parameter necessity depends on the Acquiring channel and necessity of it must be clarified with Tech Support.	Necessity: Required Type: String Length: 50
last_name	Payer's last name. Parameter necessity depends on the Acquiring channel and necessity of it must be clarified with Tech Support.	Necessity: Required Type: String Length: 50
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Conditional Type: String Length: 2-3
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status . Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command . Pass http://https://doc2.codetime.net if you have no need to return payer anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url , not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 71 - continued from previous page

Parameter Name	Description	Value
redirect_success_url	URL, where the Payer is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put http://https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
redirect_fail_url	URL, where the Payer is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass http://https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
ssn	Last four digits of the payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYYM-MDD.	Necessity: Optional Type: Numeric Length: 8
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15

continues on next page

Table 71 – continued from previous page

Parameter Name	Description	Value
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128
purpose	Destination to where the payment goes. It is useful for the Connecting Party's who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 71 – continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
merchant_data	Custom Connecting Party details, which can be attached to the transaction and passed back in the status response , Connecting Party callback parameters or server_callback_url. Additional information of transaction which may be useful in Connecting Party's external system, e.g. VIP customer, TV promo campaign lead. Information returns in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
dapi_imei	Unique device identifier.	Necessity: Optional Type: String Length: 32
mini-mum_transaction_amount	This parameter can be used to limit the minimum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10

continues on next page

Table 71 - continued from previous page

Parameter Name	Description	Value
maximum_transaction_amount	This parameter can be used to limit the maximum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10
customer_level	Customer level in CMS system.	Necessity: Optional Type: Varchar Length: 32
customer_id	Customer ID in CMS system. Required if transaction goes via CMS (Payment Gateway mode).	Necessity: Optional Type: Int Length: 10
merchant_customer_identifier	Connecting Party Customer ID in CMS system. Required if transaction goes via CMS (CRM mode).	Necessity: Optional Type: Varchar Length: 64
recurring-payment-id	Recurring Payment ID can be sent instead of cardholder data. CVV for native is not needed. Customer Data can be updated via /api/v4/update-recurring-payment/ . Recurring Payment ID creation is initiated through HTTPS POST request by using URLs and the parameters specified below. Use OAuth RSA-SHA256 for authentication.	Necessity: Conditional Type: Long

7.21.4 SBP Sale Request Parameters

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128

continues on next page

Table 72 - continued from previous page

Parameter Name	Description	Value
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15

continues on next page

Table 72 – continued from previous page

Parameter Name	Description	Value
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL) 2. Request parameter: client_orderid 3. Request parameter: amount (in minor units) 4. Request parameter: email 5. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status . Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command . Pass http://https://doc2.codetime.net if you have no need to return payer anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 72 – continued from previous page

Parameter Name	Description	Value
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128
purpose	Destination to where the payment goes. It is useful for the Connecting Party's who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 72 – continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
merchant_data	Custom Connecting Party details, which can be attached to the transaction and passed back in the status response , Connecting Party callback parameters or server_callback_url. Additional information of transaction which may be useful in Connecting Party's external system, e.g. VIP customer, TV promo campaign lead. Information returns in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
dapi_imei	Unique device identifier.	Necessity: Optional Type: String Length: 32

7.21.5 Additional Parameters

For Connecting Party

Note:

Browser data for 3DS 2.X is gathered by Connecting Party system on 3DS authentication stage. For some processing channels, however, the browser data and/or Connecting Party URL for 3DS challenge results must be provided in initial transaction request. Please

contact Support manager to clarify if these parameters should be included in request parameters.

The Connecting Party's site needs to accurately populate the browser information for each transaction. This data can be obtained by Connecting Party's servers. Ensure that the data is not altered or hard-coded, and that it is unique to each transaction.

Parameter Name	Description	Value
cus-tomer_browser_accept_headers	Exact content of the HTTP accept headers as sent to the 3DS Requestor from the Cardholder's browser.	Necessity: Required Type: String Length: 2048
cus-tomer_browser_javascript_enabled	Boolean that represents the ability of the cardholder browser to execute JavaScript.	Necessity: Required Type: Boolean Length: -
cus-tomer_browser_accept_language	Value representing the browser language as defined in IETF BCP47.	Necessity: Required Type: String Length: 8
cus-tomer_browser_user_agent	Exact content of the HTTP user-agent header.	Necessity: Required Type: String Length: 2048
customer_browser_info	If true, then the fields below must be present.	Necessity: Optional Type: Boolean Length: -
cus-tomer_browser_color_depth	Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: String Length: 2
cus-tomer_browser_java_enabled	Boolean that represents the ability of the cardholder browser to execute Java. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: Boolean Length: -
cus-tomer_browser_screen_height	Total height of the Cardholder's screen in pixels. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: Numeric Length: 6

continues on next page

Table 73 - continued from previous page

Parameter Name	Description	Value
cus-tomer_browser_screen_width	Total width of the cardholder's screen in pixels. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: Numeric Length: 6
cus-tomer_browser_time_zone	Time-zone offset in minutes between UTC and the Cardholder browser local time. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: String Length: 5
tds_areq_notification_url, alias tds_cres_notification_url	Fully qualified URL of Connecting Party system that will receive the CRes message or Error Message. This CRes message must be sent to Connecting Party. See details here Upload CRes Result .	Necessity: Optional Type: String Length: 256

For Payment Institutions

The PSP or Acquirer can fill the 3DS results for each transaction, if 3DS authentication is performed on their side.

Parameter Name	Description	Value
tds_authentication_result_type	Type of result. Possible values are: - SIMPLE	Type: String Length: 6
tds_authentication_result_authentication_type	Authentication Type. Indicates the type of authentication method the Issuer will use to challenge the Cardholder, whether in the ARes message or what was used by the ACS when in the RReq message. Possible values are: - 01 = Static - 02 = Dynamic - 03 = OOB - 04 = Decoupled - 05-79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) - 80-99 = Reserved for DS use	Type: String Length: 2

continues on next page

Table 74 - continued from previous page

Parameter Name	Description	Value
tds_authentication_result_authentication_value	Authentication Value. Payment System-specific value provided by the ACS or the DS using an algorithm defined by Payment System. Authentication Value may be used to provide proof of authentication. A 20-byte value that has been Base64 encoded, giving a 28-byte result	Type: String Length: 19-28
tds_authentication_result_transaction_id	transaction id.0.2 or dsTransID for 2.1.0/2.2.0	Type: String Length: 19-36
tds_authentication_result_transaction_status	Transaction Status. Indicates whether a transaction qualifies as an authenticated transaction or account verification. Possible values are: - Y = Authentication Verification Successful - N = Not Authenticated/Account Not Verified, Transaction denied - U = Authentication/Account Verification Could Not Be Performed, Technical or other problem, as indicated in ARes or RReq - A = Attempts Processing Performed, Not Authenticated/Verified, but a proof of attempted authentication/verification is provided - C = Challenge Required, Additional authentication is required using the CReq/CRes - D = Challenge Required, Decoupled Authentication confirmed - R = Authentication/ Account Verification Rejected, Issuer is rejecting	Type: String Length: 1

continues on next page

Table 74 - continued from previous page

Parameter Name	Description	Value
tds_authentication_result_message_version	Message Version Number. Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction. Possible values are: - 1.0.2 - 2.1.0 - 2.2.0	Type: String Length: 5

7.21.6 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter Name	Description
type	The type of response. May be async-response, validation-error, error etc.. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is declined or error this parameter contains the reason for decline or error details.
error-code	The error code in case of declined or error status.
end-point-id	Endpoint id used for the transaction.

7.21.7 Request with Cardholder Data Example

```
POST /paynet/api/v2/sale/39529 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 314
Content-Type: application/x-www-form-urlencoded
Connection: close

credit_card_number=4538977399606732
&card_printed_name=John
&expire_month=01
&expire_year=2042
&cvv2=123
&client_orderid=34T43R77N
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100%20Main%20st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&amount=156
&email=john.smith@gmail.com
&currency=RUB
&ipaddress=65.153.12.232
&site_url=https://doc2.codetime.net
&purpose=user_account1
&redirect_url=http%3A%2F%2Fhttps://doc2.codetime.net%2Fdoc%2Fdummy.htm
&server_callback_url=https%3A%2F%2Fhttpstat.us%2F200
&merchant_data=VIP customer
&dapi_imei=123
&control=c821e33bd22773c05c23725d0b1d2dbd9f191399
```

Request for SBP Example

```
POST /paynet/api/v2/sale/39529 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 314
Content-Type: application/x-www-form-urlencoded
Connection: close

&client_orderid=34T43R77N
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
```

(continues on next page)

(continued from previous page)

```
&address1=100%20Main%20st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&amount=156
&email=john.smith@gmail.com
&currency=RUB
&ipaddress=65.153.12.232
&site_url=https://doc2.codetime.net
&purpose=user_account1
&redirect_url=http%3A%2F%2Fhttps://doc2.codetime.net%2Fdoc%2Fdummy.htm
&server_callback_url=https%3A%2F%2Fhttpstat.us%2F200
&merchant_data=VIP customer
&dapi_imei=123
&control=c821e33bd22773c05c23725d0b1d2dbd9f191399
```

7.21.8 Request with Card Recurring Payment ID Example

```
POST /paynet/api/v2/sale/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 317
Content-Type: application/x-www-form-urlencoded
Connection: close

card_recurring_payment_id=1491954
&cvv2=123
&client_orderid=34T43R77N
&order_desc=Test Order Description
&amount=777
&currency=USD
&ipaddress=65.153.12.232
&redirect_url=http://doc2.doc2.com/doc/dummy.htm
&server_callback_url=https://httpstat.us/200
&control=80761544c64373d1624240add048d36d42fe528a
```

7.21.9 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 11:03:17 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
```

(continues on next page)

(continued from previous page)

```
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=async-response
&serial-number=00000000-0000-0000-0000-000002d9b22a
&merchant-order-id=inv4097763
&paynet-order-id=6768788
&end-point-id=22903
```

7.21.10 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 13:14:40 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=validation-error
&serial-number=00000000-0000-0000-0000-000002b36f64
&merchant-order-id=inv4097763
&error-message=End+point+with+id+22903+not+found
&error-code=3
```

7.21.11 Postman Collection

7.21.12 Request Builder

7.22 /api/v2/sale-form

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Request Parameters for Insurance](#)
- [Response Parameters](#)
- [Request Example](#)
- [Request Example for Insurance](#)

- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.22.1 Introduction

Sale-form is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.22.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/sale-form/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/sale-form/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/sale-form/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v2/sale-form/group/ENDPOINTGROUPID

7.22.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Acquirer can redefine the necessity of some fields so they become required instead of optional.

Leading and trailing whitespace in input parameters will be omitted.

Warning: The following characters must be escaped in the parameter values: & + “.

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128

continues on next page

Table 77 - continued from previous page

Parameter Name	Description	Value
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
first_name	Payer's first name. Parameter necessity depends on the Acquiring channel and necessity of it must be clarified with Tech Support.	Necessity: Required Type: String Length: 50
last_name	Payer's last name. Parameter necessity depends on the Acquiring channel and necessity of it must be clarified with Tech Support.	Necessity: Required Type: String Length: 50
ssn	Last four digits of the payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYYMMDD.	Necessity: Optional Type: Numeric Length: 8
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Conditional Type: String Length: 2-3
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10

continues on next page

Table 77 - continued from previous page

Parameter Name	Description	Value
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50
purpose	Destination to where the payment goes. It is useful for the Connecting Party's who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 20
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128

continues on next page

Table 77 - continued from previous page

Parameter Name	Description	Value
control	Checksum generated by SHA-1. Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL) 2. Request parameter: client_orderid 3. Request parameter: amount in minor units (if sent). 4. Request parameter: email 5. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status. Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass http://https://doc2.codetime.net if you have no need to return payer anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 77 - continued from previous page

Parameter Name	Description	Value
redirect_success_url	URL, where the Payer is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put http://https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
redirect_fail_url	URL, where the Payer is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass http://https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 77 - continued from previous page

Parameter Name	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
preferred_language	Payer's two-letter language code for multi-language payment-forms.	Necessity: Optional Type: String Length: 2

continues on next page

Table 77 - continued from previous page

Parameter Name	Description	Value
merchant_data	Custom Connecting Party details, which can be attached to the transaction and passed back in the status response , Connecting Party callback parameters or server_callback_url. Additional information of transaction which may be useful in Connecting Party's external system, e.g. VIP customer, TV promo campaign lead. Information returns in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
merchant_form_data	Parameters sent in MERCHANT_FORM_DATA API parameter are parsed into macros with the same name, the parameter is url-encoded, example: test-param%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters[a-zA-Z0-9], parameter value characters[a-zA-Z0-9], control characters [=&], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128
minimum_transaction_amount	This parameter can be used to limit the minimum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10
maximum_transaction_amount	This parameter can be used to limit the maximum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10
customer_level	Customer level in CMS system	Necessity: Optional Type: Varchar Length: 32

continues on next page

Table 77 - continued from previous page

Parameter Name	Description	Value
customer_id	Customer ID in CMS system. Required if transaction goes via CMS (Payment Gateway mode)	Necessity: Optional Type: Int Length: 10
merchant_customer_id	Merchant Customer ID in CMS system. Required if transaction goes via CMS (CRM mode)	Necessity: Optional Type: Varchar Length: 64
card_recurring_payment_id	Payer's tokenized cardholder's data ID, referred as Recurring Payment ID (RPI). Recurring Payment ID can be created with v4 tokenization request .	Necessity: Conditional Type: Long
cardrefid	Card reference ID used in subsequent recurring payments. Card reference ID can be created with v4 tokenization request or v2 tokenization request .	Necessity: Conditional Type: Long

For Faster Payments System (CBII) integration use purpose field for fpsRequestId parameter.

7.22.4 Request Parameters for Insurance

Note:

The acquirer can override the mandatory nature of some fields to make them mandatory instead of optional.

Leading and trailing whitespace in input parameters will be trimmed.

Parameter Name	Description	Value
insurance_amount	The insurance amount. The amount must be specified in minimum units with a . delimiter. For example, 100.5 in USD means 100 US dollars and 50 cents.	Type: Numeric Length: 10
insured_person_first_name	Insured person's first name.	Type: String Length: 256

continues on next page

Table 78 – continued from previous page

Parameter Name	Description	Value
insured_person_last_name	Insured person's last name.	Type: String Length: 256
insured_person_middle_name	Insured person's middle name.	Type: String Length: 256
insured_person_birthday	Insured person's date of birth.	Type: String Length: 256
insured_person_document_series	Insured person's document series.	Type: String Length: 256
insured_person_document_number	Insured person's document number.	Type: String Length: 256
insured_person_document_issue_date	Date the insured person's document was issued.	Type: String Length: 256
insured_person_document_issuer_name	Name of the authority that issued the insured person's document.	Type: String Length: 256
insured_person_document_issuer_code	Code of the authority that issued the insured person's document.	Type: String Length: 256

continues on next page

Table 78 – continued from previous page

Parameter Name	Description	Value
insured_person_registration_address	Insured person's registration address.	Type: String Length: 256
insured_person_phone	Full international phone number of the insured person.	Type: String Length: 256
insured_person_email	Insured person's email address.	Type: String Length: 256
card_insurance_agreement_number	Agreement number.	Type: String Length: 256
card_insurance_agreement_sell_date	Sale date of the agreement.	Type: String Length: 256
card_insurance_agreement_start_date	Agreement start date.	Type: String Length: 256
card_insurance_agreement_end_date	Agreement end date.	Type: String Length: 256
card_insurance_agreement_amount	Insurance amount.	Type: String Length: 256

continues on next page

Table 78 – continued from previous page

Parameter Name	Description	Value
card_insurance_agreement_bonus	Insurance premium.	Type: String Length: 256

7.22.5 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter name	Description
type	The type of response. May be async-form-response, validation-error, error etc.. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is declined or error this parameter contains the reason for decline or error details.
error-code	The error code in case of declined or error status.
redirect-url	The URL to the page where the Connecting Party should redirect the client's browser. Connecting Party should send HTTP 302 redirect, see General Payment-form Process Flow .

7.22.6 Request Example

```
POST /paynet/api/v2/sale-form/39519 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 314
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=inv1409911
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
```

(continues on next page)

(continued from previous page)

```
&address1=100 Main st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&amount=156
&email=john.smith@gmail.com
&currency=USD
&ipaddress=65.153.12.232
&site_url=https://doc2.codetime.net
&purpose=user_account1
&redirect_url=http://connectingparty.com/result
&server_callback_url=https://httpstat.us/200
&merchant_data=VIP customer
&merchant_form_data=testparam%3Dtest1%26mynewparam%3Dtest2
&control=185aea68b751221b78fa9138e9d44a6aa4c2c446
```

7.22.7 Request Example for Insurance

```
client_orderid=902B4FF5
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100 Main st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&amount=10.42
&insurance_amount=10
&email=john.smith@gmail.com
&currency=AED
&ipaddress=65.153.12.232
&site_url=www.google.com
&purpose=user_account1
&redirect_url=https://doc2.codetime.net/doc/dummy.htm
&server_callback_url=https://httpstat.us/200
&merchant_data=VIP customer
&merchant_form_data=testparam%3Dtest1%26mynewparam%3Dtest2
&insured_person_first_name=John
&insured_person_last_name=Doe
&insured_person_middle_name=J
&insured_person_birthday=19820115
&insured_person_document_series=4111
&insured_person_document_number=562297
&insured_person_document_issue_date=19970117
&insured_person_document_issuer_name=First document department
&insured_person_registration_address=Seattle 100 Main st
```

(continues on next page)

(continued from previous page)

```
&insured_person_phone=+12063582043
&insured_person_email=john.smith@gmail.com
&card_insurance_agreement_number=210H3FIM2426726391
&card_insurance_agreement_sell_date=20210115
&card_insurance_agreement_start_date=20210116
&card_insurance_agreement_end_date=20250115
&card_insurance_agreement_amount=20000
&card_insurance_agreement_bonus=4000
&control=c1ac1d532c96ddde35ad87be8b80a3d5aa0f2f48
```

7.22.8 Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Tue, 11 Oct 2022 14:25:25 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 280

type=async-form-response
&serial-number=00000000-0000-0000-0000-000002ddb0b9
&merchant-order-id=Test
&paynet-order-id=6863099
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
%2FBB587546567A31587163597A68633370432F786752582F6154674965594A696F4D41306C50596E334F5453553D
```

7.22.9 Fail Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Tue, 11 Oct 2022 14:16:06 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 170

type=validation-error
&serial-number=00000000-0000-0000-0000-000002ddb0b7
&error-message=Project+with+currency+RUB+does+not+apply+request+with+currency+USD
&error-code=16
```

7.22.10 Postman Collection

7.22.11 Request Builder

7.23 /api/v2/payout-form

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Test Scenario](#)
- [Postman Collection](#)
- [Request Builder](#)

7.23.1 Introduction

Payout-form is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#).

7.23.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/payout-form/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/payout-form/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/payout-form/group/ENDPOINTIDGROUPID	https://gate.doc2.com/paynet/api/v2/payout-form/group/ENDPOINTIDGROUPID

7.23.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Ask support manager if conditional fields are required for integration.

Parameter Name	Description	Value
client_orderid	Connecting Party's order identifier.	Necessity: Required Type: String Length: 128
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Brief order description.	Necessity: Required Type: String Length: 65K
ipaddress	Receiver's IP address (IPv4 or IPv6).	Necessity: Conditional Type: String Length: 7-45
purpose	Payout purpose.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 81 – continued from previous page

Parameter Name	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 81 – continued from previous page

Parameter Name	Description	Value
redirect_url	URL, where the Receiver is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status. Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if you have no need to return Receiver anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Required if both redirect_success_url and redirect_fail_url are missing Type: String Length: 1024
redirect_success_url	URL, where the Receiver is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Receiver anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Required if redirect_url parameter is missing Type: String Length: 1024

continues on next page

Table 81 – continued from previous page

Parameter Name	Description	Value
redirect_fail_url	URL, where the Receiver is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if you use non-3DS schema for transactions processing and you have no need to return Receiver anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Required if redirect_url parameter is missing Type: String Length: 1024
account_number	Account number.	Necessity: Conditional Type: String Length: 32
account_name	Bank account.	Necessity: Conditional Type: String Length: 512
ewallet_wallet	E-wallet ID.	Necessity: Conditional Type: String Length: 128
crypto_wallet_address	Address of crypto wallet.	Necessity: Conditional Type: String Length: 64
bank_name	Bank Name.	Necessity: Conditional Type: String Length: 512

continues on next page

Table 81 – continued from previous page

Parameter Name	Description	Value
bank_branch	Bank Branch Name.	Necessity: Conditional Type: String Length: 512
bank_code	Bank code.	Necessity: Conditional Type: String Length: 32
bank_address1	Bank address.	Necessity: Conditional Type: String Length: 255
bank_zip_code	Bank postal ZIP code.	Necessity: Conditional Type: String Length: 32
bank_province	Bank province.	Necessity: Conditional Type: String Length: 128
bank_area	Bank area.	Necessity: Conditional Type: String Length: 128
bank_city	Bank city.	Necessity: Conditional Type: String Length: 128
routing_number	Routing number used to identify specific bank branches in China.	Necessity: Conditional Type: String Length: 32

continues on next page

Table 81 – continued from previous page

Parameter Name	Description	Value
legal_person_name	Name on the legal document.	Necessity: Conditional Type: String Length: 128
legal_person_document_number	Number of legal document.	Necessity: Conditional Type: String Length: 128
receiver_first_name	Receiver first name, also can be sent as first_name.	Necessity: Conditional Type: String Length: 128
receiver_last_name	Receiver last name, also can be sent as last_name.	Necessity: Conditional Type: String Length: 128
receiver_birthday	Receiver birthday, also can be sent as birthday.	Necessity: Conditional Type: Numeric Length: 30
receiver_country_code	Receiver country code, also can be sent as country.	Necessity: Conditional Type: String Length: 3
receiver_state	Receiver state, should be provided for countries that have states (USA, Canada, Australia), also can be sent as state.	Necessity: Conditional Type: String Length: 4
receiver_city	Receiver city, also can be sent as city.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 81 – continued from previous page

Parameter Name	Description	Value
receiver_zip_code	Receiver zip code, also can be sent as zip_code.	Necessity: Conditional Type: Numeric Length: 32
receiver_address1	Receiver address, also can be sent as address1.	Necessity: Conditional Type: String Length: 255
receiver_phone	Receiver phone, also can be sent as phone.	Necessity: Conditional Type: Numeric Length: 128
receiver_email	Receiver E-mail, also can be sent as email.	Necessity: Conditional Type: String Length: 128
receiver_identity_document_id	Receiver identity document identifier, also can be sent as identity_document_id.	Necessity: Conditional Type: String Length: 128
receiver_identity_document_number	Receiver identity document number, also can be sent as identity_document_number.	Necessity: Conditional Type: String Length: 128
merchant_data	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party's Callback.	Necessity: Optional Type: String Length: 65k

continues on next page

Table 81 – continued from previous page

Parameter Name	Description	Value
merchant_form_data	Parameters sent in merchant_form_data API parameter are parsed into macros with the same name, the parameter is url-encoded, example: test-param%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters[a-zA-Z0-9], parameter value characters[a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 2M
preferred_language	Preferred language.	Necessity: Optional Type: String Length: 2

7.23.4 Response Parameters

Note:

Response has a Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter Name	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.
redirect_url	The URL to the page where the Connecting Party should redirect the client's browser. Connecting Party should send HTTP 302 redirect.

7.23.5 Request Example

```
POST /paynet/api/v2/payout-form/39529 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/8.12.1
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="merchantlogin",
  ↳oauth_timestamp="1753337681",oauth_nonce="T5v7kcMBsgi",oauth_signature_method="HMAC-
  ↳SHA1",oauth_signature="wTTQqIN%2F2bGj fCTcSAQ3ZhAHMLw%3D"
Content-Length: 249
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

account_number=1234567890
&order_desc=Test_Order_Description
&amount=100
&bank_branch=test_branch
&bank_name=test_bank
&client_orderid=12345
&currency=USD
&oauth_consumer_key=merchantlogin
&oauth_nonce=T5v7kcMBsgi
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1753337681
&oauth_version=1.0
```

7.23.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 24 Jul 2025 06:45:56 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 142

type=async-response
&serial-number=00000000-0000-0000-0000-000002f3b45d
&merchant-order-id=12345
&paynet-order-id=7366391
&end-point-id=132490
```

7.23.7 Fail Response Example

```
HTTP/1.1 403
Server: server
Date: Thu, 24 Jul 2025 06:26:23 GMT
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Length: 102

type=error
&serial-number=000000000-0000-0000-0000-000002f3b456
&error-message=Forbidden
&error-code=-1
```

7.23.8 Test Scenario

Different Payout transaction statuses can be received on sandbox, depending on the account_number value passed in Payout request.

Testing account_number values:

- account_number = 1234567890 to get APPROVED
- account_number = 0987654321 to get DECLINED
- account_number = 1987654321 to get PROCESSOR_INTERNAL_ERROR

7.23.9 Postman Collection

7.23.10 Request Builder

7.24 /api/v2/sale-echeck

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.24.1 Introduction

Sale-eCheck is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.24.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/sale-echeck/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/sale-echeck/ENDPOINTID

7.24.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.
Leading and trailing whitespace in input parameters will be omitted.

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
first_name	Payer's first name. Parameter necessity depends on the Acquiring channel and necessity of it must be clarified with Tech Support.	Necessity: Required Type: String Length: 50
last_name	Payer's last name. Parameter necessity depends on the Acquiring channel and necessity of it must be clarified with Tech Support.	Necessity: Required Type: String Length: 50

continues on next page

Table 84 - continued from previous page

Parameter Name	Description	Value
ssn	Last four digits of the Payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYYMMDD.	Necessity: Optional Type: Numeric Length: 8
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Conditional Type: String Length: 2-3
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50

continues on next page

Table 84 - continued from previous page

Parameter Name	Description	Value
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
routing_number	This element should contain the Payer's 9 digit bank routing number.	Necessity: Required Type: String Length: 9
account_number	This element should contain the Payer's bank account number.	Necessity: Required Type: String Length: 20
check_number	This element should contain the Payer's check number.	Necessity: Optional Type: String Length: 22
check_date	This element should contain the Payer's check date. Format is MM/DD/YYYY.	Necessity: Optional Type: String Length: 12
bank_name	This element should contain the Payer's bank name.	Necessity: Optional Type: String Length: 12
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45

continues on next page

Table 84 – continued from previous page

Parameter Name	Description	Value
control	Checksum generated by SHA-1. Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID> (See: Request URL) 2. Request parameter: client_orderid 3. Request parameter: amount (in minor units) 4. Request parameter: email 5. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters. This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 84 – continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

7.24.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. May be async-response, validation-error, error etc. If type equals validation-error or error, error-message and error-code parameters contain error details.
status	See Status List for details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.24.5 Request Example

```
POST /paynet/api/v2/sale-echeck/39790 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 314
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=902B4FF5
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100 Main st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&amount=10.55
&email=john.smith@gmail.com
&currency=USD
&ipaddress=65.153.12.232
&routing_number=113024588
&account_number=1234567890
&check_number=77778888
&check_date=02/22/2025
&bank_name=testbank
&server_callback_url=https%3A%2F%2Fhttpstat.us%2F200
&control=5ce2641b344bf35930ba9536055d24f038a1065b
```

7.24.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 26 Apr 2023 13:51:16 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 144

type=async-response
&serial-number=000000000-0000-0000-0000-000002e3450d
&merchant-order-id=902B4FF5
&paynet-order-id=6994012
```

(continues on next page)

(continued from previous page)

`&end-point-id=39790`

7.24.7 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 26 Apr 2023 13:51:16 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 144

type=validation-error
&serial-number=00000000-0000-0000-0000-000002b36f64
&merchant-order-id=inv4097763
&error-message=End+point+with+id+39790+not+found
&error-code=3
```

7.24.8 Postman Collection

7.24.9 Request Builder

7.25 /api/v2/status

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [PaReqForm Status Response Parameters](#)
- [CReqForm Status Response Parameters](#)
- [MethodUrlFrame Status Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.25.1 Introduction

To make an order status request one have to send an HTTPS POST request to the [URLs](#) and the [parameters](#) specified below. Use [SHA-1](#) for authentication. See [Statuses](#).

7.25.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/status/ENDPOINTID	https://paynet.doc2.com/paynet/api/v2/status/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/status/group/ENDPOINTID	https://paynet.doc2.com/paynet/api/v2/status/group/ENDPOINTID

7.25.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Parameter Name	Description	Necessity
login	Merchant login name.	Mandatory
client_orderid	Unique order identifier assigned by merchant.	Mandatory
orderid	Order id assigned to the order by Doc2.0.	Conditional
by-request-sn	Serial number assigned to the specific request by doc2. If this field exist in status request, status response return for this specific request. Include this parameter to get the status request with the particular transaction stage (can be used in specific cases). To get the latest transaction status, don't include this parameter in status request.	Optional

continues on next page

Table 87 - continued from previous page

Parameter Name	Description	Necessity
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Request parameter: login. 2. Request parameter: client_orderid. 3. Request parameter: orderid. 4. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Mandatory

In most common cases, the best option is to include both `client_orderid` and `orderid` parameters to status request. Order status can be requested with only `client_orderid` if it's unique to merchant and `orderid` is not received. If `orderid` is not received in response, but this response contains an error, see the received error message to get the information why transaction was not created in the system.

7.25.4 Response Parameters

The same API command for status request is used in multiple Use-Cases, therefore some of the mentioned response parameters might not be present for specific case. Below is the full list of possible parameters.

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

* - these parameters are not defined by default. Please contact tech support to include these fields in callback.

Status Response Parameter	Description
type	The type of response. May be status-response.
status	See Status List for details.
amount	Actual transaction amount. This value can be changed during the transaction flow.
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar EUR for European Euro.
paynet-order-id	Order id assigned to the order by gate.doc2.com.
merchant-order-id	Connecting Party order id.

continues on next page

Table 88 – continued from previous page

Status Response Parameter	Description
phone	Payer's full international phone number, including country code.
html	HTML code of 3DS authorization form, encoded in application/x-www-form-urlencoded MIME format. Merchant must decode this parameter before showing the form to the Payer. gate.doc2.com System returns the following response parameters when it gets 3DS authorization form from the Issuer Bank. It contains auth form HTML code which must be passed through without any changes to the client's browser. This parameter exists and has value only when the redirection HTML is already available. For non-3DS this never happens. For 3DS HTML has value after some short time after the processing has been started.
redirect-to	For 3DS authorization the merchant can redirect the Payer to URL provided in this parameter instead of rendering the page provided in html parameter. The redirect-to parameter is returned only if the html parameter is returned. Merchant should use GET HTTP method to redirect. This parameter must be used to work with 3DS 2.0.
serial-number	Unique number assigned by gate.doc2.com server to particular request from the Connecting Party.
last-four-digits	Last four digits of Payer bank card number.
dest-last-four-digits	Last four digits of customer credit card number. Relevant only for transfer transactions.
bin	Bank BIN of Payer bank card number.
card-type	Type of Payer bank card (VISA, MASTERCARD, etc).
gate-partial-reversal	Processing gate support partial reversal (enabled or disabled).
gate-partial-capture	Processing gate support partial capture (enabled or disabled).
transaction-type	Transaction type (sale, reversal, capture, preauth).
processor-rrn	Bank Receiver Registration Number.
processor-tx-id	Acquirer transaction identifier.
receipt-id	Electronic link to receipt https://gate.doc2.com/paynet/view-receipt/ENDPOINTID/receipt-id/ .
name	Payer's name.
card-ref-id	Card reference ID used in subsequent recurrent payments. Relevant only if card-ref-id was created for initial transaction.
cardholder-name	Cardholder's name.
card-exp-month	Bank card expiration month.
card-exp-year	Bank card expiration year.
card-hash-id	Unique card identifier to use for loyalty programs or fraud checks.
card-country-alpha-three-code	Three letter country code of source card issuer. See Country and State Codes for details.
destination-card-country-alpha-three-code	Three letter country code of destination card issuer. See Country and State Codes for details.
dest-bin	Bank BIN of customer credit card number.
dest-card-type	Type of customer credit card (VISA, MASTERCARD, etc).
dest-bank-name	Bank name by customer card BIN.

continues on next page

Table 88 – continued from previous page

Status Response Parameter	Description
destination-hash-id	Unique card identifier to use for loyalty programs or fraud checks. Relevant only for transfer transactions.
destination-card-hash-id	Unique card identifier to use for loyalty programs or fraud checks.
first-name	Payer's first name.
last-name	Payer's last name.
email	Payer's e-mail.
country *	Payer's country (two-letter country code). Please see Country and State Codes for a list of valid country codes.
state *	Payer's state . Please see Country and State Codes for a list of valid state codes. Mandatory for USA, Canada and Australia.
city *	Payer's city.
zip_code *	Payer's ZIP code.
address1 *	Payer's address line 1.
purpose	Destination to where the payment goes. It is useful for the merchants who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.
bank-name	Bank name by Payer card BIN.
terminal-id	Acquirer terminal identifier to show in receipt.
paynet-processing-date	Acquirer transaction processing date.
approval-code	Bank approval code.
order-stage	The current stage of the transaction processing. See Order Stage for details.
total-reversal-amount	Total amount of processed reversals. Relevant only for reversal transactions.
reversal-amount	The amount of the last processed reversal. Relevant only for reversal transactions.
auth-response-code	Response code used in Iso8583 protocol. Only returned in specific cases.
acquirer-processing-date	Acquirer transaction processing date.
processor-auth-credit-code	Approval credit code. Only returned in specific cases.
processor-credit-rrn	Retrieval Reference Number for credit transaction.
processor-credit-arn	Acquirer card reference number for credit transaction.
processor-debit-arn	Acquirer card reference number for debit transaction.
loyalty-balance	The current bonuses balance of the loyalty program for current operation. if available.
loyalty-message	The message from the loyalty program. if available.
loyalty-bonus	The bonus value of the loyalty program for current operation if available.
loyalty-program	The name of the loyalty program for current operation if available.
descriptor	Bank identifier of the payment recipient.
original-gate-descriptor	Descriptor, which is set on gate level in the system.
error-message	If status is declined, error, filtered this parameter contains the reason for decline.
error-code	The error code is case status is declined, error, filtered.

continues on next page

Table 88 – continued from previous page

Status Response Parameter	Description
by-request-sn	Serial number assigned to the specific request by gate.doc2.com. If this field exist in status request, status response return for this specific request.
verified-3d-status	See 3D Secure Status List for details.
verified-rsc-status	Returned if Random Sum Check was performed. See Alternative cardholder authentication
eci	Electronic Commerce Indicator (Visa).
ips-src-payment-product-code	Code for card set by multinational financial service (Visa/Mastercard).
ips-src-payment-product-name	Decrypted code for card set by multinational financial service (Visa/Mastercard).
ips-src-payment-type-code	Type of card code set by multinational financial service (Visa/Mastercard).
ips-src-payment-type-name	Decrypted code for type of card set by multinational financial service (Visa/Mastercard).
merchantdata	If provided in initial request, merchant_data parameter and its value will be included in status response.
initial-amount	Amount, set in initiating transaction, without any fees or commissions. This value can't change during the transaction flow.
seller-commission	Total commission for processed transaction. This is optional parameter. Please contact your manager in Doc2.0, if you would like to receive it.
acquirer-commission	Acquirer commission for processed transaction. This is optional parameter. Please contact your manager in Doc2.0, if you would like to receive it.
motivational-message	This is an optional message which contains extended information about the reason for the declined transaction.
transaction-date	Date of final status assignment for transaction.
orig-amount	Contains the original request amount if it was converted on auxiliary endpoint in Parallel form integration. Relevant only for Payment Cashier transactions.
orig-currency	Contains the original request currency if it was converted on auxiliary endpoint in Parallel form integration. Relevant only for Payment Cashier transactions.

QR Code Status Response Parameters:

Status Response Parameter	Description
qr-code	QR code in base 64 format.
qr-code-payload-type	QR Code type = SBP.
qr-code-payload-value	Link to the QR code = https://qr.nspk.ru/BS***** (only for H2H integration).

7.25.5 PaReqForm Status Response Parameters

Name	Description
tds-pareq-form-pareq	ACS 3DS PaReq data, which received by the Connecting Party.
tds-pareq-form-acurl	ACS URL to redirect the Payer to 3DS 1.0.2 Authentication Flow.

7.25.6 CReqForm Status Response Parameters

Name	Description
tds-creq-form-creq	A CReq message initiates Cardholder interaction in a Challenge Flow and is used to carry authentication data from the Cardholder. It is formed by the 3DS Server and is posted through the Cardholder's browser by the merchant to the ACS URL.
tds-creq-form-acurl	ACS URL to redirect the Payer for Challenge Flow.

7.25.7 MethodUrlFrame Status Response Parameters

Name	Description
tds-method-url-frame-3ds-server-trans-id	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.
tds-method-url-frame-3ds-method-url	3DS Method URL used in iframe form which is provided to Payer browser by the merchant.

Rules to form the HTML form.

threeDSMethodData (threeDSMethodNotificationURL + threeDSServerTransID).

7.25.8 Request Example

```
POST /paynet/api/v2/status/37211 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.77.0
Accept: */*
Content-Length: 99
Content-Type: application/x-www-form-urlencoded
Connection: close

login=TestYujik
&client_orderid=123
&orderid=6863082
&control=647f0581bbceb804a73e98d9ea7e78640a75bf1c
```

7.25.9 Success Response Example

```

type=status-response
&serial-number=00000000-0000-0000-0000-0000037704d3
&merchant-order-id=902B4FF5
&paynet-order-id=6863082
&status=processing
&amount=10.42
&currency=RUB
&original-gate-descriptor=PAYMENT-RUB
&transaction-type=sale
&receipt-id=7d59a029-2316-36e5-b29b-96139fc7af38
&card-exp-month=0
&card-exp-year=0
&email=john.smith@gmail.com
&order-stage=sale_3d_validating
&merchantdata=VIP customer
&card-type=SBP
&phone=12063582043
&paynet-processing-date=2023-03-29 13:47:40 MSK
&first-name=John
&last-name=Smith
&initial-amount=10.42
&qrcode=data:image/png;base64,
  ↪iVBORw0KGgoAAAANSUgAAQAAAAEACAIAAADTED8xAAAFIk1EQVR42u3dSXLjQAwEQP7/0/IbHFI...
&qrcode-payload-type=SBP
&qrcode-payload-value=https://qr.nspk.ru/BS100069IM4ESCN090DP55N859KG1AAD

```

7.25.10 Fail Response Example

```

HTTP/1.1 200 OK
Server: server
Date: Mon, 12 Sep 2022 09:08:02 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
X-Cached: MISS
Content-Length: 137

type=validation-error
&serial-number=00000000-0000-0000-0000-000002ddb057
&error-message=End+point+with+id+372118+not+found
&error-code=3

```

7.25.11 Postman Collection

7.25.12 Request Builder

7.26 /api/v2/void

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.26.1 Introduction

Void can only be processed for Sale and Capture transactions in the final successful status (See [Statuses](#)). To make a void request send an HTTPS POST request to the [URLs](#) and the [parameters](#). Use [SHA-1](#) for authentication.

7.26.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/void/ENDPOINTID	https://api.doc2.com/paynet/api/v2/void/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v2/void/group/ENDPOINTID	https://api.doc2.com/paynet/api/v2/void/group/ENDPOINTID

7.26.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded.

Warning: The following characters must be escaped in the parameter values: & + “.

Parameter Name	Description	Value
login	Connecting Party login name.	Necessity: Mandatory Type: String Length: 20
orderid	Unique order identifier assigned by Doc2.0 system.	Necessity: Mandatory Type: String Length: 20
client_orderid	Unique Connecting Party identifier.	Necessity: Mandatory Type: Numeric Length: 10
amount	This optional parameter specifies amount to be reversed.	Necessity: Mandatory Type: String Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Conditional Type: String Length: 3

continues on next page

Table 94 – continued from previous page

Parameter Name	Description	Value
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Request parameter: login 2. Request parameter: client_orderid 3. Request parameter: orderid 4. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Conditional Type: String Length: 128
comment	A brief description of reason.	Necessity: Mandatory Type: String Length: 50

7.26.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter Name	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
end-point-id	Entry point used for processing this transaction.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.26.5 Request Example

```
POST /paynet/api/v2/void/39914 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 123
Content-Type: application/x-www-form-urlencoded
Connection: close

login=Test
&client_orderid=55
&orderid=6862995
&comment=Service not provided
&control=246e0a2074f9e4b84cff59b2567ed48ec3aab87d
```

7.26.6 Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Tue, 16 Aug 2022 06:57:40 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 138

type=async-response
&serial-number=000000000-0000-0000-0000-000002ddae76
&merchant-order-id=55
&paynet-order-id=6862995
&end-point-id=39914
```

7.26.7 Fail Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Tue, 16 Aug 2022 07:21:37 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 148
```

(continues on next page)

(continued from previous page)

```
type=validation-error
&serial-number=00000000-0000-0000-0000-000002ddae78
&merchant-order-id=Test
&error-message=INVALID_CONTROL_CODE
&error-code=2
```

7.26.8 Postman Collection

7.26.9 Request Builder

7.27 /api/v2/card-insurance-document

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.27.1 Introduction

Card insurance document is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#).

Note:

After a successful transaction, it is necessary to generate a request to download the insurance policy with the Payer's data in PDF format.

7.27.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v2/card-insurance-document/ENDPOINTID	https://gate.doc2.com/paynet/api/v2/card-insurance-document/ENDPOINTID

7.27.3 Request Parameters

Parameter Name	Description
client-order-id	Unique order identifier assigned by Connecting Party.
order_id	Order id assigned to the order by Doc2.0.

7.27.4 Response Parameters

Parameter Name	Description
request_serial_number	A unique number assigned by the server Doc2.0 for a specific request from the Connecting Party.
document_path	The link to download the document.

7.27.5 Request Example

```
POST /paynet/api/v2/card-insurance-document/3213 HTTP/1.1
User-Agent: curl/8.7.1
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="merchantlogin",
↪oauth_signature_method="HMAC-SHA1",oauth_signature="TibClG%2FGfoxiGvKSyRovTH0P1Pg%3D
↪"
Content-Length: 120
Content-Type: application/x-www-form-urlencoded
Connection: close

client-order-id=Value&oauth_consumer_key=merchantlogin&oauth_signature_method=HMAC-
↪SHA1&oauth_version=1.0&order_id=Value
```

7.27.6 Success Response Example

```
{
  "request_serial_number": "00000000-0000-0000-0000-000001fd4472",
  "document_path": "https://sandbox.doc2.com/paynet/api/v2/download/card-ins-
  ↪12345678912345678912"
}
```

Note:

The received link must be displayed in the Payer's browser.

Warning:

The file is available for a single download only.

7.27.7 Fail Response Example

```
HTTP/1.1 403
Server: server
Date: Fri, 14 Feb 2025 10:27:47 GMT
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Length: 102

type=error
&serial-number=00000000-0000-0000-0000-000002f38fde
&error-message=Forbidden
&error-code=-1
```

7.27.8 Request Builder

7.28 /api/v4/sale

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [SBP Sale Request Parameters](#)
- [Additional Parameters](#)
 - [For Connecting Party](#)

- [For Payment Institutions](#)

- [Response Parameters](#)
- [Request with Cardholder Data Example](#)
- [Request with Card Recurring Payment ID Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.28.1 Introduction

Sale is initiated through HTTPS POST request to the [URL](#) with the [required parameters](#) depending on a certain sale Use-Case (see below). Use [OAuth RSA-SHA256](#) for authentication.

7.28.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/sale/ENDPOINTID	https://sandbox.doc2.com/paynet/api/v4/sale/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v4/sale/group/ENDPOINTGROUPID	https://sandbox.doc2.com/paynet/api/v4/sale/group/ENDPOINTGROUPID

7.28.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#)

Acquirer can redefine the necessity of some fields so they become required instead of optional.

Leading and trailing whitespace in input parameters will be omitted.

Note:

By default, the control parameter in [Connecting Party Callbacks](#) and [Final Redirect](#) is signed with SHA1.

To get HMAC-SHA256 signature in these requests from Payment Gateway, please contact support manager.

Warning: The following characters must be escaped in the parameter values: & + “.

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10

continues on next page

Table 100 – continued from previous page

Parameter Name	Description	Value
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45
cvv2	Payer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card.	Necessity: Required Type: Numeric Length: 3-4
credit_card_number	Payer's credit card number. Send either combination of credit_card_number, card_printed_name, expire_month and expire_year or card_recurring_payment_id, not all.	Necessity: Conditional Type: Numeric Length: 20
card_recurring_payment_id	Payer's tokenized cardholder's data ID. Send either card_recurring_payment_id or combination of credit_card_number, card_printed_name, expire_month and expire_year, not all. To create card_recurring_payment_id see /api/v4/create-card-ref.	Necessity: Conditional Type: Long Length: 20
card_printed_name	Cardholder name, printed on the bank card.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 100 – continued from previous page

Parameter Name	Description	Value
expire_month	Bank card expiration month.	Necessity: Conditional Type: Numeric Length: 2
expire_year	Bank card expiration year.	Necessity: Conditional Type: Numeric Length: 4
first_name	Payer's first name. Parameter necessity depends on the Acquiring channel and necessity of it must be clarified with Tech Support.	Necessity: Required Type: String Length: 50
last_name	Payer's last name. Parameter necessity depends on the Acquiring channel and necessity of it must be clarified with Tech Support.	Necessity: Required Type: String Length: 50
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Conditional Type: String Length: 2-3
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status . Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command . Pass http://https://doc2.codetime.net if you have no need to return payer anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url , not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 100 – continued from previous page

Parameter Name	Description	Value
redirect_success_url	URL, where the Payer is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put http://https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
redirect_fail_url	URL, where the Payer is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass http://https://doc2.codetime.net if there is no need to redirect Payer anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
ssn	Last four digits of the payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYYM-MDD.	Necessity: Optional Type: Numeric Length: 8
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15

continues on next page

Table 100 – continued from previous page

Parameter Name	Description	Value
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128
purpose	Destination to where the payment goes. It is useful for the Connecting Party's who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 100 – continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters. This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
merchant_data	Custom Connecting Party details, which can be attached to the transaction and passed back in the status response, Connecting Party callback parameters or server_callback_url. Additional information of transaction which may be useful in Connecting Party's external system, e.g. VIP customer, TV promo campaign lead. Information returns in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
dapi_imei	Unique device identifier.	Necessity: Optional Type: String Length: 32
mini-mum_transaction_amount	This parameter can be used to limit the minimum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10

continues on next page

Table 100 – continued from previous page

Parameter Name	Description	Value
maximum_transaction_amount	This parameter can be used to limit the maximum transaction amount, if transaction amount is submitted by Payer on the form. Contact support manager to enable this feature. Value format is the same as in the amount parameter.	Necessity: Optional Type: Numeric Length: 10
customer_level	Customer level in CMS system.	Necessity: Optional Type: Varchar Length: 32
customer_id	Customer ID in CMS system. Required if transaction goes via CMS (Payment Gateway mode).	Necessity: Optional Type: Int Length: 10
merchant_customer_identifier	Connecting Party Customer ID in CMS system. Required if transaction goes via CMS (CRM mode).	Necessity: Optional Type: Varchar Length: 64
recurring-payment-id	Recurring Payment ID can be sent instead of cardholder data. CVV for native is not needed. Customer Data can be updated via /api/v4/update-recurring-payment/ . Recurring Payment ID creation is initiated through HTTPS POST request by using URLs and the parameters specified below. Use OAuth RSA-SHA256 for authentication.	Necessity: Conditional Type: Long

7.28.4 SBP Sale Request Parameters

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Required Type: String Length: 128

continues on next page

Table 101 – continued from previous page

Parameter Name	Description	Value
order_desc	Brief order description.	Necessity: Required Type: String Length: 64k
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Required Type: String Length: 256
city	Payer's city.	Necessity: Required Type: String Length: 50
zip_code	Payer's ZIP code.	Necessity: Required Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Required Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Required Type: String Length: 15

continues on next page

Table 101 – continued from previous page

Parameter Name	Description	Value
email	Payer's e-mail address.	Necessity: Required Type: String Length: 50
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Required Type: String Length: 45
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. <ENDPOINTID ENDPOINTGROUPID> (See: Request URL) 2. Request parameter: client_orderid 3. Request parameter: amount (in minor units) 4. Request parameter: email 5. merchant_control (Control key assigned to Connecting Party account in the Doc2.0 gateway system).	Necessity: Required Type: String Length: 40
redirect_url	URL, where the Payer is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status . Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command . Pass http://https://doc2.codetime.net if you have no need to return payer anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 101 – continued from previous page

Parameter Name	Description	Value
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
site_url	The URL of the E-commerce entity, where the payment is originated from.	Necessity: Optional Type: String Length: 128
purpose	Destination to where the payment goes. It is useful for the Connecting Party's who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 101 – continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
merchant_data	Custom Connecting Party details, which can be attached to the transaction and passed back in the status response , Connecting Party callback parameters or server_callback_url. Additional information of transaction which may be useful in Connecting Party's external system, e.g. VIP customer, TV promo campaign lead. Information returns in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
dapi_imei	Unique device identifier.	Necessity: Optional Type: String Length: 32

7.28.5 Additional Parameters

For Connecting Party

Note:

Browser data for 3DS 2.X is gathered by Connecting Party system on 3DS authentication stage. For some processing channels, however, the browser data and/or Connecting Party URL for 3DS challenge results must be provided in initial transaction request. Please

contact Support manager to clarify if these parameters should be included in request parameters.

The Connecting Party's site needs to accurately populate the browser information for each transaction. This data can be obtained by Connecting Party's servers. Ensure that the data is not altered or hard-coded, and that it is unique to each transaction.

Parameter Name	Description	Value
ipaddress	IP address of the browser as returned by the HTTP headers to the 3DS Requestor.	Necessity: Required Type: String Length: 45
customer_browser_accept_headers	Exact content of the HTTP accept headers as sent to the 3DS Requestor from the Cardholder's browser.	Necessity: Required Type: String Length: 2048
customer_browser_javascript_enabled	Boolean that represents the ability of the cardholder browser to execute JavaScript.	Necessity: Required Type: Boolean Length: -
customer_browser_accept_language	Value representing the browser language as defined in IETF BCP47.	Necessity: Required Type: String Length: 8
customer_browser_user_agent	Exact content of the HTTP user-agent header.	Necessity: Required Type: String Length: 2048
tds_areq_notification_url, alias tds_cres_notification_url	Fully qualified URL of Connecting Party system that will receive the CRes message or Error Message. This CRes message must be sent to Connecting Party. See details here Upload CRes Result .	Necessity: Optional Type: String Length: 256
customer_browser_info	If true, then the fields below must be present.	Necessity: Optional Type: Boolean Length: -
customer_browser_color_depth	Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: String Length: 2

continues on next page

Table 102 – continued from previous page

Parameter Name	Description	Value
cus-tomer_browser_java_enabled	Boolean that represents the ability of the cardholder browser to execute Java. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: Boolean Length: -
cus-tomer_browser_screen_height	Total height of the Cardholder's screen in pixels. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: Numeric Length: 6
cus-tomer_browser_screen_width	Total width of the cardholder's screen in pixels. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: Numeric Length: 6
cus-tomer_browser_time_zone	Time-zone offset in minutes between UTC and the Cardholder browser local time. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. Becomes required when browser_javascript_enabled = true.	Necessity: Optional Type: String Length: 5

For Payment Institutions

The PSP or Acquirer can fill the 3DS results for each transaction, if 3DS authentication is performed on their side.

Parameter Name	Description	Value
tds_authentication_result_type	Type of result. Possible values are: - SIMPLE	Type: String Length: 6

continues on next page

Table 103 – continued from previous page

Parameter Name	Description	Value
tds_authentication_result_authentication_type	Authentication Type. Indicates the type of authentication method the Issuer will use to challenge the Cardholder, whether in the ARes message or what was used by the ACS when in the RReq message. Possible values are: - 01 = Static - 02 = Dynamic - 03 = OOB - 04 = Decoupled - 05-79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) - 80-99 = Reserved for DS use	Type: String Length: 2
tds_authentication_result_authentication_value	Authentication Value. Payment System-specific value provided by the ACS or the DS using an algorithm defined by Payment System. Authentication Value may be used to provide proof of authentication. A 20-byte value that has been Base64 encoded, giving a 28-byte result	Type: String Length: 19-28
tds_authentication_result_transaction_id	transaction_id.0.2 or dsTransID for 2.1.0/2.2.0	Type: String Length: 19-36

continues on next page

Table 103 – continued from previous page

Parameter Name	Description	Value
tds_authentication_result_transaction_status	Transaction Status. Indicates whether a transaction qualifies as an authenticated transaction or account verification. Possible values are: - Y = Authentication Verification Successful - N = Not Authenticated/Account Not Verified, Transaction denied - U = Authentication/Account Verification Could Not Be Performed, Technical or other problem, as indicated in ARes or RReq - A = Attempts Processing Performed, Not Authenticated/Verified, but a proof of attempted authentication/verification is provided - C = Challenge Required, Additional authentication is required using the CReq/CRes - D = Challenge Required, Decoupled Authentication confirmed - R = Authentication/ Account Verification Rejected, Issuer is rejecting	Type: String Length: 1
tds_authentication_result_message_version	Message Version Number. Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction. Possible values are: - 1.0.2 - 2.1.0 - 2.2.0	Type: String Length: 5

7.28.6 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter Name	Description
type	The type of response. May be async-response, validation-error, error etc.. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is declined or error this parameter contains the reason for decline or error details.
error-code	The error code in case of declined or error status.
end-point-id	Endpoint id used for the transaction.

7.28.7 Request with Cardholder Data Example

```
POST /paynet/api/v4/sale/39509 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
↪ "aGDxlip0s21lzMdyp0TGmClTvbBgnaUh", oauth_signature=
↪ "R6uBD0TnC75UmYVFqEf70YX9tKK7iRmBR
↪ %2FoJGPCUTwzzyjtkAg6jk8fk2IEbzeadqTLDgckrXShSsD5Ke8%2FP3ZqPwwh30MT5XK9MWVcDGavjKQi
↪ %2BnXfJdr4JRbhPAk1YL3IJL04TTTmAoJHFwu0EKNdzndAMQ0cWay4Hg5evPcDpJEK0a2ZaKnSMjh9UMmttvzDL4Cb2INDH
↪ %2F5VadKPji%2FbDAxxWVT%2FQ9HVJcVSDHlxFniLBgyJr4ZIg4R0wtn03hb0C
↪ %2FC5rf7USgjkbpkBUejqjk4Wgw%2BJHMFJ4PJQM0beYn1ctmVGg27M0p8UWyi2i3PsAy8FbPf
↪ %2Ff6sMTDhx8MABfbIoJHgrvke%2B3flfS%2BUmtCo5pQ%2F9blWvVmB%2BjJz%2BqznYEAUk
↪ %2BHj7ekSVzMvpf5LNDeIwpwSVEZpp5Ig80yKZcWe9i8bbmFDsWehSGpiVkaraWsLCToras3lZy73Hp5z6SN8Uc98f
↪ %2BXUF8R0JvSysAfzDU2BJrDhNo1tKjuhY%2FEQzEinYe0zCSsSH0hkl2yU2KwWxbgWXb%2FoMEHIuuZ5L
↪ %2Baa%2FGTLCb%2Bm5wJ8n9QeDSuLHM3VmTP%2F7wFXa5DwCQCDy3Lu7v0WVFhLirX0MH6V
↪ %2Fe5SZ1yQHo0uH2N7F72DGKrcNkvkLaZCB4NE8AVqISxB484AB5Mo0pC3iZzAPWNo14liezgrlacarwSNsqSwmGBfvCWakxsZ
↪ %2BRgt06SkLAgpPcGDYWjziDvT6uQvAQG3R5a5v60grjZn0PQqsWb4qT7nlirHLwXXrSL46HTRyCl9q4QCuuFKPZA0LOW3y70
↪ %2BuI%2FZHXzD0eHGIhxBwa%2B
↪ %2BrpYhwDzmx1736gnNZ6dBbzKz5hH20BW0zvaZ93mKDH32Ri62D4yvAf64ElvSdNSImImLtEstRcws9YBsdCAxJw0TXICclwN
↪ %2FUSjihshg2NDmFJe4TNZ7TZRzLD5WKR6FxeTH6xYq1RJvoHiC0SUyVKjX7knFp%2FIEx3fdpMRD8ZCT
↪ %2FgUw5Cu34q87B7xVLHjEhn9%2FV3lacVDJ48F1a56urv%2FI2cahSsIz288g1V1ThrusN7TmdIsoZ
↪ %2F7XqKxht3p34N0e7LDUVils42ySmZil78GVvr9wR2Gt0pa8FT31F0LnyVxCiA3Fqj5LNM3QunElqSaNXI
↪ %2BtwlzkFGG4oklwb7bil0in0uzYlb4ymz7x6o708dJmCeUpkfCUXFrQnpmmHGBg0VgcKS5S769otXsXICqMP
↪ %2BDU2x5dajQvbIlzzm4aKF4qm2jxSowyd0WU1Y28Q%3D%3D", oauth_signature_method="RSA-
↪ SHA256", oauth_timestamp="1661493794", oauth_version="1.0"
Content-Length: 305
```

(continues on next page)

(continued from previous page)

```

Content-Type: application/x-www-form-urlencoded
Connection: close

credit_card_number=4538977399606732
&card_printed_name=John
&expire_month=01
&expire_year=2042
&cvv2=123
&client_orderid=34T43R77N
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100%20Main%20st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&amount=156
&email=john.smith@gmail.com
&currency=RUB
&ipaddress=65.153.12.232
&site_url=https://doc2.codetime.net
&purpose=user_account1
&redirect_url=http%3A%2F%2Fhttps://doc2.codetime.net%2Fdoc%2Fdummy.htm
&server_callback_url=https%3A%2F%2Fhttpstat.us%2F200
&merchant_data=VIP customer
&dapi_imei=123

```

Request for SBP Example

```

POST /paynet/api/v4/sale/39509 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↳ "aGDxlip0s21lzMdyp0TGmClTvbBgnaUh", oauth_signature=
  ↳ "R6uBD0TnC75UmYVFqEf70YX9tKK7iRmBR
  ↳ %2FoJGPCUTwzyjtkAg6jK8fk2I0EbzeadqTLDgckrXShSsD5Ke8%2FP3ZqPwwh30MT5XK9MWVcDGavjKQi
  ↳ %2BnXfJdr4JRbhPAk1YL3IJL04TTTmAoJHFWu0EKNdzndAMQ0cWay4Hg5evPcDpJEK0a2ZaKnSMjh9UMmttvzDL4Cb2INDH
  ↳ %2F5VadKPji%2FbDAxxWVT%2FQ9HVJcVSDHlxFniLBgyJr4ZIg4R0wtn03hb0C
  ↳ %2FC5rf7USgjkbpkBUejqjk4Wgw%2BJHMFJ4PJQM0beYn1ctmVGg27M0p8UWyi2i3PsAy8FbPf
  ↳ %2Ff6sMTDhx8MABfbIoJHgrvke%2B3flfS%2BUmtCo5pQ%2F9blWvVmB%2BjJz%2BqznYEAUK
  ↳ %2BHj7ekSVzMvpf5LNDeIwpwSVEZpp5Ig80yKZcWe9i8bbmfDsWehSGpiVkaraWsLCToras3LZy73Hp5z6SN8Uc98f
  ↳ %2BXUF8R0JvSysAfzDU2BJrDhNo1tKjuhY%2FEQzEinYe0zCSsSH0hkl2yU2KwWxbgWXB%2FoMEHIuuZ5l
  ↳ %2Baa%2FGTlScb%2Bm5wJ8n9QeDSulHM3VmTP%2F7wFXa5DwCQCDy3Lu7v0WVFhLirX0MH6V
  ↳ %2Fe5SZ1yQHo0uH2N7F72DGKrcNkvkLaZCB4NE8AVqISxB484AB5Mo0pC3iZzAPWNo141iezgrlacarwSNsqSwmGBfvCWakxsZ
  ↳ %2BRgt06SkLAgpPcGDYWjziDvT6uQvAQG3R5a5v60grjZNn0PQqsWb4qT7nlirHLwXXrSL46HTRyCl9q4QCuufKPZA0LOW3y70
  ↳ %2BuI%2FZXHXD0eHGIhxBwa%2B
  ↳ %2BrpYhwDzmx1736gnNZ6dBbzKz5hH20BW0zvaZ93mKDH32Ri62D4yvAf64ElvSdNSImImLtEstRcws9YBsdCAxJw0TXICclwN

```

(continues on next page)

(continued from previous page)

```

→%2FUSjihshg2NDmFJe4TNZ7TZrZLD5WKR6FxeTH6xYq1RJvoHiC0SUyVKjX7knFp%2FIE3fdpMRD8ZCT
→%2FgUw5Cu34q87B7xVLHjEhn9%2FV3lacVDJ48F1aS6urv%2FI2cahSsIz288g1V1ThhrusN7TmdIsoZ
→%2F7XqKxht3p34N0e7LDUVils42ySmZil78GVvr9wR2Gt0pa8FT31F0LnyVxCiA3Fqj5LNM3QunElqSaNXI
→%2BtwlzKFGg4oklwb7bil0in0uzYlb4ymz7x6o708dJmCeUpkfCUxFrQnpmmHGBg0VgcKS5S769otXsXICqMP
→%2BDU2x5dajQvbIlzzm4aKF4qm2jxSowyd0WU1Y28Q%3D%3D", oauth_signature_method="RSA-
→SHA256", oauth_timestamp="1661493794", oauth_version="1.0"
Content-Length: 305
Content-Type: application/x-www-form-urlencoded
Connection: close

&client_orderid=34T43R77N
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100%20Main%20st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=+12063582043
&cell_phone=+19023384543
&amount=156
&email=john.smith@gmail.com
&currency=RUB
&ipaddress=65.153.12.232
&site_url=https://doc2.codetime.net
&purpose=user_account1
&redirect_url=http%3A%2F%2Fhttps://doc2.codetime.net%2Fdoc%2Fdummy.htm
&server_callback_url=https%3A%2F%2Fhttpstat.us%2F200
&merchant_data=VIP customer
&dapi_imei=123

```

7.28.8 Request with Card Recurring Payment ID Example

```

POST /paynet/api/v4/sale/39509 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
→"aGDxlip0s21lzMdyp0TGmClTvbBgnauH", oauth_signature=
→"R6uBD0TnC75UmYVFqEf70YX9tKK7iRmBR
→%2FoJGPCUTwzzyjtkAg6jK8fk2I0EbzeadqTLDgckrXShSsD5Ke8%2FP3ZqPwwh30MT5XK9MWvcDGavjKQi
→%2BnXfJdr4JRbhPAk1YL3IjL04TTTmAoJHFWu0EKNdzndAMQ0cWay4Hg5evPcDpJEK0a2ZaKnSMjh9UMmttvzDL4Cb2INDH
→%2F5VadKPji%2FbDAxxWVT%2FQ9HVJcVSDHlxFniLBgyJr4ZIg4R0wtn03hb0C
→%2FC5rf7USgjkbpkBUejqjk4Wgw%2BJHMFJ4PJQM0beYn1ctmVGg27M0p8UWyi2i3PsAy8FbPf
→%2Ff6sMTDhx8MABfbIoJHgrvke%2B3flfS%2BUmtCo5pQ%2F9blWvVmB%2BjJz%2BqznYEAUK
→%2BHj7ekSVzMvpf5LNDeIwpwSVEZpp5Ig80yKZCWe9i8bbmfDsWehSGpiVkarawsLCToras3LZy73Hp5z6SN8Uc98f
→%2BXUF8R0JvSysAfzDU2BJrDhNo1tKjuhY%2FEQzEinYe0zCSsSH0hkl2yU2KwWxbgWxb%2FoMEHIuuZ5l
→%2Baa%2FGTlScb%2Bm5wJ8n9QeDSulHM3VmTP%2F7wFXa5DwCQCdy3Lu7v0WVFhLirX0MH6V
→%2Fe5SZ1yQHo0uH2N7F72DGKrcNkvkLaZCB4NE8AVqISxB484AB5Mo0pC3iZzAPWNo14liezgrlacarwSNsqSwmGBfvCWakxsZ
→%2BRgt06SkLAgpPcGDYWjziDvT6uQvAQG3R5a5v60grjZn0PQqsWb4qT7nlirHLwXXrSL46HTRyCl9q4QCuufKPZA0LOW3y70
→%2BuI%2FZXHZD0eHGIhxBwa%2B

```

(continues on next page)

(continued from previous page)

```

↪%2BrpYhwDzmx1736gnNZ6dBbzKz5hH20BW0zvaZ93mKDH32Ri62D4yvAf64ElvSdNSImImLtEstRcws9YBsdCAxJw0TXICclwN
↪%2FUSjihshg2NDmFJe4TNZ7TZRzLD5WKR6FxeTH6xYq1RJvoHiC0SUyVKjX7knFp%2FIE3fdpMRD8ZCT
↪%2FgUw5Cu34q87B7xVLHjEhn9%2FV3lacVDJ48F1aS6urv%2FI2cahSsIz288g1V1ThhrusN7TMDIsoZ
↪%2F7XqKxht3p34N0e7LDUVils42ySmZil78GVvr9wR2Gt0pa8FT31F0LnyVxCiA3Fqj5LNM3QunElqSaNXI
↪%2BtwlzKFGq4oklwb7bil0in0uzYlb4ymz7x6o708dJmCeUpkfCUxFrQnpmmHGBg0VgcKS5S769otXsXICqMP
↪%2BDU2x5dajQvbIlzzm4aKF4qm2jxSowyd0WU1Y28Q%3D", oauth_signature_method="RSA-
↪SHA256", oauth_timestamp="1661493794", oauth_version="1.0"
Content-Length: 305
Content-Type: application/x-www-form-urlencoded
Connection: close

```

```

card_recurring_payment_id=1491954
&cvv2=123
&client_orderid=34T43R77N
&order_desc=Test Order Description
&amount=777
&currency=USD
&ipaddress=65.153.12.232
&redirect_url=http://doc2.doc2.com/doc/dummy.htm
&server_callback_url=https://httpstat.us/200
&control=80761544c64373d1624240add048d36d42fe528a

```

7.28.9 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 11:03:17 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=async-response
&serial-number=00000000-0000-0000-0000-000002d9b22a
&merchant-order-id=inv4097763
&paynet-order-id=6768788
&end-point-id=22903

```

7.28.10 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 13:14:40 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=validation-error
&serial-number=00000000-0000-0000-0000-000002b36f64
&merchant-order-id=inv4097763
&error-message=End+point+with+id+22903+not+found
&error-code=3
```

7.28.11 Postman Collection

7.28.12 Request Builder

Enter your private key in PKCS#1 container to use debug. See [RSA-SHA256](#) for details.

7.29 /api/v4/create-card-ref

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.29.1 Introduction

Card reference ID and Recurring Payment ID creation is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.29.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/create-card-ref/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/create-card-ref/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v4/create-card-ref/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v4/create-card-ref/group/ENDPOINTGROUPID

7.29.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Sale Request Parameters	Description
login	Connecting Party's login name.
client_orderid	Connecting Party's order identifier of the transaction.
orderid	Transaction order identifier assigned by Payment Gateway.

Connecting Party has to supply orderid and client_orderid associated with the first payment transaction. It emphasizes that the first payment is a mandatory step to process recurring payments. To authorize the credit card the information about it must be sufficient and an initial payment must be in [final status](#).

7.29.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Registration Response Parameter	Description
type	The type of response. May be create-card-ref-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
card-ref-id	Card reference ID used in subsequent recurring payments.
unq-card-ref-id	Unique card reference ID to each PAN. It can be used by Connecting Party for loyalty programs or fraud control.
recurring-payment-id	Payer's tokenized cardholder's data ID, referred as Recurring Payment ID (RPI).
dst-card-ref-id	Receiver's card reference ID used in subsequent recurring payments. Can be used only in transfer APIs.
dst-unq-card-ref-id	Receiver's unique card reference ID to each PAN. It can be used by Connecting Party for loyalty programs or fraud control. Can be used only in transfer APIs.
dst-recurring-payment-id	Receiver's tokenized cardholder's data ID, referred as Recurring Payment ID (RPI). Can be used only in transfer APIs.
error-message	If status is declined or error this parameter contains the reason for decline.
error-code	The error code is case of declined or error status.
status	See Status List for details.

7.29.5 Request Example

```
POST /paynet/api/v4/create-card-ref/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: 0Auth oauth_consumer_key="TestMerchant", oauth_nonce=
→"KT6cZmuVGqg0V6Jm2RE3q4o79KXC1v2q", oauth_signature="KucF0eYk3WZCV7oKw0i1z6PR
→%2BkHxiZwPucD6Sx%2BX2mV
→%2BbaidPy9K9USH8ciMKM60NN11LYYjywdErB1uTiDFFbQ8ZKs8M1smaMP0aHdaApceT0ldhE7u3BzBTKYhBc
→%2BwnksZz9Wyz8%2B39lHCI0Do0KZmNmXCTjjZmlx%2FrFNkK
→%2FhwJV9KwqlnPbA5QZTkF6860000LHFy3Prx649AIRgsrqDLb5
→%2FgHL9M8fSScVUPnGdLGJ2hSgKJFpIOFibT0nC89Xg8odn1hR9WIa1650glaqZntSkocBzXAK0Ka7kIbS0ZW1sFCiBjksy6o1
→%2F9cC9t86RoEY1QhVYuv0LztQm1dLhpRy%2FP0L9LCmIz03B%2FUB2wJUXPkEyFsSLZVeqQl
→%2B0IukljV6Cr1ZfuyUktbvXJsnod5AK%2FsV2GaxEf%2BttqqWv%2FFNjPLoUZYrPB6rKsIpw
→%2F0ftinIwIXzLY3FMmbKQd6zxnMJLJm7M2s6cQFGiAnfgvZFAMZhugBuuigy4T9Ckq1t5N9vQkl2htDv0TTnswx50wpF
```

(continues on next page)

(continued from previous page)

```
→%2F70KiXTVFNq0E%2FCj%2F07ZwZbxbD%2FMxKh0hfNMME1jGxzgI0wEj1166eKpxnC0k%2BTLYTJvNW5
→%2BbKuGnU43Q2Nmga0aQ007NKRfIV%2FWk6e%2BUuG048wGdi0CiKxS9hpnvpjyPLc%2BsA%3D", oauth_
→signature_method="RSA-SHA256", oauth_timestamp="1673335450", oauth_version="1.0"
Content-Length: 40
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=34T43R77N
&orderid=6868305
```

7.29.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Tue, 10 Jan 2023 07:24:36 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 174

type=create-card-ref-response
&serial-number=00000000-0000-0000-0000-000002de3113
&card-ref-id=1461608
&recurring-payment-id=1491863
&dst-card-ref-id=1461608
&status=approved
```

7.29.7 Fail Response Example

```
HTTP/1.1 403
Server: server
Date: Tue, 10 Jan 2023 07:34:08 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>403</title>

  <style type="text/css">
    body {
```

(continues on next page)

(continued from previous page)

```
        font-family: Arial, sans-serif;
        font-size: 130%;
        background-color: #eee;
    }

    p {
        margin: 10em auto 0;
        width: 500px;
        border: 1px solid gray;
        text-align: center;
        vertical-align: middle;
        padding: 40px 20px;
        background-color: #fff;
        -webkit-border-radius: 20px;
        -moz-border-radius: 20px;
        border-radius: 20px;
    }
</style>
</head>

<body>
    <p>Access is denied</p>
</body>
</html>
```

7.29.8 Postman Collection

7.29.9 Request Builder

Insert PKCS#1 PEM private key for sandbox environment in the field below. Request builder supports up to 4096 key length.

7.30 /api/v4/create-recurring-payment

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.30.1 Introduction

/api/v4/create-recurring-payment is a synchronous API command. If request is accepted with no errors, the Payment Gateway creates a recurring payment profile and assigns a new recurring-payment-id to it. The recurring transactions will be processed according to the created recurring payment profile using the payment data saved in this profile.

Create recurring payment is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.30.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/create-recurring-payment/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/create-recurring-payment/ENDPOINTID

7.30.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Note:

If the recurrent has switched to Stopped, you can update its schedule using the start-date and finish-date parameters. However, it can only be resumed via the UI by clicking on Resume.

Type native on UI means that The recurring payment setup and the actual charges are in the same acquirer.

To create recurring payment profile with automatic recurring schedule, send interval and period parameters. To create recurring payment profile without automatic recurring schedule, do not send interval and period parameters. To stop the automatic recurring schedule, use [/api/v4/update-recurring-payment](#). Available only for SRC.

Parameter Name	Description	Value
rp_card_type	SRC - Sender's source card. DST - Receiver's destination card.	Necessity: Required Type: Enum Length: 3
client-orderid	Connecting Party order ID. Supported for SRC and DST type.	Necessity: Required Type: String Length: 128
credit-card-number	Payer's credit card number. Supported for SRC and DST type.	Necessity: Required Type: Numeric Length: 19
cvv2	Payer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number printed on the back of the card in the signature area.	Necessity: Optional Type: Numeric Length: 3-4
card-printed-name	Payer's card printed name. Required for SRC, optional for DST.	Necessity: Conditional Type: String Length: 128
expire-year	Payer's card expire year. Required for SRC, optional for DST.	Necessity: Conditional Type: Numeric Length: 4
expire-month	Payer's card expire month. Required for SRC, optional for DST.	Necessity: Conditional Type: Numeric Length: 2
amount	Amount of currency must be the same as currency on the project assigned. Upon reaching finish date, Recurring payment will go into stop status. Supported for SRC and DST type. Required if amount-from and amount-to or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10

continues on next page

Table 109 – continued from previous page

Parameter Name	Description	Value
currency	Currency type. Supported for SRC and DST type.	Necessity: Required Type: String Length: 3
country	Payer`s country. Not supported for DST.	Necessity: Required Type: String Length: 2
city	Payer`s city. Not supported for DST.	Necessity: Required Type: String Length: 128
zip-code	Payer`s zip-code. Not supported for DST.	Necessity: Required Type: String Length: 10
address1	Payer`s address. Not supported for DST.	Necessity: Required Type: String Length: 256
first-name	Payer`s first-name. Not supported for DST.	Necessity: Required Type: String Length: 128
last-name	Payer`s last-name. Not supported for DST.	Necessity: Required Type: String Length: 128
email	Payer`s email. Not supported for DST.	Necessity: Required Type: String Length: 128

continues on next page

Table 109 – continued from previous page

Parameter Name	Description	Value
amount-from	If the combination of amount-from and amount-to is chosen, every charge will be of random amount between these two numbers. Supported for SRC and DST type. Required if amount or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10
amount-to	If the combination of amount-from and amount-to is chosen, every charge will be of random amount between these two numbers. Supported for SRC and DST type. Required if amount or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10
amount-sequence	If amount sequence is chosen, client will be charged amounts from this list. Example of setting up amount sequence: 10.5, 24.6, 32.0. If repeats number is higher than amount sequence number of elements, every new charge will be with last amount in amount sequence. In order for charges to begin from the first amount in the chain, current repeats number must be set as 0. Supported for SRC and DST type. Required if amount-from and amount-to or amount are not used.	Necessity: Conditional Type: Numeric Length: 10
period	Possible values are: day, week and month. In case if daily is chosen, client will be charged every day. If week - every 7 days. If monthly is chosen, client will be charged on the same date of the month, from the starting date, no matter how many days there are in a month. Interval and period can only be specified or omitted together. Not supported for DST.	Necessity: Conditional Type: String Length: 32
interval	Interval is a multiplier applied to the period. For example, if interval of 2 and period 'Daily' is selected, client will be charged once every 2 days. Interval and period can only be specified or omitted together. Not supported for DST.	Necessity: Conditional Type: Int Length: -
order_desc	Description of Recurring payment. Supported for SRC and DST type.	Necessity: Optional Type: String Length: 65K

continues on next page

Table 109 – continued from previous page

Parameter Name	Description	Value
customer-ip	Payer's IP address. Supported for SRC and DST type.	Necessity: Optional Type: String Length: 45
ssn	Social security number field. Not supported for DST.	Necessity: Optional Type: String Length: 32
birthday	Payer's birthday date. Not supported for DST.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8
phone	Payer's phone number. Not supported for DST.	Necessity: Optional Type: String Length: 128
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia. Not supported for DST.	Necessity: Optional Type: String Length: 2-3
start-date	Date, when first charge is scheduled. If start date is set as a current date and type is set as auto, first charge will be made today. Supported for SRC and DST type.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8
finish-date	Date, when the Payer will be charged last time. Supported for SRC and DST type.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8

continues on next page

Table 109 – continued from previous page

Parameter Name	Description	Value
max-repeats-number	Index of recurring transaction, first charge will hold the index of 0. Current repeats number increases even if a charge was unsuccessful. When current repeats number reaches max repeats number, Recurring payment goes into stop status and client is charged no more. If a charge was made automatically, no additional charges will be made(unless done manually), even if a recurring payment is stopped and rescheduled again. Supported for SRC and DST type.	Necessity: Optional Type: Int Length: -
purpose	Purpose of transaction. Not supported for DST.	Necessity: Optional Type: String Length: 128
notify_url	Notify url field. server_callback_url parameter can also be used. For more information please see Connecting Party Callbacks . Supported for SRC and DST type.	Necessity: Optional Type: String Length: 1024
server_callback_url	Connecting Party URL which will receive callback request once the transaction reaches final status. Connecting Party may use Server Callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. See callback details in Connecting Party callback parameters . Send either notify_url or server_callback_url, not both.	Necessity: Optional Type: String Length: 128

7.30.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Warning: /api/v4/create-recurring-payment is a synchronous API command. The response will be returned after the recurring payment profile will be created on Payment Gateway side.

Response Parameters	Description
type	The type of response. May be create-recurring-payment-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
recurring-payment-id	Recurring ID assigned to the order by Doc2.0.
status	If request is accepted, this parameter has approved value. This is not a transaction status.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is declined or error this parameter contains the reason for decline or error details.
error-code	The error code in case of declined or error status.

7.30.5 Request Example

```

POST /paynet/api/v4/create-recurring-payment/ HTTP/1.1
User-Agent: curl/8.4.0
Accept: */*
Authorization: 0Auth oauth_consumer_key="ErwinTestMerchant", oauth_nonce=
↪ "qdyD66xc8sDEQwo3r12VjJfJhA0Xuj60", oauth_signature="eZve%2FSvPCTBXtQM
↪ %2BEwTER0tsQE27gZJr36EThL8ECCrnWH9Xw8JxcysNEPTG5HfcYwR2IOjDk3uiSNFaIoT2bQT9XONQTl1JVTOhIiMrMJT7GT1
↪ %2F%2FIny0xmIc%2B79E3T8hkpH6J6VI%2BSRG162%2B1HBPc1u1SGGnhkkRYU621AUGu2FDrc4neob4QJSm
↪ %2BaF3a0AgWDGiLbeQ0Ivn3YJ441EKd99kBWdaCuUp0eSfcmh3U2Js66edCjWXDEW7uuywP%2BU
↪ %2FBEeq06XxoVNGE2NF2LARXgoNqfzz2Tpisq35%2FE7Yg
↪ %2F9hphww75HMoJCECe0yVIHmKXd8HBDNMyT7gg3w5bnKbo%2FURX0m
↪ %2Fpe7bxxkJ1mLNsKouHVkvjLZKpWVVUyC1%2BJ3x0BoEN5QK7xh0fNbV4Ue9TU0z59q0v
↪ %2B7lz5E1B4Tx09pa%2FmNCL4P2JgTosLvXNcdGn9Dq0TC0gb6202W81MoZAljJQew
↪ %2FckAUFTXBZK6eIXzA81BPQVejv8nIVsvHKNo2Ko0s3hJPKGaeVAN0gyUAPw0
↪ %2BsJI3bnN2CPSW92xysGRWJYBtsHhqdHmgtoDNvW7LFB6BzRk5GPa9iQju34l47GsCUUIs3
↪ %2ByipIl7Q3HnM%2Bet9L8JS17K02MwR6zLLxtD9UXHDq3pnmKzly6I%3D", oauth_signature_method=
↪ "RSA-SHA256", oauth_timestamp="1721312072", oauth_version="1.0"
Content-Length: 595
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

address1=1234%20Peace%20street
&amount=55
&birthday=1980-01-02
&card-printed-name=JOHN%20SMITH
&city=Chicago
&client-orderid=1575634981130
&country=US
&credit-card-number=4464920026265488
&currency=USD
&customer-ip=1.2.3.4
&cvv2=123
&email=john.smith%40example.com
&expire-month=12
&expire-year=2040
&finish-date=2040-01-01
&first-name=John
&interval=1
&last-name=Smith

```

(continues on next page)

(continued from previous page)

```
&max-repeats-number=1000
&notify-url=http%3A%2F%2Fexample.com%2Fnotify-me
&order_desc=testing%20purposes
&period=week
&phone=12345678
&purpose=No%20purpose%20at%20all
&rp_card_type=SRC
&ssn=1234
&start-date=2030-01-01
&state=IL
&zip-code=123456
&server_callback_url=https://httpstat.us/200
```

7.30.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 18 Jul 2024 14:16:20 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=create-recurring-payment-response
&serial-number=00000000-0000-0000-0000-000002f36d81
&recurring-payment-id=1492124
&status=approved
```

7.30.7 Fail Response Example

```
HTTP/1.1 500
Server: server
Date: Thu, 18 Jul 2024 14:24:28 GMT
Content-Length: 61
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

Internal server error [8485772c-21d6-42aa-b2eb-3b5ae4e07d19]
```

7.30.8 Request Builder

7.31 /api/v4/process-recurring-payment

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.31.1 Introduction

If request is accepted with no errors, the Payment Gateway finds the recurring payment profile by the provided recurring-payment-id and initiates a new recurring transaction using the payment data saved in this profile.

Process recurring payment is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.31.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/process-recurring-payment/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/process-recurring-payment/ENDPOINTID

7.31.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
client-orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
recurring-payment-id	Recurring ID assigned to the order by QA.	Necessity: Required Type: String Length: 10
amount	Amount of currency must be the same as currency on the project assigned. Upon reaching finish date, Recurring payment will go into stop status. Supported for SRC and DST type. Required if amount-from and amount-to or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10

7.31.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. Example: process-recurring-payment-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
status	If request is accepted, this parameter has processing value. This is not a transaction status.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.

7.31.5 Request Example

```
POST /paynet/api/v4/process-recurring-payment/ HTTP/1.1
User-Agent: curl/8.4.0
Accept: */*
Authorization: OAuth oauth_consumer_key="ErwinTestMerchant", oauth_nonce=
↪ "XGsxKTfd4uiaQK0SyXkHQwKxFqLCEKRz", oauth_signature="KX7KEjFUEJo8rLJ2PvusGDlu5cxqMm
↪ %2BU%2Bn
↪ %2F1KxH18QeZ57QghfnXyiXKCxwAr76uxMk9X10QL8YI4kuEHVmxThIomtIZML3yRLESneHNqZzlwXvTSG5o
↪ %2B7DTSdAeNGZ40PIbF4XAUzwp%2FQbpYGWBS4Yk36%2BPQbziJlsFa21gRo7e03fizxPEIYilalEq0T5uu
↪ %2F07Yci6NsSyGGMGo4vyDhqPMLuDcaTe0
```

(continues on next page)

(continued from previous page)

```
→%2Fzr51RfRa4jh2bT4tUriZLVrjYP0AJeHGLZdosTkqEyVAz5Q5yD67xmjUGLW4luez0WdR70wrfQ8JZ7QuXzueAb9TqyZfi
→%2BvCD4MrhXG299ujWUUtZ%2F8qi09p%2F7GmBiFSud7wgMjPG0C%2BijBEFIpsfiBFzd6
→%2FX7MBLAdTpHXrExt2C71dJysde31isAMVuIIuvXogY7YE0ZttWY1oYqcWDNSWXSUx5pQTjbBf1vyJB3CqtAFVFi6LwckqsYT
→%2B2IgFFhadEvf35XzFPv
→%2FEjsTqRLJuPYBa85mK9vaY7WJRtX89TjSdEPulFnz1EEFg5KHe5WxVp9GFw2XqLneqBu
→%2FwGAGg3NXRumBMHK3E
→%2Bh96cVdU5b3AkqVjS2Q36jpQtAhUIIGYyT02n4W9avhw2cHCJzErqGjALHJjkFQD1bhDRKyTm1Fqeh
→%2BplDaPwDKjy5EHY%3D", oauth_signature_method="RSA-SHA256", oauth_timestamp=
→"1721316898", oauth_version="1.0"
Content-Length: 407
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

recurring-payment-id=1492124
&amount=10
&client-orderid=1575634981130
```

7.31.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 18 Jul 2024 15:35:46 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 110

type=process-recurring-payment-response
&serial-number=00000000-0000-0000-0000-000002f36d8d
&status=processing
```

7.31.7 Fail Response Example

```
HTTP/1.1 500
Server: server
Date: Thu, 18 Jul 2024 15:39:05 GMT
Content-Length: 61
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

Internal server error [b2a06c5a-c770-4c01-b32c-e4fcb1da4cd6]
```

7.31.8 Request Builder

7.32 /api/v4/update-recurring-payment

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.32.1 Introduction

/api/v4/update-recurring-payment is a synchronous API command. If request is accepted with no errors, the Payment Gateway finds the recurring payment profile by the provided recurring-payment-id and updates this profile. The recurring transactions will be processed according to the updated recurring payment profile using the payment data saved in this profile.

Update recurring payment is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.32.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/update-recurring-payment/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/update-recurring-payment/ENDPOINTID

7.32.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Note:

The RPI details screen (recurring payment profile) contains information about linked cardholder and customer data, recurring schedule and processed transactions with this RPI. This screen also contains change history for this RPI. The change history is currently available for source cards (SRC) only.

Type native on UI means that The recurring payment setup and the actual charges are in the same acquirer.

If the recurrent has switched to Stopped, you can update its schedule using the start-date and finish-date parameters. However, it can only be resumed via the UI by clicking on Resume.

To stop the automatic recurring schedule, use [Request Parameters](#). Available only for SRC.

Parameter Name	Description	Value
client-orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
recurring-payment-id	Recurring ID assigned to the order by QA.	Necessity: Required Type: String Length: 128
credit-card-number	Payer's credit card number. card-printed-name can only be specified when card number is specified.	Necessity: Conditional Type: Numeric Length: 19
card-printed-name	Payer's card printed name. Required when card number is specified.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 115 – continued from previous page

Parameter Name	Description	Value
expire-year	Payer`s card expire year. Can only be specified when card number is specified. Required when card number is specified.	Necessity: Conditional Type: Numeric Length: 4
expire-month	Payer`s card expire month. Can only be specified when card number is specified. Required when card number is specified.	Necessity: Conditional Type: Numeric Length: 2
amount	Amount of currency must be the same as currency on the project assigned. Upon reaching finish date, Recurring payment will go into stop status. Supported for SRC and DST type. Required if amount-from and amount-to or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10
amount-from	If the combination of amount-from and amount-to is chosen, every charge will be of random amount between these two numbers. Supported for SRC and DST type. Required if amount or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10
amount-to	If the combination of amount-from and amount-to is chosen, every charge will be of random amount between these two numbers. Supported for SRC and DST type. Required if amount or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10
amount-sequence	If amount sequence is chosen, client will be charged amounts from this list. Example of setting up amount sequence: 10.5, 24.6, 32.0. If repeats number is higher than amount sequence number of elements, every new charge will be with last amount in amount sequence. In order for charges to begin from the first amount in the chain, current repeats number must be set as 0. Supported for SRC and DST type. Required if amount-from and amount-to or amount are not used.	Necessity: Conditional Type: Numeric Length: 10

continues on next page

Table 115 – continued from previous page

Parameter Name	Description	Value
period	Possible values are: day, week and month. In case if daily is chosen, client will be charged every day. If week - every 7 days. If monthly is chosen, client will be charged on the same date of the month, from the starting date, no matter how many days there are in a month. Interval and period can only be specified or omitted together. Not supported for DST.	Necessity: Conditional Type: String Length: 32
interval	Interval is a multiplier applied to the period. For example, if interval of 2 and period 'Daily' is selected, client will be charged once every 2 days. Interval and period can only be specified or omitted together. Not supported for DST.	Necessity: Conditional Type: Int Length: -
customer-ip	Payer`s IP address. Supported for SRC and DST type.	Necessity: Optional Type: String Length: 45
country	Payer`s country.	Necessity: Optional Type: String Length: 2
city	Payer`s city.	Necessity: Optional Type: String Length: 128
address1	Payer`s address.	Necessity: Optional Type: String Length: 256
first-name	Payer`s first-name.	Necessity: Optional Type: String Length: 128

continues on next page

Table 115 – continued from previous page

Parameter Name	Description	Value
last-name	Payer`s last-name.	Necessity: Optional Type: String Length: 128
order_desc	Description of Recurring payment.	Necessity: Optional Type: String Length: 65K
zip-code	Payer`s zip-code.	Necessity: Optional Type: String Length: 10
birthday	Payer`s birthday date.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8
email	Payer`s email.	Necessity: Optional Type: String Length: 128
ssn	Social security number field.	Necessity: Optional Type: String Length: 32
phone	Payer`s full international phone number, including country code. Not supported for DST.	Necessity: Optional Type: String Length: 128

continues on next page

Table 115 – continued from previous page

Parameter Name	Description	Value
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia. Not supported for DST.	Necessity: Optional Type: String Length: 2-3
start-date	Date, when first charge is scheduled. If start date is set as a current date and type is set as auto, first charge will be made today.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8
finish-date	Date, when the Payer will be charged last time.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8
max-repeats-number	Index of recurring transaction, first charge will hold the index of 0. Current repeats number increases even if a charge was unsuccessful. When current repeats number reaches max repeats number, Recurring payment goes into stop status and client is charged no more. If a charge was made automatically, no additional charges will be made(unless done manually), even if a recurring payment is stopped and rescheduled again.	Necessity: Optional Type: Int Length: -
purpose	Purpose of transaction.	Necessity: Optional Type: String Length: 128
notify_url	Notify url field. <code>server_callback_url</code> parameter can also be used. For more information please see Connecting Party Callbacks .	Necessity: Optional Type: String Length: 1024

continues on next page

Table 115 – continued from previous page

Parameter Name	Description	Value
server_callback_url	Connecting Party URL which will receive callback request once the transaction reaches final status. Connecting Party may use Server Callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. See callback details in Connecting Party callback parameters . Send either notify_url or server_callback_url, not both.	Necessity: Optional Type: String Length: 128

7.32.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Warning: /api/v4/create-recurring-payment is a synchronous API command. The response will be returned after the recurring payment profile will be created on Payment Gateway side.

Response Parameters	Description
type	The type of response. Example: update-recurring-payment-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
status	If request is accepted, this parameter has approved value. This is not a transaction status.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is declined or error this parameter contains the reason for decline or error details.
error-code	The error code in case of declined or error status.

7.32.5 Request Example

```
POST /paynet/api/v4/update-recurring-payment/ HTTP/1.1
User-Agent: curl/8.4.0
Accept: */*
Authorization: OAuth oauth_consumer_key="ErwinTestMerchant", oauth_nonce=
↳ "XBhUVVJU4N8fBnNRGfY91Z3wuZ6iNgjVM", oauth_signature="RIRufAZeoBZBMpVlr7Bt
↳ %2Bc6fDTUxZJgxjJULTo7Pm9zlGeZAl427VF6c%2B0lkGvZfATlooYBK
↳ %2FeoZaUU5gJITePycMmZr2gPzAk8xCielclB8N0j5Rp3ga0A%2B3uJDTgQdvsrosYK4tsES%2BPsR6qjhF
↳ %2FqWGHbpwCooXbMLwI9a9yMdkwmcRNQGPWAz7I%2FJ8gdDLvkjM0H8fZRp%2Bz%2FSAAd3%2FgX%2F
```

(continues on next page)

(continued from previous page)

```

→%2BCJv7bMn26h0GUxJua7u5GIKX6mmZ5FpV71xdy04ebPu4qAXGPGNKBZXLJGqzYfYVlW9XWKKkFhUA4RLYUZqnfokHA8uy3zb
→%2BYAiHxLmt%2BEq%2BlovLAXWENbIXvJYhRiX
→%2F3Q02cq2ZznAsanZQiyU7AT309lnLHuvKc2wbuFKr277lNR24cykI3ja%2FGMR%2F8T%2BXjZKFF
→%2F1sYVRGd93CQDx6NHnH98vp%2Bv3PMop0mLWwggY0ApmDBDsa8jYoU1TD0s6gNRTsXIyFiSwl3e48fNAp
→%2FjFZfUl90K8wGusNzrof05UdTPR%2B7zpv4jL1hd1XyN%2F4x7aNNQ28tm5LSoVU6t%2BQkvwRm
→%2FNxnDKMYEE9rq7s4Uq5KTzAmM89pAu52WFbDQXyoZgy3vm
→%2B0SsJDXH0IVXsDpLDt1zcJClWoeXxT0fq0cEojJhuosvbcI%2FIaEpxWH8%3D", oauth_signature_
→method="RSA-SHA256", oauth_timestamp="1721312815", oauth_version="1.0"
Content-Length: 624
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

address1=1234%20Peace%20street
&amount=55
&birthday=1980-01-02
&card-printed-name=JOHN%20SMITH
&recurring-payment-id=1492124
&city=Chicago
&client-orderid=1575634981130
&country=US
&credit-card-number=4464920026265488
&currency=USD
&customer-ip=1.2.3.4
&email=john.smith%40example.com
&expire-month=12
&expire-year=2040
&finish-date=2040-01-01
&first-name=John
&interval=1
&last-name=Smith
&max-repeats-number=1000
&notify-url=http%3A%2F%2Fexample.com%2Fnotify-me
&order_desc=testing%20purposes
&period=week
&phone=12345678
&purpose=No%20purpose%20at%20all
&ssn=1234
&start-date=2030-01-01
&state=IL
&zip-code=123456
&server_callback_url=https://httpstat.us/200

```

7.32.6 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Thu, 18 Jul 2024 14:27:39 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

```

(continues on next page)

(continued from previous page)

```
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 107

type=update-recurring-payment-response
&serial-number=000000000-0000-0000-0000-000002f36d82
&status=approved
```

7.32.7 Fail Response Example

```
HTTP/1.1 500
Server: server
Date: Thu, 18 Jul 2024 14:41:27 GMT
Content-Length: 61
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

Internal server error [00b20f25-2933-48e3-973b-0664114b8492]
```

7.32.8 Request Builder

7.33 /api/v4/create-recurring-payments

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.33.1 Introduction

If request is accepted with no errors, the Payment Gateway creates separate recurring payment profiles for each entry provided in the request and assigns new recurring-payment-id to each of them. The recurring transactions will be processed according to the created recurring payment profiles using the payment data saved in each profile.

Create recurring payments Multiple is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.33.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/create-recurring-payments/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/create-recurring-payments/ENDPOINTID

7.33.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Below is a description of each parameter that can be included in the CSV and added to payload parameter which will be used in the request.

The recurring payment profile details can be viewed only via UI.

Note:

If the recurrent has switched to Stopped, you can update its schedule using the start-date and finish-date parameters. However, it can only be resumed via the UI by clicking on Resume.

Type native on UI means that The recurring payment setup and the actual charges are in the same acquirer.

To create recurring payment profile with automatic recurring schedule, send interval and period parameters. To create recurring payment profile without automatic recurring schedule, do not send interval and period parameters. To stop the automatic recurring schedule, use [/api/v4/update-recurring-payments](#). Available only for SRC.

CSV Parameter Name	Description	Value
rp_card_type	SRC - Sender's source card. DST - Receiver's destination card.	Necessity: Required Type: Enum Length: 3
client-orderid	Connecting Party order ID. Supported for SRC and DST type.	Necessity: Required Type: String Length: 128
credit-card-number	Payer's credit card number. Supported for SRC and DST type.	Necessity: Required Type: Numeric Length: 19
cvv2	Payer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number printed on the back of the card in the signature area.	Necessity: Optional Type: Numeric Length: 3-4
card-printed-name	Payer's card printed name. Required for SRC, optional for DST.	Necessity: Conditional Type: String Length: 128
expire-year	Payer's card expire year. Required for SRC, optional for DST.	Necessity: Conditional Type: Numeric Length: 4
expire-month	Payer's card expire month. Required for SRC, optional for DST.	Necessity: Conditional Type: Numeric Length: 2
amount	Amount of currency must be the same as currency on the project assigned. Upon reaching finish date, Recurring payment will go into stop status. Supported for SRC and DST type. Required if amount-from and amount-to or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10

continues on next page

Table 118 – continued from previous page

CSV Parameter Name	Description	Value
currency	Currency type. Supported for SRC and DST type.	Necessity: Required Type: String Length: 3
country	Payer`s country. Not supported for DST.	Necessity: Required Type: String Length: 2
city	Payer`s city. Not supported for DST.	Necessity: Required Type: String Length: 128
zip-code	Payer`s zip-code. Not supported for DST.	Necessity: Required Type: String Length: 10
address1	Payer`s address. Not supported for DST.	Necessity: Required Type: String Length: 256
first-name	Payer`s first-name. Not supported for DST.	Necessity: Required Type: String Length: 128
last-name	Payer`s last-name. Not supported for DST.	Necessity: Required Type: String Length: 128
email	Payer`s email. Not supported for DST.	Necessity: Required Type: String Length: 128

continues on next page

Table 118 – continued from previous page

CSV Parameter Name	Description	Value
amount-from	If the combination of amount-from and amount-to is chosen, every charge will be of random amount between these two numbers. Supported for SRC and DST type. Required if amount or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10
amount-to	If the combination of amount-from and amount-to is chosen, every charge will be of random amount between these two numbers. Supported for SRC and DST type. Required if amount or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10
amount-sequence	If amount sequence is chosen, client will be charged amounts from this list. Example of setting up amount sequence: 10.5, 24.6, 32.0. If repeats number is higher than amount sequence number of elements, every new charge will be with last amount in amount sequence. In order for charges to begin from the first amount in the chain, current repeats number must be set as 0. Supported for SRC and DST type. Required if amount-from and amount-to or amount are not used.	Necessity: Conditional Type: Numeric Length: 10
period	Possible values are: day, week and month. In case if daily is chosen, client will be charged every day. If week - every 7 days. If monthly is chosen, client will be charged on the same date of the month, from the starting date, no matter how many days there are in a month. Interval and period can only be specified or omitted together. Not supported for DST.	Necessity: Conditional Type: String Length: 32
interval	Interval is a multiplier applied to the period. For example, if interval of 2 and period 'Daily' is selected, client will be charged once every 2 days. Interval and period can only be specified or omitted together. Not supported for DST.	Necessity: Conditional Type: Int Length: -
order_desc	Description of Recurring payment. Supported for SRC and DST type.	Necessity: Optional Type: String Length: 65K

continues on next page

Table 118 – continued from previous page

CSV Parameter Name	Description	Value
customer-ip	Payer's IP address. Supported for SRC and DST type.	Necessity: Optional Type: String Length: 45
ssn	Social security number field. Not supported for DST.	Necessity: Optional Type: String Length: 32
birthday	Payer's birthday date. Not supported for DST.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8
phone	Payer's phone number. Not supported for DST.	Necessity: Optional Type: String Length: 128
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia. Not supported for DST.	Necessity: Optional Type: String Length: 2-3
start-date	Date, when first charge is scheduled. If start date is set as a current date and type is set as auto, first charge will be made today. Supported for SRC and DST type.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8
finish-date	Date, when the Payer will be charged last time. Supported for SRC and DST type.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8

continues on next page

Table 118 – continued from previous page

CSV Parameter Name	Description	Value
max-repeats-number	Index of recurring transaction, first charge will hold the index of 0. Current repeats number increases even if a charge was unsuccessful. When current repeats number reaches max repeats number, Recurring payment goes into stop status and client is charged no more. If a charge was made automatically, no additional charges will be made(unless done manually), even if a recurring payment is stopped and rescheduled again. Supported for SRC and DST type.	Necessity: Optional Type: Int Length: -
purpose	Purpose of transaction. Not supported for DST.	Necessity: Optional Type: String Length: 128
notify_url	Notify url field. server_callback_url parameter can also be used. For more information please see Connecting Party Callbacks . Supported for SRC and DST type.	Necessity: Optional Type: String Length: 1024
server_callback_url	Connecting Party URL which will receive callback request once the transaction reaches final status. Connecting Party may use Server Callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. See callback details in Connecting Party callback parameters . Send either notify_url or server_callback_url, not both.	Necessity: Optional Type: String Length: 128

7.33.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. May be create-recurring-payment-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details. Multiple error codes may be received: 200, 403, and 500. For the 500 error code, an additional error ID will be returned.
recurring-payment-id	Recurring ID assigned to the order by Doc2.0.
status	See Status List for details.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.

7.33.5 Request Example

Step 1. Create a CSV with the provided structure:

```
"client-orderid";"rp_card_type";"payment-description";"first-name";"last-name";
↪ "address1";"city";"zip-code";
"country";"state";"phone";"email";"customer-ip";"period";"interval";"start-date";
↪ "finish-date";"max-repeats-number";"amount";
"amount-from";"amount-to";"amount-sequence";"currency";"card-printed-name";"credit-
↪ card-number";"expire-month";"expire-year"
```

Step 2. Encode CSV to base64 with the following command:

```
base64 create-recurring-payments-example.csv
```

Step 3. Assign the base64 encoded value to payload parameter and send the request:

```
POST /paynet/api/v4/create-recurring-payments/ HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/8.4.0
Accept: */*
Content-Type: application/x-www-form-urlencoded
Authorization: OAuth oauth_consumer_key="ErwinTestMerchant",oauth_signature_method=
↪ "RSA-SHA256",oauth_timestamp="1727178538",oauth_nonce="cdtNANOCATq",oauth_version="1.
↪ 0",oauth_signature="B5m%2F0Q0SBgqWI%2F
↪ %2FG3GhZw2kCJZDda7J6fLp0u6wFyBEj60PJp5u1mqBnIufzkkmrSkyZkCT6v12WIhwE%2BWTtJ7Wg
↪ %2BRx9cqXPfhk5JhQpnofsVheXpCpftoLvvtqn8p67WaS
↪ %2B6kfzGtGhdXHYVBzIXDXabnMjMC8u8EtfWsoVzTp
↪ %2BAYfZKn2H0qH7B62dfUw4PuRCTmy9svbWlhbPULkh3oJ3a1M
↪ %2BnRqXIasIrd3WY1MD9lkqUetAtW3xIwn0oMvPZ0wzk2UEbU%2BoZ2KtiA9K009
↪ %2Ffq70ylRhIDpqtz65EMBiKujFFSF42bzYsB1SmPssk09xJoWmsNR9tFZsdX2Hg17yFZTXhxV3nLSZTGDVZaVe2pPSi8va5W9
↪ %2BvX9IWdUHx1xEMdBPWdUQRfNlLgVt2N8nYrtZe96L
↪ %2BfwxarYR2Fh1LtyqwCtEXhQm5jhjpr3NQpqqdY3NSd75NUP7zLDQwXCknMQps6uSdng9Uv
↪ %2B7IFx3oNUfjVMxp4CGcmEtSxH7Ykx8dX5srNOKLDJT5tyExLLjXk%3D"
Content-Length: 754
```

(continues on next page)

(continued from previous page)

Connection: keep-alive

payload=Y2xpZW50LW9yZGVyaWQ7cnBfY2FyZF90eXB1029yZGVyX2Rlc2M7Zmlyc3QtbmFtZTtsYXN0LW5hbWU7YWJkcmVzc2E7
↪%3D

7.33.6 Success Response Example

Note:

The successful response has empty body and HTTP code 200.

```
HTTP/1.1 200
Server: server
Date: Tue, 24 Sep 2024 09:44:01 GMT
Content-Length: 0
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000
```

7.33.7 Fail Response Example

Note:

The unsuccessful response has empty body and HTTP code 403.

```
HTTP/1.1 403
Server: server
Date: Wed, 25 Sep 2024 08:45:42 GMT
Content-Length: 0
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
```


7.33.8 Request Builder

7.34 /api/v4/process-recurring-payments

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.34.1 Introduction

If request is accepted with no errors, the Payment Gateway finds the recurring payment profiles for each provided recurring-payment-id and initiates new recurring transactions using the payment data saved in each profile.

Process recurring payment Multiple is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.34.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/process-recurring-payments/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/process-recurring-payments/ENDPOINTID

7.34.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Below is a description of each parameter that can be included in the CSV and added to payload parameter which will be used in the request.

CSV Parameter Name	Description	Value
client-orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
recurring-payment-id	Recurring ID assigned to the order by QA.	Necessity: Required Type: String Length: 10
amount	Amount of currency must be the same as currency on the project assigned. Upon reaching finish date, Recurring payment will go into stop status. Supported for SRC and DST type. Required if amount-from and amount-to or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10

7.34.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. Example: process-recurring-payment-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details. Multiple error codes may be received: 200, 403, and 500. For the 500 error code, an additional error ID will be returned.
status	See Status List for details
serial-number	Unique number assigned by Doc2.0 server to particular request from the Merchant

7.34.5 Request Example

Step 1. Create a CSV with the provided structure:

```
"client-orderid";"recurring-payment-id";"amount"
```

Step 2. Encode CSV to base64 with the following command:

```
base64 process-recurring-payments-example.csv
```

Step 3. Assign the base64 encoded value to payload parameter and send the request:

```
POST /paynet/api/v4/process-recurring-payments/ HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/8.4.0
Accept: */*
Content-Type: application/x-www-form-urlencoded
Authorization: 0Auth oauth_consumer_key="ErwinTestMerchant",oauth_signature_method=
  ↳ "RSA-SHA256",oauth_timestamp="1727177782",oauth_nonce="Js4dwXkF8eI",oauth_version="1.
  ↳ 0",oauth_signature="EGLiLoJJsZzZ4LgTdexkExuNCeyJEkSYuQwsrLMTvucJiIoG2rXL
  ↳ %2FaIkwnXwY6ncXS%2BopSYszmvAE1KnR3l6YxfyPzvGw07LkrCE9r80yEfYuG0jwdB7fqvCyYtPqylS7MJ0
  ↳ %2BJM1EIowuWkiwURLqU3mG4J00t1Kl7CRrsjT9nwKoj8mRrljb1St3rAh8bJ7SsdKKQwir1Va6t0cuVMxjsUHJqCtztCXAPj
  ↳ %2BRz4UKrjbQnwNvJ1Mf0nTyeRXTAJgGg7XwctumcG5W2vfuKICwEUQ9S95jxXIci
  ↳ %2FmHuA8gkV7DmfolT1CnxDYzhJwvspvg9BP%2FFVx0GjUTEKDwNhCX5BB3B6EUkyEFI79V8tAtUdUy
  ↳ %2FzRmChPA%2FYvyBky8egkVXoKgwoUHiTQ5ITEhbzCTLjuVm9J
  ↳ %2FEwqVHQkwPbn6T5aTcvQBoMkDtZ70qm7J73wPKuN%2B7S%2Fi0SBc6gwGyu6G%2FfXYu
  ↳ %2BvDY8Ihtd6HRUC59sUYQ6D3tVbh8CE4W9tF1UiXCAoPehdBga%2BR6WUq%2FnD%2FRFrBDZ2Y%2Bz
  ↳ %2FMghjQHBNZu%2B5z4JjNn7QusHp01ZwvrI1kSlK79toV0iwVX
  ↳ %2FcP4H5M20s5EPjvSrcc2tnqQSUU0LFGyRF0bQgs7kmBxUjEMXbVh9QXlIBdSvy4S8PkFnqYjVLh2aNgTpw
  ↳ %2BgB8RrZlMY%3D"
Content-Length: 888
Connection: keep-alive

payload=cmVjdXJyaW5nLXBheW1lbnQtawQ7Y2xpZW50LW9yZGVyaWQ7cGF5bWVudC1kZXNjcmlwdGlvbjtmaXJzdC1uYW1lO2xh
```

7.34.6 Success Response Example

Note:

The successful response has empty body and HTTP code 200.

```
HTTP/1.1 200
Server: server
Date: Tue, 24 Sep 2024 09:47:01 GMT
Content-Length: 0
```

(continues on next page)

(continued from previous page)

```
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000
```

7.34.7 Fail Response Example

Note:

The unsuccessful response has empty body and HTTP code 403.

```
HTTP/1.1 403
Server: server
Date: Wed, 25 Sep 2024 08:46:12 GMT
Content-Length: 0
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
```

7.34.8 Request Builder

7.35 /api/v4/update-recurring-payments

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.35.1 Introduction

If request is accepted with no errors, the Payment Gateway finds the recurring payment profile for each provided recurring-payment-id and updates these profiles. The recurring transactions will be processed according to the updated recurring payment profiles using the payment data saved in each profile.

Update recurring payment Multiple is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.35.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/update-recurring-payments/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/update-recurring-payments/ENDPOINTID

7.35.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Below is a description of each parameter that can be included in the CSV and added to payload parameter which will be used in the request.

Any changes to recurring payment profile made with Update command can be viewed only via UI.

Note:

The RPI details screen (recurring payment profile) contains information about linked cardholder and customer data, recurring schedule and processed transactions with this RPI. This screen also contains change history for this RPI. The change history is currently available for source cards (SRC) only.

Type native on UI means that The recurring payment setup and the actual charges are in the same acquirer.

If the recurrent has switched to Stopped, you can update its schedule using the start-date and finish-date parameters. However, it can only be resumed via the UI by clicking on Resume.

To stop the automatic recurring schedule, use finish-date in the past in [Request Parameters](#).

CSV Parameter Name	Description	Value
client-orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
recurring-payment-id	Recurring ID assigned to the order by QA.	Necessity: Required Type: String Length: 128
credit-card-number	Payer`s credit card number.	Necessity: Required Type: Numeric Length: 19
card-printed-name	Payer`s card printed name.	Necessity: Required Type: String Length: 128
expire-year	Payer`s card expire year.	Necessity: Required Type: Numeric Length: 4
expire-month	Payer`s card expire month.	Necessity: Required Type: Numeric Length: 2
amount	Amount of currency must be the same as currency on the project assigned. Upon reaching finish date, Recurring payment will go into stop status. Supported for SRC and DST type. Required if amount-from and amount-to or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10
amount-from	If the combination of amount-from and amount-to is chosen, every charge will be of random amount between these two numbers. Supported for SRC and DST type. Required if amount or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10

continues on next page

Table 124 – continued from previous page

CSV Parameter Name	Description	Value
amount-to	If the combination of amount-from and amount-to is chosen, every charge will be of random amount between these two numbers. Supported for SRC and DST type. Required if amount or amount-sequence are not used.	Necessity: Conditional Type: Numeric Length: 10
amount-sequence	If amount sequence is chosen, client will be charged amounts from this list. Example of setting up amount sequence: 10.5, 24.6, 32.0. If repeats number is higher than amount sequence number of elements, every new charge will be with last amount in amount sequence. In order for charges to begin from the first amount in the chain, current repeats number must be set as 0. Supported for SRC and DST type. Required if amount-from and amount-to or amount are not used.	Necessity: Conditional Type: Numeric Length: 10
period	Possible values are: day, week and month. In case if daily is chosen, client will be charged every day. If week - every 7 days. If monthly is chosen, client will be charged on the same date of the month, from the starting date, no matter how many days there are in a month. Interval and period can only be specified or omitted together. Not supported for DST.	Necessity: Conditional Type: String Length: 32
interval	Interval is a multiplier applied to the period. For example, if interval of 2 and period 'Daily' is selected, client will be charged once every 2 days. Interval and period can only be specified or omitted together. Not supported for DST.	Necessity: Conditional Type: Int Length: -
country	Payer`s country.	Necessity: Optional Type: String Length: 2
city	Payer`s city.	Necessity: Optional Type: String Length: 128

continues on next page

Table 124 – continued from previous page

CSV Parameter Name	Description	Value
address1	Payer`s address.	Necessity: Optional Type: String Length: 256
first-name	Payer`s first-name.	Necessity: Optional Type: String Length: 128
last-name	Payer`s last-name.	Necessity: Optional Type: String Length: 128
customer-ip	Payer`s IP address. Supported for SRC and DST type.	Necessity: Optional Type: String Length: 45
order_desc	Description of Recurring payment.	Necessity: Optional Type: String Length: 65K
zip-code	Payer`s zip-code.	Necessity: Optional Type: String Length: 10
birthday	Payer`s birthday date.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8

continues on next page

Table 124 – continued from previous page

CSV Parameter Name	Description	Value
email	Payer's email.	Necessity: Optional Type: String Length: 128
ssn	Social security number field.	Necessity: Optional Type: String Length: 32
phone	Payer's full international phone number, including country code. Not supported for DST.	Necessity: Optional Type: String Length: 128
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia. Not supported for DST.	Necessity: Optional Type: String Length: 2-3
start-date	Date, when first charge is scheduled. If start date is set as a current date and type is set as auto, first charge will be made today.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8
finish-date	Date, when the Payer will be charged last time.	Necessity: Optional Type: 8/Numeric, DD.MM.YYYY Length: 8

continues on next page

Table 124 – continued from previous page

CSV Parameter Name	Description	Value
max-repeats-number	Index of recurring transaction, first charge will hold the index of 0. Current repeats number increases even if a charge was unsuccessful. When current repeats number reaches max repeats number, Recurring payment goes into stop status and client is charged no more. If a charge was made automatically, no additional charges will be made(unless done manually), even if a recurring payment is stopped and rescheduled again.	Necessity: Optional Type: Int Length: -
purpose	Purpose of transaction.	Necessity: Optional Type: String Length: 128
notify_url	Notify url field. server_callback_url parameter can also be used. For more information please see Connecting Party Callbacks .	Necessity: Optional Type: String Length: 1024
server_callback_url	Connecting Party URL which will receive callback request once the transaction reaches final status. Connecting Party may use Server Callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. See callback details in Connecting Party callback parameters . Send either notify_url or server_callback_url, not both.	Necessity: Optional Type: String Length: 128

7.35.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response Parameters	Description
type	The type of response. Example: update-recurring-payment-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details. Multiple error codes may be received: 200, 403, and 500. For the 500 error code, an additional error ID will be returned.
status	See Status List for details
serial-number	Unique number assigned by Doc2.0 server to particular request from the Merchant

7.35.5 Request Example

Step 1. Create a CSV with the provided structure:

```
"client-orderid";"recurring-payment-id";"payment-description";"first-name";"last-name"
↪";"address1";"city";"zip-code";
"country";"state";"phone";"email";"customer-ip";"period";"interval";"start-date";
↪"finish-date";"max-repeats-number";"amount";
"amount-from";"amount-to";"amount-sequence";"currency";"card-printed-name";"credit-
↪card-number";"expire-month";"expire-year"
```

Step 2. Encode CSV to base64 with the following command:

```
base64 update-recurring-payments-example.csv
```

Step 3. Assign the base64 encoded value to payload parameter and send the request:

```
POST /paynet/api/v4/update-recurring-payments/ HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/8.4.0
Accept: */*
Content-Type: application/x-www-form-urlencoded
Authorization: OAuth oauth_consumer_key="ErwinTestMerchant",oauth_signature_method=
↪"RSA-SHA256",oauth_timestamp="1727177782",oauth_nonce="Js4dwXkF8eI",oauth_version="1.
↪0",oauth_signature="VwdA7BQ68v%2BmpB0N%2BU0xK
↪%2BxYk355i1QzeUPGGGFwDEBn7Y8v1xpSolGQ45HehGmJHNjXHc7A1mP3x7V7r2pQju1LpDEvAb2MHNLSCHiCEjI95sCtrotE
↪%2Fs5%2FmQmIJ8te%2FGFCR1uK%2BzMPG8bpHqn%2B5EIEsuLPq6TS0jd0N3RvnV
↪%2BHdmjH5cxTcmtRrcY0u6VIpvkBUlqpKuTVJXLvbpXReXvgPMDow78Q53DLRQhi6G7Y%2FVshKpKC
↪%2FSRThhe8L33tFckX6KaEbJ3XIMEmp70%2F%2FQdLfwQBn4ldSp8K8lpkgZks4CZbAjDY
↪%2BQpSfwdc1s8kJf17Ymk1R69aGBmjzJrw00tV4dzY4DE6XVqSTUR8X%2FCa0XMrtD46ichsFoRvFtIeyV
↪%2FIud%2F%2FLSb8XDqk%2BafTSLazBokmT8Qe1FMf0UMgUYBLCL0B4066Ys8kH4Z6guC2MXarwu
↪%2BDlfuelrcAHevS68hewrMb
↪%2FjppSJWAbQiB0ABeW6s1Rb4dVbJnZribkKhEwrXmnT5drstYvukC2UCoUblE0gVkfHdHk5E30qT4wMxqXojLYr5Il7m1GkHV
↪%2FUHirjybdWeynTCSaGe%2Bv3JtEbkydWXannkdtTvK0xkbT6LurBiNWy1FuSTmKod9ibqyNiEv6j
↪%2Be14BBGXs3xc5A%3D"
Content-Length: 888
```

(continues on next page)

(continued from previous page)

Connection: keep-alive

payload=cmVjdXJyaW5nLXBheW1lbnQtaWQ7dHlwZTtjbGllbnQtb3JkZXJpZDtwYXltZW50LWRlc2NyaXB0aW9u02ZpcnN0LW5h

7.35.6 Success Response Example

Note:

The successful response has empty body and HTTP code 200.

```
HTTP/1.1 200
Server: server
Date: Tue, 24 Sep 2024 11:40:45 GMT
Content-Length: 0
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
```

7.35.7 Fail Response Example

Note:

The unsuccessful response has empty body and HTTP code 403.

```
HTTP/1.1 403
Server: server
Date: Wed, 25 Sep 2024 08:47:54 GMT
Content-Length: 0
Connection: keep-alive
Keep-Alive: timeout=60
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
```

7.35.8 Request Builder

7.36 /api/v4/status

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)

- [Response Parameters](#)
- [PaReqForm Status Response Parameters](#)
- [CReqForm Status Response Parameters](#)
- [MethodUrlFrame Status Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.36.1 Introduction

To make an order status request one have to send an HTTPS POST request to the [URLs](#) and the [parameters](#) specified below. Use [RSA-SHA256](#) for authentication. See [Statuses](#).

7.36.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/status/ENDPOINTID	https://doc2.com/paynet/api/v4/status/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v4/status/group/ENDPOINTID	https://doc2.com/paynet/api/v4/status/group/ENDPOINTID

7.36.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Necessity
login	Connecting Party login name.	Mandatory
client_orderid	Unique order identifier assigned by Connecting Party.	Mandatory
orderid	Order id assigned to the order by Doc2.0.	Conditional

continues on next page

Table 127 – continued from previous page

Parameter Name	Description	Necessity
by-request-sn	Serial number assigned to the specific request by doc2. If this field exist in status request, status response return for this specific request. Include this parameter to get the status request with the particular transaction stage (can be used in specific cases). To get the latest transaction status, don't include this parameter in status request.	Optional

In most common cases, the best option is to include both `client_orderid` and `orderid` parameters to status request. Order status can be requested with only `client_orderid` if it's unique to Connecting Party and `orderid` is not received. If `orderid` is not received in response, but this response contains an error, see the received error message to get the information why transaction was not created in the system.

7.36.4 Response Parameters

The same API command for status request is used in multiple Use-Cases, therefore some of the mentioned response parameters might not be present for specific case. Below is the full list of possible parameters.

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

* - these parameters are not defined by default. Please contact tech support to include these fields in callback.

Status Response Parameter	Description
type	The type of response. May be status-response.
status	See Status List for details.
amount	Actual transaction amount. This value can be changed during the transaction flow.
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar EUR for European Euro.
paynet-order-id	Order id assigned to the order by gate.doc2.com.
merchant-order-id	Connecting Party order id.
phone	Payer's full international phone number, including country code.

continues on next page

Table 128 – continued from previous page

Status Response Parameter	Description
html	HTML code of 3DS authorization form, encoded in application/x-www-form-urlencoded MIME format. Connecting Party must decode this parameter before showing the form to the Payer. gate.doc2.com System returns the following response parameters when it gets 3DS authorization form from the Issuer Bank. It contains auth form HTML code which must be passed through without any changes to the client's browser. This parameter exists and has value only when the redirection HTML is already available. For non-3DS this never happens. For 3DS HTML has value after some short time after the processing has been started.
redirect-to	For 3DS authorization Connecting Party can redirect the Payer to URL provided in this parameter instead of rendering the page provided in html parameter. The redirect-to parameter is returned only if the html parameter is returned. Connecting Party should use GET HTTP method to redirect. This parameter must be used to work with 3DS 2.0.
serial-number	Unique number assigned by gate.doc2.com server to particular request from the Connecting Party.
last-four-digits	Last four digits of Payer bank card number.
dest-last-four-digits	Last four digits of customer credit card number. Relevant only for transfer transactions.
bin	Bank BIN of Payer bank card number.
card-type	Type of Payer bank card (VISA, MASTERCARD, etc).
gate-partial-reversal	Processing gate support partial reversal (enabled or disabled).
gate-partial-capture	Processing gate support partial capture (enabled or disabled).
transaction-type	Transaction type (sale, reversal, capture, preauth).
processor-rrn	Bank Receiver Registration Number.
processor-tx-id	Acquirer transaction identifier.
receipt-id	Electronic link to receipt https://gate.doc2.com/paynet/view-receipt/ENDPOINTID/receipt-id/ .
name	Payer's name.
card-ref-id	Card reference ID used in subsequent recurrent payments. Relevant only if card-ref-id was created for initial transaction.
cardholder-name	Cardholder's name.
card-exp-month	Bank card expiration month.
card-exp-year	Bank card expiration year.
card-hash-id	Unique card identifier to use for loyalty programs or fraud checks.
card-country-alpha-three-code	Three letter country code of source card issuer. See Country and State Codes for details.
destination-card-country-alpha-three-code	Three letter country code of destination card issuer. See Country and State Codes for details.
dest-bin	Bank BIN of customer credit card number.
dest-card-type	Type of customer credit card (VISA, MASTERCARD, etc).
dest-bank-name	Bank name by customer card BIN.
destination-hash-id	Unique card identifier to use for loyalty programs or fraud checks. Relevant only for transfer transactions.

continues on next page

Table 128 – continued from previous page

Status Response Parameter	Description
destination-card-hash-id	Unique card identifier to use for loyalty programs or fraud checks.
first-name	Payer's first name.
last-name	Payer's last name.
email	Payer's e-mail.
country *	Payer's country (two-letter country code). Please see Country and State Codes for a list of valid country codes.
state *	Payer's state . Please see Country and State Codes for a list of valid state codes. Mandatory for USA, Canada and Australia.
city *	Payer's city.
zip_code *	Payer's ZIP code.
address1 *	Payer's address line 1.
purpose	Destination to where the payment goes. It is useful for Connecting Party who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.
bank-name	Bank name by Payer card BIN.
terminal-id	Acquirer terminal identifier to show in receipt.
paynet-processing-date	Acquirer transaction processing date.
approval-code	Bank approval code.
order-stage	The current stage of the transaction processing. See Order Stage for details.
total-reversal-amount	Total amount of processed reversals. Relevant only for reversal transactions.
reversal-amount	The amount of the last processed reversal. Relevant only for reversal transactions.
auth-response-code	Response code used in Iso8583 protocol. Only returned in specific cases.
acquirer-processing-date	Acquirer transaction processing date.
processor-auth-credit-code	Approval credit code. Only returned in specific cases.
processor-credit-rrn	Retrieval Reference Number for credit transaction.
processor-credit-arn	Acquirer card reference number for credit transaction.
processor-debit-arn	Acquirer card reference number for debit transaction.
loyalty-balance	The current bonuses balance of the loyalty program for current operation. if available.
loyalty-message	The message from the loyalty program. if available.
loyalty-bonus	The bonus value of the loyalty program for current operation if available.
loyalty-program	The name of the loyalty program for current operation if available.
descriptor	Bank identifier of the payment recipient.
original-gate-descriptor	Descriptor, which is set on gate level in the system.
error-message	If status is declined, error, filtered this parameter contains the reason for decline.
error-code	The error code is case status is declined, error, filtered.

continues on next page

Table 128 – continued from previous page

Status Response Parameter	Description
by-request-sn	Serial number assigned to the specific request by gate.doc2.com. If this field exist in status request, status response return for this specific request.
verified-3d-status	See 3D Secure Status List for details.
verified-rsc-status	Returned if Random Sum Check was performed. See Alternative cardholder authentication
eci	Electronic Commerce Indicator (Visa).
ips-src-payment-product-code	Code for card set by multinational financial service (Visa/Mastercard).
ips-src-payment-product-name	Decrypted code for card set by multinational financial service (Visa/Mastercard).
ips-src-payment-type-code	Type of card code set by multinational financial service (Visa/Mastercard).
ips-src-payment-type-name	Decrypted code for type of card set by multinational financial service (Visa/Mastercard).
merchantdata	If provided in initial request, merchant_data parameter and its value will be included in status response.
initial-amount	Amount, set in initiating transaction, without any fees or commissions. This value can't change during the transaction flow.
seller-commission	Total commission for processed transaction. This is optional parameter. Please contact your manager in Doc2.0, if you would like to receive it.
acquirer-commission	Acquirer commission for processed transaction. This is optional parameter. Please contact your manager in Doc2.0, if you would like to receive it.
motivational-message	This is an optional message which contains extended information about the reason for the declined transaction.
transaction-date	Date of final status assignment for transaction.
orig-amount	Contains the original request amount if it was converted on auxiliary endpoint in Parallel form integration. Relevant only for Payment Cashier transactions.
orig-currency	Contains the original request currency if it was converted on auxiliary endpoint in Parallel form integration. Relevant only for Payment Cashier transactions.

QR Code Status Response Parameters:

Status Response Parameter	Description
qr-code	QR code in base 64 format.
qr-code-payload-type	QR Code type = SBP.
qr-code-payload-value	Link to the QR code = https://qr.nspk.ru/BS***** (only for H2H integration).

7.36.5 PaReqForm Status Response Parameters

Name	Description
tds-pareq-form-pareq	ACS 3DS PaReq data, which received by the Connecting Party.
tds-pareq-form-acs-url	ACS URL to redirect the Payer to 3DS 1.0.2 Authentication Flow.

7.36.6 CReqForm Status Response Parameters

Name	Description
tds-creq-form-creq	A CReq message initiates Cardholder interaction in a Challenge Flow and is used to carry authentication data from the Cardholder. It is formed by the 3DS Server and is posted through the Cardholder's browser by Connecting Party to the ACS URL.
tds-creq-form-acs-url	ACS URL to redirect the Payer for Challenge Flow.

7.36.7 MethodUrlFrame Status Response Parameters

Name	Description
tds-method-url-frame-3ds-server-trans-id	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.
tds-method-url-frame-3ds-method-url	3DS Method URL used in iframe form which is provided to Payer browser by Connecting Party.

Rules to form the HTML form.

threeDSMethodData (threeDSMethodNotificationURL + threeDSSTransID).

7.36.8 Request Example

```
POST /paynet/api/v4/status/46750 HTTP/1.1
User-Agent: curl/7.88.1
Accept: */*
Authorization: 0Auth oauth_consumer_key="Test_Merchant", oauth_nonce=
  ↳ "BK0z6eH0s6sDJLLPAJhbDHAaXCY9xxNv", oauth_signature="eQXkV
  ↳ %2BJdiqJlyRqNaEzIKmYa3FZzUjdcMR6LXSfRn9t0YKUNPxI3UKU%2F%2FGsPpofXL%2B2RBjGh3Gqv
  ↳ %2BZBoKaVK0gNKwNNwpA4IOaskV71uuMbZCp0gPEbS%2BvAWLD8vzqpcgZ%2Bd5DRNMfimyXkWVwbsMUYj8N
  ↳ %2BSpXl4YnGIO0nXz9Q0Ppxetie3EG9NrN7CNu7NdovVjmsfYqPDRv90hLo4tSQTD9C6bWvW2kmEvZsb2d1KANsGUW6rXyjkI
  ↳ %2FNk%2FVjNWgav2uw9J9I%2FiTqLLcKZ1pTWj1W0MwXhfoMCP10X0A0e72CQHx0DJL
  ↳ %2BFt01jm0XLvLdEkUZTFzsC6DGfHSDdcsjXquc9gxFKVr3d8e15by3566UI4pXKef%2Fe
  ↳ %2B3Ytvlrj7IUhIcNyA%2BVXp%2FwivxgwYu2xpQJMs6wlvw6Lz3N2wcFRqLs5ZEbdZ1
  ↳ %2F29pox8XW0ae8yZ2z2PClPzmJI0Dc0r0GETwyz5Byyew0m33XA67UbPN6rwbdlVL2gwMqWwkn7KDYp7
  ↳ %2BifP%2B2BdbYXnw2LeJcuYDYAIDHa%2Bi0P09ZVT0BpeL0x%2FobSF2y%2FsheVgo00
```

(continues on next page)

(continued from previous page)

```
↪%2FRWtUEEX0Nvd0n7hEdnJ7mMYNivNitbfQ4SryQ2o8CdUDk9RgEaR7pn7ybTi4rQEhDqWF8sFMtaKhFn9o
↪%3D", oauth_signature_method="RSA-SHA256", oauth_timestamp="1734327207", oauth_
↪version="1.0"
Content-Length: 58
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=1&login=Test_Merchant&order_id=7364742
```

7.36.9 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Mon, 16 Dec 2024 05:38:33 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 1469

type=status-response
&serial-number=000000000-0000-0000-0000-000002f38234
&merchant-order-id=123456
&processor-tx-id=PNTEST-7364748
&paynet-order-id=7364748
&status=approved
&amount=10.42
&currency=USD
&descriptor=Test
&original-gate-descriptor=Test
&transaction-type=transfer
&receipt-id=2f885ae0-5220-3549-8eb5-622f4201f882
&name=John+Doe
&cardholder-name=John+Doe
&card-exp-month=12
&card-exp-year=2099
&processor-rrn=0435171505391
&approval-code=466875
&order-stage=transfer_approved
&last-four-digits=5721
&bin=421070
&card-type=VISA
&bank-name=DEMIRBANK+OJSC
&dest-bank-name=JPMORGAN+CHASE+BANK+N.A.
&dest-bin=423261
&dest-last-four-digits=1636
&dest-card-type=VISA
&auth-response-code=00
&paynet-processing-date=2024-12-16+08%3A37%3A25+MSK
&acquirer-processing-date=2024-12-16+08%3A37%3A25+MSK
```

(continues on next page)

(continued from previous page)

```
&processor-auth-credit-code=311830
&card-hash-id=2511341
&destination-card-hash-id=2511340
&card-country-alpha-three-code=AZE
&destination-card-country-alpha-three-code=USA
&verified-3d-status=NOT_AUTHENTICATED
&processor-credit-rrn=0435147814453
&processor-credit-arn=899834666
&processor-debit-arn=668539305
&ips-src-payment-product-code=UNK
&ips-src-payment-product-name=Unknown
&ips-src-payment-type-code=Credit
&ips-src-payment-type-name=VISA+Credit
&ips-dst-payment-product-code=UNK
&ips-dst-payment-product-name=Unknown
&ips-dst-payment-type-code=Prepaid
&ips-dst-payment-type-name=VISA+Prepaid
&initial-amount=10.42
&transaction-date=2024-12-16+08%3A37%3A34+MSK
```

7.36.10 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Mon, 16 Dec 2024 05:33:57 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 164

type=status-response
&serial-number=000000000-0000-0000-0000-000002f38231
&merchant-order-id=1
&status=error
&error-message=AMBIGUOUS_CLIENT_ORDER_ID
&error-code=124
```

7.36.11 Postman Collection

7.36.12 Request Builder

Enter your private key in PKCS#1 container to use debug. See [RSA-SHA256](#) for details.

7.37 /api/v4/payout

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Test Scenario](#)
- [Postman Collection](#)
- [Request Builder](#)

7.37.1 Introduction

Payout is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.37.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/payout/group/ENDPOINTID	https://paynet.doc2.com/paynet/api/v4/payout/group/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v4/payout/group/ENDPOINTID	https://paynet.doc2.com/paynet/api/v4/payout/group/ENDPOINTID

7.37.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
client_orderid	Connecting Party's order identifier.	Necessity: Required Type: String Length: 128
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Brief order description.	Necessity: Optional Type: String Length: 64
ipaddress	Receiver's IP address (IPv4 or IPv6).	Necessity: Conditional Type: String Length: 7-45
purpose	Payout purpose.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 134 – continued from previous page

Parameter Name	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 128
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 128

continues on next page

Table 134 – continued from previous page

Parameter Name	Description	Value
redirect_url	URL, where the Receiver is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status. Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if you have no need to return Receiver anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 128
redirect_success_url	URL, where the Receiver is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Receiver anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 134 – continued from previous page

Parameter Name	Description	Value
redirect_fail_url	URL, where the Receiver is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command . Pass https://doc2.codetime.net if there is no need to redirect Receiver anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
credit_card_number	Customer's credit card number. Note: For the scenario of payment to a card inside the system, this card will be considered as a source, and all processing limits, lists and fraud scoring will be applied to it as a source card.	Necessity: Conditional Type: Numeric Length: 20
card_printed_name	Customer's full name, as printed on the card.	Necessity: Conditional Type: String Length: 128
expire_month	Credit card expiration month.	Necessity: Conditional Type: Numeric Length: 2
expire_year	Credit card expiration year.	Necessity: Conditional Type: Numeric Length: 4
cvv2	Receiver's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card.	Necessity: Conditional Type: Numeric Length: 3-4

continues on next page

Table 134 – continued from previous page

Parameter Name	Description	Value
account_number	Account Number.	Necessity: Conditional Type: String Length: 32
account_name	Bank account.	Necessity: Optional Type: String Length: 128
ewallet_type	Type of e-wallet.	Necessity: Conditional Type: String Length: 64
ewallet_wallet	E-wallet ID.	Necessity: Conditional Type: String Length: 128
crypto_wallet_address	Address of crypto wallet.	Necessity: Conditional Type: String Length: 64
bank_name	Bank Name.	Necessity: Conditional Type: String Length: 255
bank_branch	Bank Branch Name.	Necessity: Conditional Type: String Length: 255
bank_code	Bank code.	Necessity: Conditional Type: String Length: 32

continues on next page

Table 134 – continued from previous page

Parameter Name	Description	Value
bank_city	Bank city.	Necessity: Conditional Type: String Length: 128
bank_address1	Bank address.	Necessity: Conditional Type: String Length: 255
bank_zip_code	Bank postal ZIP code.	Necessity: Conditional Type: String Length: 255
bank_province	Bank province.	Necessity: Conditional Type: String Length: 255
bank_area	Bank area.	Necessity: Conditional Type: String Length: 255
routing_number	Routing number used to identify specific bank branches in China.	Necessity: Conditional Type: String Length: 16
legal_person_name	Name on the legal document.	Necessity: Conditional Type: String Length: 128
legal_person_document_number	Number of legal document.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 134 – continued from previous page

Parameter Name	Description	Value
receiver_first_name	Receiver first name, also can be sent as first_name.	Necessity: Conditional Type: String Length: 128
receiver_last_name	Receiver last name, also can be sent as last_name.	Necessity: Conditional Type: String Length: 128
receiver_middle_name	Receiver middle name, also can be sent as middle_name.	Necessity: Conditional Type: String Length: 128
receiver_birthday	Receiver birthday, also can be sent as birthday.	Necessity: Conditional Type: Numeric Length: 30
receiver_country_code	Receiver country code, also can be sent as country.	Necessity: Conditional Type: String Length: 3
receiver_state	Receiver state, should be provided for countries that have states (USA, Canada, Australia), also can be sent as state.	Necessity: Conditional Type: String Length: 4
receiver_city	Receiver city, also can be sent as city.	Necessity: Conditional Type: String Length: 128
receiver_zip_code	Receiver zip code, also can be sent as zip_code.	Necessity: Conditional Type: Numeric Length: 32

continues on next page

Table 134 – continued from previous page

Parameter Name	Description	Value
receiver_address1	Receiver address, also can be sent as address1.	Necessity: Conditional Type: String Length: 256
receiver_phone	Receiver phone, also can be sent as phone.	Necessity: Conditional Type: Numeric Length: 128
receiver_email	Receiver E-mail, also can be sent as email.	Necessity: Conditional Type: String Length: 128
receiver_identity_document_id	Receiver identity document identifier, also can be sent as identity_document_id.	Necessity: Conditional Type: String Length: 128
receiver_identity_document_number	Receiver identity document number, also can be sent as identity_document_number.	Necessity: Conditional Type: String Length: 128
merchant_data	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
bank_bic	BIC of the receiver's bank.	Necessity: Optional Type: String Length: 128
receiver_inn	Receiver's INN.	Necessity: Optional Type: String Length: 128

continues on next page

Table 134 – continued from previous page

Parameter Name	Description	Value
customer_level	Customer level in CMS system	Necessity: Optional Type: Varchar Length: 32
customer_id	Customer ID in CMS system. Required if transaction goes via CMS (Payment Gateway mode)	Necessity: Optional Type: Int Length: 10
merchant_customer_identifier	Merchant Customer ID in CMS system. Required if transaction goes via CMS (CRM mode)	Necessity: Optional Type: Varchar Length: 64
card_recurring_payment_id	Payer's tokenized cardholder's data ID, referred as Recurring Payment ID (RPI). Send either card_recurring_payment_id or combination of credit_card_number, card_printed_name, expire_month and expire_year, not all. To create card_recurring_payment_id see /api/v4/create-card-ref . Note: For the scenario of payment to a card inside the system, this card will be considered as a source, and all processing limits, lists and fraud scoring will be applied to it as a source card.	Necessity: Conditional Type: Long
recurring-payment-id	Recurring Payment ID can be sent instead of cardholder data. Customer Data can be updated via /api/v4/update-recurring-payment/ . Recurring Payment ID creation is initiated through HTTPS POST request by using URLs and the parameters specified below. Use OAuth RSA-SHA256 for authentication.	Necessity: Conditional Type: Long

* Ask Support Manager if Conditional fields are Required for integration.

7.37.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Payout Response Parameters	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.37.5 Request Example

```
POST /paynet/api/v4/payout/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
↳ "b5E31Tw6SVjauE29u0f2j0LnuUSXmVdE", oauth_signature=
↳ "WrW79JHNUVwDRhCGWQYgaN6xmJXpQxy8XSNyCOL6b2Wyf7V5BWGMe2TZa1bjC9Ze00Q3FcQxeGGHv0
↳ %2F7hsMAsJNQEET321VNsbdwao2Ep
↳ %2Bp7eoYiGYrVveSrSWldiCrBf3AJYJZM0PTJ67Sl8XyeTVBHT4kpC5qBu3xDQ3aFfKnMRTmn9fiVsYsYu3DQrsHM1K9uAoltG
↳ %2FvkiP%2BZ44i8ghIU61NWSrJ4Ky69JiZ%2FoaVvmUTEaanc%2F%2B%2BQT6jBwWy%2Bb
↳ %2FTLURxtSakeNfcjn1JVwRf4aCX2fhyG5ozH
↳ %2BjiXF6eRb83WVqBUAwykSq35pLU3Vmua3pKMKAJK1ZRDZdjGrT50KJg5tBniC4JFzdQqjQf
↳ %2FhFnDYodfIK3S2qZo%2FD3Bmlya46iEcK6SAQdNBBQue3E5Qi8FEHYrY1o7K8wDyzT1QzqqHF
↳ %2BQdmXcElSGu9ge0Y655%2BbGtXhnsUWnKE00NGqErvAwzm7yUg0e5QWHVf505aE7pr5K4z
↳ %2Fzj7AvkuD7R1savqam%2BnnuSfq1E%2BnnnN7mTcC0g18Sr38vdTshcGq99YW3xWKc
↳ %2FpuooZYdYa5A6u46o%2BREZSTCD2XexcV49%2F9eVn3xdoTXyq4NISJSY8U7ThKnr0g%3D", oauth_
↳ signature_method="RSA-SHA256", oauth_timestamp="1677831012", oauth_version="1.0"
Content-Length: 150
Content-Type: application/x-www-form-urlencoded
Connection: close

account_name=1234
&account_number=1234
&amount=10.42
&client_orderid=1
&currency=USD
&routing_number=15
&server_callback_url=https%3A%2F%2Fhttpstat.us%2F200
```

7.37.6 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Fri, 03 Mar 2023 08:10:34 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=async-response
&serial-number=00000000-0000-0000-0000-000002e2c322
&merchant-order-id=1
&paynet-order-id=6982864
&end-point-id=39915

```

7.37.7 Fail Response Example

```

HTTP/1.1 403 Forbidden
Server: server
Date: Thu, 25 Aug 2022 06:50:16 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>403</title>

  <style type="text/css">
body {
font-family: Arial, sans-serif;
font-size: 130%;
background-color: #eee;
}

p {
margin: 10em auto 0;
width: 500px;
border: 1px solid gray;
text-align: center;
vertical-align: middle;
padding: 40px 20px;
background-color: #fff;

```

(continues on next page)

(continued from previous page)

```
-webkit-border-radius: 20px;  
-moz-border-radius: 20px;  
border-radius: 20px;  
}  
</style>  
</head>  
  
<body>  
<p>Access is denied</p>  
</body>  
</html>
```

7.37.8 Test Scenario

Different Payout transaction statuses can be received on sandbox depending on the account_number value passed in Payout request.

Testing account_number values:

- account_number = 1234567890 to get APPROVED
- account_number = 0987654321 to get DECLINED
- account_number = 1987654321 to get PROCESSOR_INTERNAL_ERROR

7.37.9 Postman Collection

7.37.10 Request Builder

Insert PKCS#1 PEM private key for sandbox environment in the field below. Request builder supports up to 4096 key length.

7.38 /api/v4/payout-form

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Test Scenario](#)
- [Postman Collection](#)
- [Request Builder](#)

7.38.1 Introduction

To make a payout-form request send an HTTPS POST request - by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.38.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/payout-form/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/payout-form/ENDPOINTID

7.38.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Ask Support Manager if Conditional fields are Required for integration.

Parameter Name	Description	Value
client_orderid	Connecting Party's order identifier.	Necessity: Required Type: String Length: 128
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3

continues on next page

Table 137 – continued from previous page

Parameter Name	Description	Value
order_desc	Brief order description.	Necessity: Conditional Type: String Length: 64
ipaddress	Receiver's IP address (IPv4 or IPv6).	Necessity: Conditional Type: String Length: 7-45
purpose	Payout purpose.	Necessity: Conditional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 137 – continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024
redirect_url	URL, where the Receiver is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status . Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command . Pass https://doc2.codetime.net if you have no need to return Receiver anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 137 – continued from previous page

Parameter Name	Description	Value
redirect_succes_url	URL, where the Receiver is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Receiver anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Lenght: 1024
redirect_fail_url	URL, where the Receiver is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if you use non-3DS schema for transactions processing and you have no need to return Receiver anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
account_number	Account number.	Necessity: Conditional Type: String Length: 24
account_name	Bank account.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 137 – continued from previous page

Parameter Name	Description	Value
ewallet_type	Type of e-wallet.	Necessity: Conditional Type: String Length: 64
ewallet_wallet	E-wallet ID.	Necessity: Conditional Type: String Length: 128
crypto_wallet_address	Address of crypto wallet.	Necessity: Conditional Type: String Length: 64
bank_name	Bank Name.	Necessity: Conditional Type: String Length: 255
bank_branch	Bank Branch Name.	Necessity: Conditional Type: String Length: 255
bank_code	Bank code.	Necessity: Conditional Type: String Length: 32
bank_address1	Bank address.	Necessity: Conditional Type: String Length: 255
bank_zip_code	Bank postal ZIP code.	Necessity: Conditional Type: String Length: 255

continues on next page

Table 137 – continued from previous page

Parameter Name	Description	Value
bank_province	Bank province.	Necessity: Conditional Type: String Length: 255
bank_area	Bank area.	Necessity: Conditional Type: String Length: 255
routing_number	Routing number used to identify specific bank branches in China.	Necessity: Conditional Type: String Length: 16
legal_person_name	Name on the legal document.	Necessity: Conditional Type: String Length: 128
legal_person_document_number	Number of legal document.	Necessity: Conditional Type: String Length: 128
receiver_first_name	Receiver first name, also can be sent as first_name.	Necessity: Conditional Type: String Length: 128
receiver_last_name	Receiver last name, also can be sent as last_name.	Necessity: Conditional Type: String Length: 128
receiver_birthday	Receiver birthday, also can be sent as birthday.	Necessity: Conditional Type: Numeric Length: 30

continues on next page

Table 137 – continued from previous page

Parameter Name	Description	Value
receiver_country_code	Receiver country code, also can be sent as country.	Necessity: Conditional Type: String Length: 3
receiver_state	Receiver state, should be provided for countries that have states (USA, Canada, Australia), also can be sent as state.	Necessity: Conditional Type: String Length: 4
receiver_city	Receiver city, also can be sent as city.	Necessity: Conditional Type: String Length: 128
receiver_zip_code	Receiver zip code, also can be sent as zip_code.	Necessity: Conditional Type: Numeric Length: 32
receiver_address1	Receiver address, also can be sent as address1.	Necessity: Conditional Type: String Length: 256
receiver_phone	Receiver phone, also can be sent as phone.	Necessity: Conditional Type: Numeric Length: 128
receiver_email	Receiver E-mail, also can be sent as email.	Necessity: Conditional Type: String Length: 128
receiver_identity_document_id	Receiver identity document identifier, also can be sent as identity_document_id.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 137 – continued from previous page

Parameter Name	Description	Value
receiver_identity_document_number	Receiver identity document number, also can be sent as identity_document_number.	Necessity: Conditional Type: String Length: 128
merchant_data	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party's Callback.	Necessity: Optional Type: String Length: 64k
merchant_form_data	Parameters sent in merchant form data API parameter are parsed into macros with the same name, the parameter is url-encoded, example: test-param%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters[a-zA-Z0-9], parameter value characters[a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128
preferred_language	Preferred language.	Necessity: Optional Type: String Length: 2
customer_level	Customer level in CMS system	Necessity: Optional Type: Varchar Length: 32
customer_id	Customer ID in CMS system. Required if transaction goes via CMS (Payment Gateway mode)	Necessity: Optional Type: Int Length: 10

continues on next page

Table 137 – continued from previous page

Parameter Name	Description	Value
merchant_customer_identifier	Merchant Customer ID in CMS system. Required if transaction goes via CMS (CRM mode)	Necessity: Optional Type: Varchar Length: 64
card_recurring_payment_id	Payer's tokenized cardholder's data ID, referred as Recurring Payment ID (RPI). Send either card_recurring_payment_id or combination of credit_card_number, card_printed_name, expire_month and expire_year, not all. To create card_recurring_payment_id see /api/v4/create-card-ref . Note: For the scenario of payment to a card inside the system, this card will be considered as a source, and all processing limits, lists and fraud scoring will be applied to it as a source card.	Necessity: Conditional Type: Long

7.38.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter Name	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.
redirect_url	The URL to the page where the Connecting Party should redirect the client's browser. Connecting Party should send HTTP 302 redirect.

7.38.5 Request Example

```

POST /paynet/api/v4/payout-form/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↳ "GtAAIvMXjF6QLjWdAFk8L9C4glV9rwQ0", oauth_signature=
  ↳ "RpPfm4BDtjrDikqy3hvQUIiLdWOM4Gao0VzSkFIbvEI1RSK969cr0UmNHXFNXgoNKV7yvI98jlTar3IZPin
  ↳ %2B8JwXRN0EgS8SUQHd1xPQaKD6RdLXazrwNUaxl0yeg9IwRLBJz5TzF7DphCQVwTvKZkSYfFnLayQyheExhzSJPCFUm
  ↳ %2Bh33PtxJAAsCscYgTGLNkaVRYARZ5b2Pe
  ↳ %2FSygiTeg2xevn0yjoqd1Rl0wbB3d1EvqGB7AFJMxPMG1lMe33w6FKvU
  ↳ %2B6rJgIGEipkoTZ8HITwZybZvhrFDWst10DTzJxfuxd8JBE0Pn1dwDBAbLkPKQd5%2F%2BL0sszUndJ
  ↳ %2FJSAItNWMmEQ7QBumvYG2qgUSKJi%2FsG7VM%2FJY1esr5CELW%2FeMXfWEwNMN%2F%2BUQ8t
  ↳ %2F7Y0WQpZAmtfykRyM%2BNwGbHaFWt%2F6hncfXtwbYI0u5XtWyi0n37CxdY5CB9sZyo
  ↳ %2FAFP7isByhs3kRppc%2BioFlpsyXWi1K3LvevqheGC8jDs f6XTqh%2Fn%2B1nj jopUmkuKFB1qz xu0I
  ↳ %2F04AIIPzm%2BvSfJmTz05iYV11
  ↳ %2FtFzLEr9BCVRXShbjACwRFEDEQv9C73csGpWop9XGB7CKLaPD3K1LswVNMu0hZyU4FxLP
  ↳ %2Bg1EpJ7xJB45arMHSBHU1lGnedAHh7Nq46Si1mEOBpm0rdEUgRjfZbGKu12V02U9B5q8Nack3QNHD9yJ3hyEEaURGg2yzSa
  ↳ %2FeaMkF%2FFL%2B6DxicGPQaPTMezTRnHTvkI4rZcRZoxl0cEeI3a%2FWTiBXxcUVwfnp0aWe0i4DWdSY
  ↳ %2BJuiIIGRjBjh5owtR87lexWAXUPH8a5bmrS9TcHNF1amggunLjzk5hDAiMgJRyhL7btB2B3rkHbNvbfZ6QnNAjvgBWdoB7dj
  ↳ %2Bg5aTFxhyQSEilhkIq9WpPULUR3stXyP31zgs4BgCbi3t10YyV2XCl0W%2BrXa5x
  ↳ %2FqFbfN3AJB7ttfq4TNI5G0CeabcZ0T
  ↳ %2FhlHn2sopQPkX1ZnokM03Tof7TANmZ9nM0avlTjFJj0ppunPoc7Uq4VbT7QyTayf09d38IizWEnc77f6ZiDxnUQW0jJhrt0
  ↳ %2B82LcA2oqVeu6ifgjGuCJBQiC%2F9nsEetosRz8Z2vp1jiZ%2F3u61n0dwkowGftGcnIT73JFr7hg%3D
  ↳ %3D", oauth_signature_method="RSA-SHA256", oauth_timestamp="1678178824", oauth_
  ↳ version="1.0"
Content-Length: 381
Content-Type: application/x-www-form-urlencoded
Connection: close

account_number=1234567890
&amount=100
&bank_branch=test_branch
&bank_name=test_bank
&client_orderid=12345
&currency=USD
&order_desc=TEST
&redirect_url=http%3A%2F%2Fhttps://doc2.codetime.net%2Fdoc%2Fdummy.htm%09
&server_callback_url=https%3A%2F%2Fhttpstat.us%2F200

```

7.38.6 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Tue, 07 Mar 2023 08:47:54 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US

```

(continues on next page)

(continued from previous page)

```

Strict-Transport-Security: max-age=31536000
Content-Length: 281

type=async-form-response
&serial-number=00000000-0000-0000-0000-000002e33afd
&merchant-order-id=12345
&paynet-order-id=6993513
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
↪%2FBB587546567A31587163597A684535634A775969614A5A6367507733385468565A54514E48467135715A74773D

```

7.38.7 Fail Response Example

```

HTTP/1.1 403 Forbidden
Server: server
Date: Thu, 25 Aug 2022 06:50:16 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>403</title>

<style type="text/css">
body {
font-family: Arial, sans-serif;
font-size: 130%;
background-color: #eee;
}

p {
margin: 10em auto 0;
width: 500px;
border: 1px solid gray;
text-align: center;
vertical-align: middle;
padding: 40px 20px;
background-color: #fff;
-webkit-border-radius: 20px;
-moz-border-radius: 20px;
border-radius: 20px;
}
</style>
</head>

<body>
<p>Access is denied</p>
</body>
</html>

```

7.38.8 Test Scenario

Different Payout transaction statuses can be received on sandbox depending on the account_number value passed in Payout request.

Testing account_number values:

- account_number = 1234567890 to get APPROVED
- account_number = 0987654321 to get DECLINED
- account_number = 1987654321 to get PROCESSOR_INTERNAL_ERROR

7.38.9 Postman Collection

7.38.10 Request Builder

Insert PKCS#1 PEM private key for sandbox environment in the field below. Request builder supports up to 4096 key length.

7.39 /api/v4/payout-by-ref/

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.39.1 Introduction

Payout By Reference is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.39.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/payout-by-ref/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/payout-by-ref/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v4/payout-by-ref/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v4/payout-by-ref/group/ENDPOINTGROUPID

7.39.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Note:

Ask Support Manager if Conditional fields are Required for integration.

Parameter Name	Description	Value
client_orderid	Connecting Party's order identifier.	Necessity: Required Type: String Length: 128
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3

continues on next page

Table 140 – continued from previous page

Parameter Name	Description	Value
destination-card-ref-id	Card reference id to destination card, obtained at Card Registration step. For the scenario of payment to a card inside the system, this card will be considered as a destination, and all processing limits, lists and fraud scoring will be applied to it as a destination card.	Necessity: Required Type: Numeric Length: 20
order_desc	Brief order description.	Necessity: Optional Type: String Length: 64
ipaddress	Receiver's IP address (IPv4 or IPv6).	Necessity: Conditional Type: String Length: 7-45
purpose	Payout purpose.	Necessity: Conditional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 128

continues on next page

Table 140 – continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 128
redirect_url	URL, where the Receiver is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status . Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command . Pass https://doc2.codetime.net if you have no need to return Receiver anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 128

continues on next page

Table 140 – continued from previous page

Parameter Name	Description	Value
redirect_success_url	URL, where the Receiver is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Receiver anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
redirect_fail_url	URL, where the Receiver is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if there is no need to redirect Receiver anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
account_number	Account Number.	Necessity: Conditional Type: String Length: 24
account_name	Bank account.	Necessity: Optional Type: String Length: 128
ewallet_type	Type of e-wallet.	Necessity: Conditional Type: String Length: 64

continues on next page

Table 140 – continued from previous page

Parameter Name	Description	Value
ewallet_wallet	E-wallet ID.	Necessity: Conditional Type: String Length: 128
crypto_wallet_address	Address of crypto wallet.	Necessity: Conditional Type: String Length: 64
bank_name	Bank Name.	Necessity: Conditional Type: String Length: 255
bank_branch	Bank Branch Name.	Necessity: Conditional Type: String Length: 255
bank_code	Bank code.	Necessity: Conditional Type: String Length: 32
bank_city	Bank city.	Necessity: Conditional Type: String Length: 128
bank_address1	Bank address.	Necessity: Conditional Type: String Length: 255
bank_zip_code	Bank postal ZIP code.	Necessity: Conditional Type: String Length: 255

continues on next page

Table 140 – continued from previous page

Parameter Name	Description	Value
bank_province	Bank province.	Necessity: Conditional Type: String Length: 255
bank_area	Bank area.	Necessity: Conditional Type: String Length: 255
routing_number	Routing number used to identify specific bank branches in China.	Necessity: Conditional Type: String Length: 16
legal_person_name	Name on the legal document.	Necessity: Conditional Type: String Length: 128
legal_person_document_number	Number of legal document.	Necessity: Conditional Type: String Length: 128
receiver_first_name	Receiver first name, also can be sent as first_name.	Necessity: Conditional Type: String Length: 128
receiver_last_name	Receiver last name, also can be sent as last_name.	Necessity: Conditional Type: String Length: 128
receiver_birthday	Receiver birthday, also can be sent as birthday.	Necessity: Conditional Type: Numeric Length: 30

continues on next page

Table 140 – continued from previous page

Parameter Name	Description	Value
re-ceiver_country_code	Receiver country code, also can be sent as country.	Necessity: Conditional Type: String Length: 3
receiver_state	Receiver state, should be provided for countries that have states (USA, Canada, Australia), also can be sent as state.	Necessity: Conditional Type: String Length: 4
receiver_city	Receiver city, also can be sent as city.	Necessity: Conditional Type: String Length: 128
receiver_zip_code	Receiver zip code, also can be sent as zip_code.	Necessity: Conditional Type: Numeric Length: 32
receiver_address1	Receiver address, also can be sent as address1.	Necessity: Conditional Type: String Length: 256
receiver_phone	Receiver phone, also can be sent as phone.	Necessity: Conditional Type: Numeric Length: 128
receiver_email	Receiver E-mail, also can be sent as email.	Necessity: Conditional Type: String Length: 128
re-ceiver_identity_document_id	Receiver identity document identifier, also can be sent as identity_document_id.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 140 – continued from previous page

Parameter Name	Description	Value
re- ceiver_identity_document_number	Receiver identity document number. Also can be sent as identity_document_number.	Necessity: Conditional Type: String Length: 128
merchant_data	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
bank_bic	BIC of the receiver's bank.	Necessity: Optional Type: String Length: 128
receiver_inn	Receiver's INN.	Necessity: Optional Type: String Length: 128
customer_level	Customer level in CMS system	Necessity: Optional Type: Varchar Length: 32
customer_id	Customer ID in CMS system. Required if transaction goes via CMS (Payment Gateway mode)	Necessity: Optional Type: Int Length: 10
mer- chant_customer_identity	Merchant Customer ID in CMS system. Required if transaction goes via CMS (CRM mode)	Necessity: Optional Type: Varchar Length: 64
card-ref-id	Card reference ID used in subsequent recurring payments. Card reference ID can be created with v4 tokenization request or v2 tokenization request .	Necessity: Conditional Type: Long

7.39.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Payout Request Parameters	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.39.5 Request Example

```
POST /paynet/api/v4/payout-by-ref/121799 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/8.4.0
Accept: */*
Authorization: OAuth oauth_consumer_key="ErwinTestMerchant", oauth_nonce=
↳ "8qsDrhiDelGHlxY01aDjCh44u7isXJjL", oauth_signature="WpbNcRxNSDW%2BdJgftRc
↳ %2BAN0oe8KAP4CnpUkxjxStFYZEU9Vo%2F7uk9dSKEG%2B76C
↳ %2FdlzemILaRmikQqZg93ZK1cibT8zy97uYVDdWVmY2obDS0Vb8UipGjt2KFUyKr4awHW9xH%2BTzbR
↳ %2FMXwm3y%2Fx2RotELZWxumsL37P5Q%2BCquoAGc06jjkvkkSH9P4lBYKSmwhfqD40
↳ %2Br8FB3exzNz12FFBTwLp4ch2G9Cis5a0CkVrpjDB
↳ %2Fbbbr0utbNPZYhtH45rNz91QAbpvNJ91XjVwxiCm4lCldIO66gF73GQNSFKVG1mstc
↳ %2B941Dj1bVhXWQQW%2F9TyPiXfWGP8szg18rwsTHbj1zKCRIaw%2FqVbr0BPhbo%2BJjGMZp2hB1ei0
↳ %2FUPIkjeIZkWD2NgJRQEniJfjU7AlILs9laugm50fPwWR5Ji0uE5uSvwM6VWgXmpYe4USb819ZAK
↳ %2BFYTRI%2BenrycFZHmY
↳ %2Baf0la0HFtpnbn8FsrzzNb0vGXmXct2xXAQTa0RqMeLRRtkD3DY4rJa8qe00zYnmT4ZhKsFZjImDdonA3WTG8G7awP9W0RK
↳ %2BAC6I00rkB13tw2eXP4yIPtEC7iGu%2FkElGz40rWFEU6cUcEDpeE0mA2%2Fe%2BZFK0w%3D", oauth_
↳ signature_method="RSA-SHA256", oauth_timestamp="1721975900", oauth_version="1.0"
Content-Length: 74
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

amount=10.42
&client_orderid=1
&currency=USD
&destination-card-ref-id=1461897
```

7.39.6 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Fri, 26 Jul 2024 06:39:23 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 138

type=async-response
&serial-number=00000000-0000-0000-0000-000002f36e41
&merchant-order-id=1
&paynet-order-id=7363634
&end-point-id=121799
```

7.39.7 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Fri, 26 Jul 2024 06:42:43 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 170

type=validation-error
&serial-number=00000000-0000-0000-0000-000002f36e42
&error-message=Project+with+currency+USD+does+not+apply+request+with+currency+AZN
&error-code=16
```

7.39.8 Postman Collection

7.39.9 Request Builder

Insert PKCS#1 PEM private key for sandbox environment in the field below. Request builder supports up to 4096 key length.

7.40 /api/v4/payout-check

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.40.1 Introduction

Payout check is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.40.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/payout-check/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/payout-check/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v4/payout-check/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v4/payout-check/group/ENDPOINTGROUPID

7.40.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
client_orderid	Connecting Party's order identifier.	Necessity: Required Type: String Length: 128
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Brief order description.	Necessity: Optional Type: String Length: 64
ipaddress	Receiver's IP address (IPv4 or IPv6).	Necessity: Conditional Type: String Length: 7-45
purpose	Payout check purpose.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 143 – continued from previous page

Parameter Name	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 128
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 128

continues on next page

Table 143 – continued from previous page

Parameter Name	Description	Value
redirect_url	URL, where the Receiver is redirected to upon completion of the transaction. Please note that redirection is performed in any case, no matter whether transaction is approved, declined in any other final status. Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Pass https://doc2.codetime.net if you have no need to return Receiver anywhere. Use either redirect_url or combination of redirect_success_url and redirect_fail_url, not both.	Necessity: Optional Type: String Length: 128
redirect_success_url	URL, where the Receiver is redirected to when transaction status is approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command. Otherwise put https://doc2.codetime.net if there is no need to redirect Receiver anywhere. Use either combination of redirect_success_url and redirect_fail_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 143 – continued from previous page

Parameter Name	Description	Value
redirect_fail_url	URL, where the Receiver is redirected to when transaction status is not approved (See status list). Connecting Party must not use the parameters come along with the redirect HTTP Request to treat the status of the transaction. Instead Connecting Party can utilize server_callback_url or status API command . Pass https://doc2.codetime.net if there is no need to redirect Receiver anywhere. Use either combination of redirect_fail_url and redirect_success_url or redirect_url, not both.	Necessity: Optional Type: String Length: 1024
credit_card_number	Customer's credit card number. Note: For the scenario of payment to a card inside the system, this card will be considered as a source, and all processing limits, lists and fraud scoring will be applied to it as a source card.	Necessity: Conditional Type: Numeric Length: 20
card_printed_name	Customer's full name, as printed on the card.	Necessity: Conditional Type: String Length: 128
expire_month	Credit card expiration month.	Necessity: Conditional Type: Numeric Length: 2
expire_year	Credit card expiration year.	Necessity: Conditional Type: Numeric Length: 4
cvv2	Receiver's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card.	Necessity: Conditional Type: Numeric Length: 3-4

continues on next page

Table 143 – continued from previous page

Parameter Name	Description	Value
account_number	Account Number.	Necessity: Conditional Type: String Length: 32
account_name	Bank account.	Necessity: Optional Type: String Length: 128
ewallet_type	Type of e-wallet.	Necessity: Conditional Type: String Length: 64
ewallet_wallet	E-wallet ID.	Necessity: Conditional Type: String Length: 128
crypto_wallet_address	Address of crypto wallet.	Necessity: Conditional Type: String Length: 64
bank_name	Bank Name.	Necessity: Conditional Type: String Length: 255
bank_branch	Bank Branch Name.	Necessity: Conditional Type: String Length: 255
bank_code	Bank code.	Necessity: Conditional Type: String Length: 32

continues on next page

Table 143 – continued from previous page

Parameter Name	Description	Value
bank_city	Bank city.	Necessity: Conditional Type: String Length: 128
bank_address1	Bank address.	Necessity: Conditional Type: String Length: 255
bank_zip_code	Bank postal ZIP code.	Necessity: Conditional Type: String Length: 255
bank_province	Bank province.	Necessity: Conditional Type: String Length: 255
bank_area	Bank area.	Necessity: Conditional Type: String Length: 255
routing_number	Routing number used to identify specific bank branches in China.	Necessity: Conditional Type: String Length: 16
legal_person_name	Name on the legal document.	Necessity: Conditional Type: String Length: 128
legal_person_document_number	Number of legal document.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 143 – continued from previous page

Parameter Name	Description	Value
receiver_first_name	Receiver first name, also can be sent as first_name.	Necessity: Conditional Type: String Length: 128
receiver_last_name	Receiver last name, also can be sent as last_name.	Necessity: Conditional Type: String Length: 128
receiver_middle_name	Receiver middle name, also can be sent as middle_name.	Necessity: Conditional Type: String Length: 128
receiver_birthday	Receiver birthday, also can be sent as birthday.	Necessity: Conditional Type: Numeric Length: 30
receiver_country_code	Receiver country code, also can be sent as country.	Necessity: Conditional Type: String Length: 3
receiver_state	Receiver state, should be provided for countries that have states (USA, Canada, Australia), also can be sent as state.	Necessity: Conditional Type: String Length: 4
receiver_city	Receiver city, also can be sent as city.	Necessity: Conditional Type: String Length: 128
receiver_zip_code	Receiver zip code, also can be sent as zip_code.	Necessity: Conditional Type: Numeric Length: 32

continues on next page

Table 143 – continued from previous page

Parameter Name	Description	Value
receiver_address1	Receiver address, also can be sent as address1.	Necessity: Conditional Type: String Length: 256
receiver_phone	Receiver phone, also can be sent as phone.	Necessity: Conditional Type: Numeric Length: 128
receiver_email	Receiver E-mail, also can be sent as email.	Necessity: Conditional Type: String Length: 128
receiver_identity_document_id	Receiver identity document identifier, also can be sent as identity_document_id.	Necessity: Conditional Type: String Length: 128
receiver_identity_document_number	Receiver identity document number, also can be sent as identity_document_number.	Necessity: Conditional Type: String Length: 128
merchant_data	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k
bank_bic	BIC of the receiver's bank.	Necessity: Optional Type: String Length: 128
receiver_inn	Receiver's INN.	Necessity: Optional Type: String Length: 128

continues on next page

Table 143 – continued from previous page

Parameter Name	Description	Value
customer_level	Customer level in CMS system	Necessity: Optional Type: Varchar Length: 32
customer_id	Customer ID in CMS system. Required if transaction goes via CMS (Payment Gateway mode)	Necessity: Optional Type: Int Length: 10
merchant_customer_identifier	Merchant Customer ID in CMS system. Required if transaction goes via CMS (CRM mode)	Necessity: Optional Type: Varchar Length: 64
card_recurring_payment_id	Payer's tokenized cardholder's data ID, referred as Recurring Payment ID (RPI). Send either card_recurring_payment_id or combination of credit_card_number, card_printed_name, expire_month and expire_year, not all. To create card_recurring_payment_id see /api/v4/create-card-ref . Note: For the scenario of payment to a card inside the system, this card will be considered as a source, and all processing limits, lists and fraud scoring will be applied to it as a source card.	Necessity: Conditional Type: Long
recurring-payment-id	Recurring Payment ID can be sent instead of cardholder data. Customer Data can be updated via /api/v4/update-recurring-payment/ . Recurring Payment ID creation is initiated through HTTPS POST request by using URLs and the parameters specified below. Use OAuth RSA-SHA256 for authentication.	Necessity: Conditional Type: Long

* Ask Support Manager if Conditional fields are Required for integration.

7.40.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Payout Check Response Parameters	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.40.5 Request Example

```

POST /paynet/api/v4/payout-check/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
↳ "b5E31Tw6SVjauE29u0f2j0LnuUSXmVdE", oauth_signature=
↳ "WrW79JHNUVwDRhCGWQYgaN6xmJXpQxy8XSNyCOL6b2Wyf7V5BWGMe2TZa1bjC9Ze00Q3FcQxeGGHv0
↳ %2F7hsMAsJNQEET321VNsbdwao2Ep
↳ %2Bp7eoYiGYrVveSrSWldiCrBf3AJYJZM0PTJ67Sl8XyeTVBHT4kpC5qBu3xDQ3aFfKnMRTmn9fiVsYsYu3DQrsHM1K9uAoltG
↳ %2FvkiP%2BZ44i8ghIUp61NWSrJ4Ky69JiZ%2FoaVvmUTEaanc%2F%2B%2BQT6jBwWY%2Bb
↳ %2FTLURxtSakeNfcjn1JVwRf4aCX2fhYg5ozH
↳ %2BjiXF6eRb83WVqBUAwykSq35pLU3Vmua3pKMKAJK1ZRDZdjGrT50KJg5tBniC4JFzdQqjQf
↳ %2FhFnDYodfIK3S2qZo%2FD3Bmlya46iEcK6SAQdNBBQue3E5Qi8FEHYrY1o7K8wDyzT1QzqqHF
↳ %2BQdmXcElSGu9ge0Y655%2BbGtXhnsUWnKE00NGqErvAwzm7yUg0e5QWHVf505aE7pr5K4z
↳ %2Fzj7AvkuD7R1savqam%2BnnuSfq1E%2BnnnN7mTcC0g18Sr38vdTshcGq99YW3xWKc
↳ %2FpuooZYdYa5A6u46o%2BREZSTCD2XexcV49%2F9eVn3xdoTXyq4NISJSY8U7ThKnr0g%3D", oauth_
↳ signature_method="RSA-SHA256", oauth_timestamp="1677831012", oauth_version="1.0"
Content-Length: 150
Content-Type: application/x-www-form-urlencoded
Connection: close

account_name=1234
&account_number=1234
&amount=10.42
&client_orderid=1
&currency=USD
&routing_number=15
&server_callback_url=https%3A%2F%2Fhttpstat.us%2F200

```

7.40.6 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Fri, 03 Mar 2023 08:10:34 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=async-response
&serial-number=00000000-0000-0000-0000-000002e2c322
&merchant-order-id=1
&paynet-order-id=6982864
&end-point-id=39915

```

7.40.7 Fail Response Example

```

HTTP/1.1 403 Forbidden
Server: server
Date: Thu, 25 Aug 2022 06:50:16 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>403</title>

  <style type="text/css">
  body {
    font-family: Arial, sans-serif;
    font-size: 130%;
    background-color: #eee;
  }

  p {
    margin: 10em auto 0;
    width: 500px;
    border: 1px solid gray;
    text-align: center;
    vertical-align: middle;
    padding: 40px 20px;
    background-color: #fff;
  }

```

(continues on next page)

(continued from previous page)

```
-webkit-border-radius: 20px;  
-moz-border-radius: 20px;  
border-radius: 20px;  
}  
</style>  
</head>  
  
<body>  
<p>Access is denied</p>  
</body>  
</html>
```

7.40.8 Postman Collection

7.40.9 Request Builder

Insert PKCS#1 PEM private key for sandbox environment in the field below. Request builder supports up to 4096 key length.

7.41 /api/v4/payout-pay

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.41.1 Introduction

Payout pay is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.41.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/payout-pay/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/payout-pay/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v4/payout-pay/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v4/payout-pay/group/ENDPOINTGROUPID

7.41.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
client_orderid	Connecting Party's order identifier.	Necessity: Required
orderid	Order id assigned to the order by Doc2.0.	Necessity: Required
login	Connecting Party login name.	Necessity: Required

* Ask Support Manager if Conditional fields are Required for integration.

7.41.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Payout Pay Response Parameters	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.41.5 Request Example

```
POST /paynet/api/v4/payout-pay/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
↳ "b5E31Tw6SVjauE29u0f2j0LnuUSXmVdE", oauth_signature=
↳ "WrW79JHNUVwDRhCGWQYgaN6xmJXpQxy8XSNyCOL6b2Wyf7V5BWGMe2TZa1bjC9Ze00Q3FcQxeGGHv0
↳ %2F7hsMAsJNQEET321VNsbdwao2Ep
↳ %2Bp7eoYiGYrVveSrSW1diCrBf3AJYJZM0PTJ67S18XyeTVBHT4kpC5qBu3xDQ3aFfKnmRTmn9fiVsYsYu3DQrsHM1K9uAo1tg
↳ %2FvkiP%2BZ44i8ghIUp61NWSrJ4Ky69JiZ%2FoaVVMUTEaanc%2F%2B%2BQT6jBwWly%2Bb
↳ %2FTLURxtSakeNfcjn1JVwRf4aCX2fhYG5ozH
↳ %2BjiXF6eRb83WVqBUAwykSq35pLU3Vmua3pKMKAJK1ZRDZdjGrT50KJg5tBniC4JFzdQqjQf
↳ %2FhFnDYodfIK3S2qZo%2FD3Bmly46iEcK6SAQdNBBQue3E5Qi8FEHYrY1o7K8wDyzT1QzqqHF
↳ %2BQdmXcElSGu9ge0Y655%2BbGtXhnsUWnKE00NGqErvAwzm7yUg0e5QWHVf505aE7pr5K4z
↳ %2Fzj7AvkuD7R1savqam%2BnnuSfq1E%2BnnnN7mTcC0g18Sr38vdTshcGq99YW3xWKc
↳ %2FpuooZYdYa5A6u46o%2BREZSTCD2XexcV49%2F9eVn3xdoTXyq4NISJSY8U7ThKnr0g%3D", oauth_
↳ signature_method="RSA-SHA256", oauth_timestamp="1677831012", oauth_version="1.0"
Content-Length: 51
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=3
&login=test-merchant
&orderid=900258
```

7.41.6 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Fri, 03 Mar 2023 08:10:34 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=async-response
&serial-number=00000000-0000-0000-0000-000002e2c322
&merchant-order-id=1
&paynet-order-id=900258
&end-point-id=39915

```

7.41.7 Fail Response Example

```

HTTP/1.1 403 Forbidden
Server: server
Date: Thu, 25 Aug 2022 06:50:16 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>403</title>

  <style type="text/css">
  body {
    font-family: Arial, sans-serif;
    font-size: 130%;
    background-color: #eee;
  }

  p {
    margin: 10em auto 0;
    width: 500px;
    border: 1px solid gray;
    text-align: center;
    vertical-align: middle;
    padding: 40px 20px;
    background-color: #fff;
  }

```

(continues on next page)

(continued from previous page)

```
-webkit-border-radius: 20px;  
-moz-border-radius: 20px;  
border-radius: 20px;  
}  
</style>  
</head>  
  
<body>  
<p>Access is denied</p>  
</body>  
</html>
```

7.41.8 Postman Collection

7.41.9 Request Builder

Insert PKCS#1 PEM private key for sandbox environment in the field below. Request builder supports up to 4096 key length.

7.42 /api/v4/withdrawal-sbp

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.42.1 Introduction

To make a withdrawal send an HTTPS POST request - by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.42.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/withdrawal-sbp/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/withdrawal-sbp/ENDPOINTID

7.42.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter Name	Description	Value
client_orderid	Connecting Party's order identifier.	Necessity: Required Type: String Length: 128
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Sample values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3

continues on next page

Table 149 – continued from previous page

Parameter Name	Description	Value
receiver_phone	Receiver phone, also can be sent as phone.	Necessity: Required Type: Numeric Length: 128
merchant_data	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party Callback. For SBP, this parameter is used to pass Receiver's bank code. Contact Doc2.0 for details.	Necessity: Required Type: String Length: 64k
bank_id	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party Callback. For SBP, this parameter is used to pass Receiver's bank code. Also can be sent as merchant_data. Contact Doc2.0 for details.	Necessity: Required Type: String Length: 64k
ipaddress	Receiver's IP address (IPv4 or IPv6).	Necessity: Conditional Type: String Length: 7-45
purpose	Withdrawal purpose.	Necessity: Conditional Type: String Length: 128
receiver_first_name	Receiver first name, also can be sent as first_name.	Necessity: Conditional Type: String Length: 128
receiver_last_name	Receiver last name, also can be sent as last_name.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 149 – continued from previous page

Parameter Name	Description	Value
receiver_country_code	Receiver country code, also can be sent as country.	Necessity: Conditional Type: String Length: 3
receiver_state	Receiver state, should be provided for countries that have states (USA, Canada, Australia), also can be sent as state.	Necessity: Conditional Type: String Length: 4
receiver_city	Receiver city, also can be sent as city.	Necessity: Conditional Type: String Length: 128
receiver_zip_code	Receiver zip code, also can be sent as zip_code.	Necessity: Conditional Type: Numeric Length: 32
receiver_address1	Receiver address, also can be sent as address1.	Necessity: Conditional Type: String Length: 256
receiver_email	Receiver E-mail, also can be sent as email.	Necessity: Conditional Type: String Length: 128
order_desc	Brief order description.	Necessity: Optional Type: String Length: 64

continues on next page

Table 149 – continued from previous page

Parameter Name	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 128
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 128

* Ask Support Manager if Conditional fields are Required for integration.

7.42.4 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Withdrawal Request Parameters	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.42.5 Request Example

```
POST /paynet/api/v4/withdrawal-sbp/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
↳ "b5E31Tw6SVjauE29u0f2j0LnuUSXmVdE", oauth_signature=
↳ "WrW79JHNUVwDRhCGWQYgaN6xmJXpQxy8XSNyCOL6b2Wyf7V5BWGMe2TZa1bjC9Ze00Q3FcQxeGGHv0
↳ %2F7hsMA sJNQEET321VNs bDwao2Ep
↳ %2Bp7eoYiGYrVveSrSWldiCrBf3AJYJZM0PTJ67Sl8XyeTVBHT4kpC5qBu3xDQ3aFfKn mRTmn9fiVsYsYu3DQrsHM1K9uAoltG
↳ %2FvkiP%2BZ44i8ghIU p61NWSrJ4Ky69JiZ%2FoaVvmUTEaanc%2F%2B%2BQT6jBwW y%2Bb
↳ %2FTL UrxtSakeNfcjn1JVwRf4aCX2fh yG5ozH
↳ %2BjiXf6eRb83WVqBUAw ykSq35pLU3Vmua3pKMKAJK1ZR DZdjGrT50KJg5tBniC4JFzdQqjQf
↳ %2FhFnDYodfIK3S2qZo%2FD3Bmly a46iEcK6SAQdNBBQue3E5Qi8FEHYrY1o7K8wDyzT1QzqqHF
↳ %2BQdmXcElSGu9ge0Y655%2BbGtXhnsUWnKE00NGqErvAwzm7yUg0e5QWHVf505aE7pr5K4z
↳ %2Fzj7AvkuD7R1savqam%2BnnuSfq1E%2BnnnN7mTcC0g18Sr38vdTshcGq99YW3xWKc
↳ %2FpuooZYdYa5A6u46o%2BREZSTCD2XexcV49%2F9eVn3xdoTX Yq4NISJSY8U7ThKnr0g%3D", oauth_
↳ signature_method="RSA-SHA256", oauth_timestamp="1677831012", oauth_version="1.0"
Content-Length: 150
Content-Type: application/x-www-form-urlencoded
Connection: close

amount: 100.00
&client_orderid=123456
&currency=RUB
&merchant_data=100000000111
&order_desc=Test Order Description
&phone=89031234567
&ipaddress=127.0.0.1
&server_callback_url=https://httpstat.us/200
```

7.42.6 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Fri, 03 Mar 2023 08:10:34 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=async-response
&serial-number=00000000-0000-0000-0000-000002e2c322
&merchant-order-id=1
&paynet-order-id=6982864
&end-point-id=39915

```

7.42.7 Fail Response Example

```

HTTP/1.1 403 Forbidden
Server: server
Date: Thu, 25 Aug 2022 06:50:16 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>403</title>

  <style type="text/css">
body {
font-family: Arial, sans-serif;
font-size: 130%;
background-color: #eee;
}

p {
margin: 10em auto 0;
width: 500px;
border: 1px solid gray;
text-align: center;
vertical-align: middle;
padding: 40px 20px;
background-color: #fff;

```

(continues on next page)

(continued from previous page)

```
-webkit-border-radius: 20px;  
-moz-border-radius: 20px;  
border-radius: 20px;  
}  
</style>  
</head>  
  
<body>  
<p>Access is denied</p>  
</body>  
</html>
```

7.42.8 Postman Collection

7.42.9 Request Builder

Insert PKCS#1 PEM private key for sandbox environment in the field below. Request builder supports up to 4096 key length.

7.43 /api/v4/transfer

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Additional Parameters](#)
 - [For Connecting Party](#)
 - [For Payment Institutions](#)
 - [Sender Customer Data](#)
 - [Receiver Customer Data](#)
- [Response Parameters](#)
- [Request & Response Examples](#)
 - [Transfer from PAN to PAN](#)
 - [Transfer from PAN to RPI](#)
 - [Transfer from RPI to PAN](#)
 - [Transfer from RPI to RPI](#)
 - [Transfer from Deposit to PAN](#)
 - [Transfer from Deposit to RPI](#)
- [Postman Collection](#)
- [Request Builder](#)

7.43.1 Introduction

Transfer is a type of complex transaction that allows to move funds between bank cards (Primary Account Number - PAN), card tokens (Recurring Payment ID - RPI) and Connecting Party bank account (deposits).

Transfer is initiated through HTTPS POST request to the [URL](#) with the [required parameters](#) depending on a certain transfer Use-Case (see below). Some integrations might require [Additional parameters](#). Use [OAuth RSA-SHA256](#) for authentication.

Possible transfer Use-Cases:

PAN to PAN	Transfer from PAN to PAN occurs when transferring funds from one bank card to another, when specifying for each card their card numbers, and not the recurring payment id.
PAN to RPI	Transfer from PAN to RPI occurs when funds are transferred from one bank card for which its number is indicated to another bank card for which its recurring payment id is indicated.
RPI to PAN	Transfer from RPI to PAN occurs when funds are transferred from one bank card for which its recurring payment id is indicated to another bank card for which its number is indicated.
RPI to RPI	Transfer from RPI to RPI occurs when transferring funds from one bank card to another, when specifying for each card their recurring payment id, and not the card number.
Deposit to PAN	Deposit Transfer to PAN occurs when funds are transferred from Connecting Party account to a bank card by its number.
Deposit to RPI	Deposit Transfer to RPI occurs when funds are transferred from Connecting Party account to a bank card by recurring payment id.

7.43.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/transfer/	https://api.doc2.com/paynet/api/v4/transfer/
https://sandbox.doc2.com/paynet/api/v4/transfer/	https://api.doc2.com/paynet/api/v4/transfer/

7.43.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

PAN to PAN

Parameter	Description	Value
credit_card_number	Sender`s credit card number. For the scenario of payment from a card inside the system, this card will be considered as a source.	Necessity: Required Type: Numeric Length: 19
cvv2	Sender`s CVV2 code. CVV2 (Card verification value) is the three of four-digit number farthest to the right on the flip side of a credit card.	Necessity: Required Type: String Length: 3-4
expire_month	Sender credit card`s month of expiration.	Necessity: Required Type: String Length: 2
expire_year	Sender credit card`s year of expiration.	Necessity: Required Type: String Length: 2-4
card_printed_name	Sender`s card printed name.	Necessity: Required Type: String Length: 128
destination-card-no	Receiver`s card PAN. For the scenario of payment to a card inside the system, this card will be considered as a destination, and all processing limits, lists and fraud scoring will be applied to it as a destination card.	Necessity: Required Type: Numeric Length: 19
deposit2card	Marker of Deposit to Card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: False Type: Boolean
destination_expire_month	Receiver credit card`s month of expiration.	Necessity: Optional Type: String Length: 2

continues on next page

Table 153 – continued from previous page

Parameter	Description	Value
destination_expire_year	Receiver credit card's year of expiration.	Necessity: Optional Type: String Length: 2-4
destination_card_printed_name	Receiver's card printed name.	Necessity: Optional Type: String Length: 128
ipaddress	The Sender or Receiver's IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.100.101	Necessity: Required Type: String Length: 7-45
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
purpose	Destination to where the payment goes. It is useful for the Connecting Party who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999; mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128

continues on next page

Table 153 – continued from previous page

Parameter	Description	Value
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved, declined or in any other final status. Optional for direct integration (non-form) deposit2card.	Necessity: Optional Type: String Length: 1024
order_desc	Order description.	Necessity: Required Type: String Length: 64k
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 153 – continued from previous page

Parameter	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

PAN to RPI

Parameter	Description	Value
credit_card_number	Sender's credit card number. For the scenario of payment from a card inside the system, this card will be considered as a source.	Necessity: Required Type: Numeric Length: 19
cvv2	Sender's CVV2 code. CVV2 (Card verification value) is the three of four-digit number farthest to the right on the flip side of a credit card.	Necessity: Required Type: String Length: 3-4
expire_month	Sender credit card's month of expiration.	Necessity: Required Type: String Length: 2
expire_year	Sender credit card's year of expiration.	Necessity: Required Type: String Length: 2-4

continues on next page

Table 154 – continued from previous page

Parameter	Description	Value
card_printed_name	Sender's card printed name.	Necessity: Required Type: String Length: 128
deposit2card	Marker of Deposit to Card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: False Type: Boolean
destination_card_recurring_payment_id	Receiver's recurring payment ID. To create destination_card_recurring_payment_id please see Card Reference ID Creation . For the scenario of payment to a card inside the system, this card will be considered as a destination, and all processing limits, lists and fraud scoring will be applied to it as a destination card.	Necessity: Required Type: Long
ipaddress	The Sender or Receiver's IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.1.101	Necessity: Required Type: String Length: 7-45
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10

continues on next page

Table 154 – continued from previous page

Parameter	Description	Value
purpose	Destination to where the payment goes. It is useful for the Connecting Party who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +99999999999; mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved, declined or in any other final status. Optional for direct integration (non-form) deposit2card.	Necessity: Optional Type: String Length: 1024
order_desc	Order description.	Necessity: Required Type: String Length: 64k
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 154 – continued from previous page

Parameter	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

RPI to PAN

Parameter	Description	Value
destination-card-no	Receiver's card PAN. For the scenario of payment to a card inside the system, this card will be considered as a destination, and all processing limits, lists and fraud scoring will be applied to it as a destination card.	Necessity: Required Type: Numeric Length: 19
deposit2card	Marker of Deposit to Card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: False Type: Boolean
cvv2	Sender's CVV2 code. CVV2 (Card verification value) is the three of four-digit number farthest to the right on the flip side of a credit card.	Necessity: Required Type: String Length: 3-4
destination_expire_month	Receiver credit card's month of expiration.	Necessity: Optional Type: String Length: 2

continues on next page

Table 155 – continued from previous page

Parameter	Description	Value
destination_expire_year	Receiver credit card's year of expiration.	Necessity: Optional Type: String Length: 2-4
destination_card_printed_name	Receiver's card printed name.	Necessity: Optional Type: String Length: 128
card_recurring_payment_id	Sender's recurring payment ID. To create card_recurring_payment_id please see Card Reference ID Creation. For the scenario of payment from a card inside the system, this card will be considered as a source.	Necessity: Required Type: Long
ipaddress	The Sender or Receiver's IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.100.101	Necessity: Required Type: String Length: 7-45
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
purpose	Destination to where the payment goes. It is useful for the Connecting Party who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999; mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128

continues on next page

Table 155 – continued from previous page

Parameter	Description	Value
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved, declined or in any other final status. Optional for direct integration (non-form) deposit2card.	Necessity: Optional Type: String Length: 1024
order_desc	Order description.	Necessity: Required Type: String Length: 64k
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 155 – continued from previous page

Parameter	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

RPI to RPI

Parameter	Description	Value
deposit2card	Marker of Deposit to Card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: False Type: Boolean
cvv2	Sender's CVV2 code. CVV2 (Card verification value) is the three of four-digit number farthest to the right on the flip side of a credit card.	Necessity: Required Type: String Length: 3-4
destination_card_recurring_payment_id	Receiver's recurring payment ID. To create destination_card_recurring_payment_id please see Card Reference ID Creation . For the scenario of payment to a card inside the system, this card will be considered as a destination, and all processing limits, lists and fraud scoring will be applied to it as a destination card.	Necessity: Required Type: Long

continues on next page

Table 156 – continued from previous page

Parameter	Description	Value
card_recurring_payment_id	Sender's recurring payment ID. To create card_recurring_payment_id please see Card Reference ID Creation. For the scenario of payment from a card inside the system, this card will be considered as a source.	Necessity: Required Type: Long
ipaddress	The Sender or Receiver's IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.1.100.191	Necessity: Required Type: String Length: 7-45
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
purpose	Destination to where the payment goes. It is useful for the Connecting Party who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999; mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved, declined or in any other final status. Optional for direct integration (non-form) deposit2card.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 156 – continued from previous page

Parameter	Description	Value
order_desc	Order description.	Necessity: Required Type: String Length: 64k
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 156 – continued from previous page

Parameter	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

Deposit to PAN

Parameter	Description	Value
destination-card-no	Receiver's card PAN. For the scenario of payment to a card inside the system, this card will be considered as a destination, and all processing limits, lists and fraud scoring will be applied to it as a destination card.	Necessity: Required Type: Numeric Length: 19
deposit2card	Marker of Deposit to Card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: True Type: Boolean
destination_expire_month	Receiver credit card's month of expiration.	Necessity: Optional Type: String Length: 2
destination_expire_year	Receiver credit card's year of expiration.	Necessity: Optional Type: String Length: 2-4

continues on next page

Table 157 – continued from previous page

Parameter	Description	Value
destination_card_printed_name	Receiver's card printed name.	Necessity: Optional Type: String Length: 128
ipaddress	The Sender or Receiver's IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.100.101	Necessity: Required Type: String Length: 7-45
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
purpose	Destination to where the payment goes. It is useful for the Connecting Party who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999; mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved, declined or in any other final status. Optional for direct integration (non-form) deposit2card.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 157 – continued from previous page

Parameter	Description	Value
order_desc	Order description.	Necessity: Required Type: String Length: 64k
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

Deposit to RPI

Parameter	Description	Value
deposit2card	Marker of Deposit to Card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: True Type: Boolean
destination_card_recurring_payment_id	Receiver's recurring payment ID. To create destination_card_recurring_payment_id please see Card Reference ID Creation . For the scenario of payment to a card inside the system, this card will be considered as a destination, and all processing limits, lists and fraud scoring will be applied to it as a destination card.	Necessity: Required Type: Long
ipaddress	The Sender or Receiver's IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.1.100.101	Necessity: Required Type: String Length: 7-45
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
purpose	Destination to where the payment goes. It is useful for the Connecting Party who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999; mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128

continues on next page

Table 158 – continued from previous page

Parameter	Description	Value
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved, declined or in any other final status. Optional for direct integration (non-form) deposit2card.	Necessity: Optional Type: String Length: 1024
order_desc	Order description.	Necessity: Required Type: String Length: 64k
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 158 – continued from previous page

Parameter	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

7.43.4 Additional Parameters

For Connecting Party

Use additional parameters for Connecting Party to specify Sender's browser information and Connecting Party 3DS results notification URL. The Connecting Party's site needs to accurately populate the browser information for each transaction. This data can be obtained by Connecting Party's servers. Ensure that the data is not altered or hard-coded, and that it is unique to each transaction.

Note: These parameters are not applicable for cases Deposit to PAN and Deposit to RPI.

Note: Browser data for 3DS 2.X is gathered by payment gateway system on 3DS authentication stage. For some processing channels, however, the browser data and/or connecting party URL for 3DS challenge results must be provided in initial transaction request. Please contact Support manager to clarify if these parameters should be included in request parameters.

Parameter Name	Description	Value
tds_areq_notification_url, alias tds_cres_notification_url	Fully qualified URL of Connecting Party system that will receive the CRes message or Error Message. This CRes message must be sent to Doc2.0. See details here Upload CRes Result .	Type: String Length: 256
customer_browser_info	If true, then the fields below must be present.	Type: Boolean
ipaddress	IP address of the browser as returned by the HTTP headers to the 3DS Requestor.	Type: String Length: 45
customer_browser_accept_header, alias customer_accept	Exact content of the HTTP accept headers as sent to the 3DS Requestor from the Cardholder's browser.	Type: String Length: 2048
customer_browser_color_depth	Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Required when browser_javascript_enabled=true.	Type: String Length: 2
customer_browser_java_enabled	Boolean that represents the ability of the cardholder browser to execute Java. Required when browser_javascript_enabled=true.	Type: Boolean
customer_browser_javascript_enabled	Boolean that represents the ability of the cardholder browser to execute JavaScript.	Type: Boolean
customer_browser_accept_language, alias customer_accept_language	Value representing the browser language as defined in IETF BCP47.	Type: String Length: 8
customer_browser_screen_height	Total height of the Cardholder's screen in pixels. Required when browser_javascript_enabled=true.	Type: Numeric Length: 6
customer_browser_screen_width	Total width of the Cardholder's screen in pixels. Required when browser_javascript_enabled=true.	Type: Numeric Length: 6

continues on next page

Table 159 – continued from previous page

Parameter Name	Description	Value
customer_browser_time_zone	Time-zone offset in minutes between UTC and the Cardholder browser local time. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. Required when browser_javascript_enabled=true.	Type: String Length: 5
customer_browser_user_agent	Exact content of the HTTP user-agent header.	Type: String Length: 2048

For Payment Institutions

The Payment Institution can fill the 3DS results for each transaction, if 3DS authentication is performed on their side.

Note: These parameters are not applicable for cases Deposit to PAN and Deposit to RPI.

Parameter Name	Description	Value
tds_authentication_result_type	Type of result. Possible value is: SIMPLE	Type: String Length: 6
tds_authentication_result_authentication_type	Authentication Type. Indicates the type of authentication method the Issuer will use to challenge the Cardholder, whether in the ARes message or what was used by the ACS when in the RReq message. Possible values are: • 01 = Static. • 02 = Dynamic. • 03 = OOB. • 04 = Decoupled. • 05-79 = Reserved for EMVCo future use (values invalid until defined by EMVCo). • 80-99 = Reserved for DS use.	Type: String Length: 2

continues on next page

Table 160 – continued from previous page

Parameter Name	Description	Value
tds_authentication_result_authentication_value	Authentication Value. Payment System-specific value provided by the ACS or the DS using an algorithm defined by Payment System. Authentication Value may be used to provide proof of authentication. A 20-byte value that has been Base64 encoded, giving a 28-byte result.	Type: String Length: 19-28
tds_authentication_result_transactionid	Transaction ID for 1.0.2 or dsTransID for 2.1.0/2.2.0.	Type: String Length: 19-36
tds_authentication_result_transactionstatus	Transaction Status. Indicates whether a transaction qualifies as an authenticated transaction or account verification. Possible values are: • Y = Authentication Verification Successful. • N = Not Authenticated/Account Not Verified, Transaction denied. • U = Authentication/Account Verification Could Not Be Performed, Technical or other problem, as indicated in ARes or RReq. • A = Attempts Processing Performed, Not Authenticated/Verified, but a proof of attempted authentication/verification is provided. • C = Challenge Required, Additional authentication is required using the CReq/CRes. • D = Challenge Required, Decoupled Authentication confirmed. • R = Authentication/ Account Verification Rejected, Issuer is rejecting.	Type: String Length: 1

continues on next page

Table 160 – continued from previous page

Parameter Name	Description	Value
tds_authentication_result_message_version	Message Version Number. Protocol version identifier This shall be the Protocol Version Number of the specification utilised by the system creating this message. The Message Version Number is set by the 3DS Server which originates the protocol with the AReq message. The Message Version Number does not change during a 3DS transaction. Possible values are: • 1.0.2 • 2.1.0 • 2.2.0	Type: String Length: 5

Sender Customer Data

Parameters below can be required for specific integrations. For more information, please contact Support manager.

Parameter Name	Description	Value
sender_first_name	Sender's first name.	Type: String Length: 128
sender_last_name	Sender's last name.	Type: String Length: 128
sender_middle_name	Sender's middle name/patronym.	Type: String Length: 128
sender_ssn	Last four digits of the Sender's social security number.	Type: String Length: 11
sender_birth_place	Sender's birth place.	Type: String Length: 128
sender_birthday	Sender's birthday.	Type: String Length: 30

continues on next page

Table 161 – continued from previous page

Parameter Name	Description	Value
sender_address1	Sender's address.	Type: String Length: 256
sender_city	Sender's city.	Type: String Length: 128
sender_state	Sender's US states (two letter abbreviation). Not applicable outside the US.	Type: String Length: 4
sender_zip_code	Sender's zip code.	Type: String Length: 32
sender_citizenship	Sender's citizenship.	Type: String Length: 128
sender_country_code	Sender's country (two letter abbreviation).	Type: String Length: 2
sender_phone	Sender's full international phone number, including country suffix.	Type: String Length: 128
sender_cell_phone	Sender's full international cell phone number, including country suffix.	Type: String Length: 128
sender_email	Sender's email address.	Type: String Length: 128
sender_resident	Is sender a resident?	Type: Boolean
sender_identity_document_id	Sender's identity document name.	Type: String Length: 128
sender_identity_document_series	Sender's identity document series.	Type: String Length: 12

continues on next page

Table 161 – continued from previous page

Parameter Name	Description	Value
sender_identity_document_number	Sender's identity document number.	Type: String Length: 16
sender_identity_document_issuer_name	Sender's identity document issuer.	Type: String Length: 128
sender_identity_document_issuer_department_code	Sender's identity document issuer department code.	Type: String Length: 32
sender_identity_document_issue_date	Sender's identity document issue date.	Type: Date

Receiver Customer Data

Parameters below can be required for specific integrations. For more information, please contact Support manager.

Parameter Name	Description	Value
receiver_first_name	Receiver's first name.	Type: String Length: 128
receiver_last_name	Receiver's last name.	Type: String Length: 128
receiver_middle_name	Receiver's middle name/patronym.	Type: String Length: 128
receiver_birth_place	Receiver's birth place.	Type: String Length: 128
receiver_birthday	Receiver's birthday.	Type: String Length: 128
receiver_address1	Receiver's address.	Type: String Length: 256

continues on next page

Table 162 – continued from previous page

Parameter Name	Description	Value
receiver_city	Receiver's city.	Type: String Length: 128
receiver_state	Receiver's state.	Type: String Length: 2-4
receiver_zip_code	Receiver's zip code.	Type: String Length: 32
receiver_region	Receiver's region.	Type: String Length: 30
receiver_area	Receiver's area.	Type: String Length: 50
receiver_citizenship	Receiver's citizenship.	Type: String Length: 128
receiver_country_code	Receiver's country (two letter abbreviation).	Type: String Length: 2
receiver_phone	Receiver's full international phone number, including country suffix.	Type: String Length: 128
receiver_email	Receiver's email address.	Type: String Length: 128
receiver_resident	Is receiver a resident?	Type: Boolean
receiver_identity_document_id	Receiver's identity document name.	Type: String Length: 128
receiver_identity_document_series	Receiver's identity document series.	Type: String Length: 12

continues on next page

Table 162 – continued from previous page

Parameter Name	Description	Value
receiver_identity_document_number	Receiver's identity document number.	Type: String Length: 16
receiver_identity_document_issuer_name	Receiver's identity document issuer.	Type: String Length: 128
receiver_identity_document_issuer_department_code	Receiver's identity document issuer department code.	Type: String Length: 32
receiver_identity_document_issue_date	Receiver's identity document issue date.	Type: Date

7.43.5 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order ID.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.
end-point-id	Endpoint id used for the transaction.

7.43.6 Request & Response Examples

Transfer from PAN to PAN

Request Example

```
POST /paynet/api/v4/transfer/39509 HTTP/1.1
Host: sandbox.doc2.com
```

(continues on next page)

(continued from previous page)

```

User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↳ "aGDxlip0s21lzMdyp0TGmClTvbBgnaUh", oauth_signature=
  ↳ "R6uBD0TnC75UmYVFqEf70YX9tKK7iRmBR
  ↳ %2FoJGPCUTwzzyjtkAg6jK8fk2I0EbzeadqTLDgckrXShSsD5Ke8%2FP3ZqPwwh30MT5XK9MWVcDGavjKQi
  ↳ %2BnXfJdr4JRbhPAk1YL3IJL04TTTmAoJHFWu0EKNdzndAMQ0cWay4Hg5evPcDpJEK0a2ZaKnSmjh9UMmttvzDL4Cb2INDH
  ↳ %2F5VadKPji%2FbDaxxWVT%2FQ9HVJcVSDHlxFnILBgyJr4ZiG4R0wtn03hb0C
  ↳ %2FC5rf7USgjkbpkBUejqjk4Wgw%2BJHMFJ4PJQM0beYn1ctmVGg27M0p8UWyi2i3PsAy8FbPf
  ↳ %2Ff6sMTDhx8MABfbIoJHgrvke%2B3flfS%2BUmtCo5pQ%2F9blWvVmB%2BjJz%2BqznYEAUk
  ↳ %2BHj7ekSVzMvpf5LNDeIwpwSVEZpp5Ig80yKZcWe9i8bbmfDsWehSGpiVkarawsLCToras3lZy73Hp5z6SN8Uc98f
  ↳ %2BXUF8R0JvSysAfzDU2BJrDhNo1tKjuhY%2FEQzEinYe0zCSsSH0hkl2yU2KwWxbgWxb%2FoMEHIuuZ5l
  ↳ %2Baa%2FGTLCB%2Bm5wJ8n9QeDSulHM3VmTP%2F7wFXa5DwCQCDy3Lu7v0WVFhLirX0MH6V
  ↳ %2Fe5SZ1yQHo0uH2N7F72DGKrcNkvkLaZCB4NE8AVqISxB484AB5Mo0pC3iZzAPWNo14liezgrlacarwSNsqSwmGBfvCWakxsZ
  ↳ %2BRgt06SkLAgpPcGDYwJziDvT6uQvAQG3R5a5v60grjZnN0PQqsWb4qT7nlirHLwXXrSL46HTRYCl9q4QCuuFKPZA0LOW3y70
  ↳ %2BuI%2FZHXzD0eHGIhxBwa%2B
  ↳ %2BrpYhwDzmx1736gnNZ6dBbzKz5hH20BW0zvaZ93mKDH32Ri62D4yvAf64ElvSdNSImImLtEstRcws9YBsdCAXJw0TXICclwN
  ↳ %2FUSjihshg2NDmFJe4TNZ7TZRzLD5WKR6FxeTH6xYq1RJvoHiC0SUyVKjX7knFp%2FIEx3fdpMRD8ZCT
  ↳ %2FgUw5Cu34q87B7xVLHjEhn9%2FV3lacVDJ48F1aS6urv%2FI2cahSsIz288g1V1ThrusN7TmdIsoZ
  ↳ %2F7XqKxht3p34N0e7LDUVils42ySmZil78GVvr9wR2Gt0pa8FT31F0LnyVxCiA3Fqj5LNM3QunElqSanXI
  ↳ %2BtwlzKFGq4oklwb7bil0in0uzYlb4ymz7x6o708dJmCeUpkfCUxFrQnpmmHGBg0VgcKS5S769otXsXICqMP
  ↳ %2BDU2x5dajQvbIlzzm4aKF4qm2jxSowyd0WU1Y28Q%3D%3D", oauth_signature_method="RSA-
  ↳ SHA256", oauth_timestamp="1661493794", oauth_version="1.0"
Content-Length: 305
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=10.42
&card_printed_name=John%20Doe
&client_orderid=1
&credit_card_number=4332435659564446
&currency=USD
&cvv2=123
&deposit2card=false
&destination-card-no=4998406231653309
&expire_month=12
&expire_year=2099
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200

```

Success Response Example

```

HTTP/1.1 200 OK
Server: server
Date: Fri, 26 Aug 2022 06:07:43 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

```

(continues on next page)

(continued from previous page)

```

type=async-response
&serial-number=00000000-0000-0000-0000-000002ddaf96
&merchant-order-id=1
&paynet-order-id=6863038
&end-point-id=39509

```

Fail Response Example

```

HTTP/1.1 500 Internal Server Error
Server: server
Date: Fri, 14 Oct 2022 09:53:35 GMT
Content-Length: 61
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

Internal server error [1ee76b38-3764-407e-9dfb-2ffc9ab0775b]

```

Transfer from PAN to RPI

Request Example

```

POST /paynet/api/v4/transfer/39509 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  "yL2jqI06NnIVBGhT3ocsWDgsaYxG3P7j", oauth_signature="U5FwR96T0huNFmDQYivbEkLFISfZKF
  ↳%2FREN
  ↳%2BwBXLNAD8K0HAAXh0FbmXDSkgvsZXb4i1UPc1xFmqvIX2LYsiZRixMesIRIbAhoHsmHHpGfZ5hS2YMfVAUQWfVbHdqCJQZEJ
  ↳%2FpXERzDrr5N76A0hl2onBPjqXQqhAdVYBfTUHhezDJChhyg4Lc4gkRsVbw8tIbx220vCT57t9CQhYvNd9b0NH6Vkn6XyUVY3
  ↳%2BNt4CQ25W2F%2FpYVK2%2BtLeCVDHyTY7JGaUalWKXg2HIhr20lmh9GRSZ51tenAL%2FgkHI
  ↳%2BDsIh5nn6nsBAMFnSCQR7ANY7cUpLw5KZnQLKLFcCeNeBdFqXDwZ2a0X1PDdHvn1KZBtf1SqGp3kkEX3AMin
  ↳%2F3QT0HobLRFaWGBIEwGxOddRL2mPI0UMvF6CE%2FGgBGHstcjyDCAQ3U2n%2BXLam1cCgWtLL2A5nBM2
  ↳%2FMnCUqqYUY4eN2Gidh2ZoLw70v%2BL4jLnK3%2BKefRxyPg0LZQBxH1qfP3tayZLMIdJ8gAdAmx2RLK
  ↳%2B3krUsy4TSpCzyAX4xloovGgC9yP0u0u4jJT6DTSdTJ3IdiCoUTyAA%2BPdkRrMz%2BswFRH
  ↳%2FCU8HETu0GQHUCnbL10DjHCvvis30tNH6j1aNETfWJqJ29lSngs8%2BU0Q
  ↳%2FLAGdlRXdAVKqj5Vd5wfLeMvSgjmhpB56Cfk60hXS%2Bj58DPdLKZ8Itg7gZ0%2BW
  ↳%2BGNfpwkD2baTdAiMjU%2F0ZGx9UW9caT1QrsXL6ub4NXQHnHluCqHH%2Fi6zN0u2pAuHcuJ
  ↳%2BOL0zcpSDRCFe0k2zy2SAiWKTKEFypMg2AQuRydrB9ZFpnmVokPnN4Y99ZSdj1PD3V0uXCW1h7AjGub8oMYCaq0vcB8bdSg
  ↳%2BJYnkWS5CMsQSMjuW5ZhRctGYlRpq3luQS25h5LLXi30rqkiQdst0I9lTmmzwQ00iFmXPcH7nKE8gantsa5hs7BVgPeypY
  ↳%2BKfVZiqs9w6y2aq%2F9CUtatVj9%2BU56
  ↳%2FucDhs2b9s6kY3kbv03p568hlJHvNxgyhv0eGfMSyWjmi130uZf0VsNlbptGC69nIlXKXgV7sX3B049gN
  ↳%2Biw08EP8tftpZPE%2BFViY8ecDrs2CquzJpZv0FX4WZ2bbIr9
  ↳%2B00cVMhGJ7QrFC96ddnbhfGaDJsc2FCgJBQJvJ8Mhxa7kksHujdxFrBa
  ↳%2BKGss0qR0WtCpxRtb4zBDEcq%2FHKlBkB0T6l0WpYvkhUy1coF8KtuhPSfqhkvG6gB5%2B
  ↳%2BLgGK6DQKI0MI9MvzSLTOR9ElE9vWUUMPlEg2b46jq0iIuAwQvJZfnxqlizF50cqBbJazBj7t
  ↳%2FiH0KKPv06Q%3D%3D", oauth_signature_method="RSA-SHA256", oauth_timestamp=
  ↳"1661494949", oauth_version="1.0"
Content-Length: 303
Content-Type: application/x-www-form-urlencoded
Connection: close

```

(continues on next page)

(continued from previous page)

```
amount=10.42
&card_printed_name=John%20Doe
&destination_card_recurring_payment_id=1491828
&client_orderid=1
&currency=USD
&cvv2=123
&deposit2card=false
&credit_card_number=4998406231653309
&expire_month=12
&expire_year=2099
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200
```

Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Fri, 26 Aug 2022 06:22:43 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=async-response
&serial-number=00000000-0000-0000-0000-000002ddafb1
&merchant-order-id=1
&paynet-order-id=6863042
&end-point-id=39509
```

Fail Response Example

```
HTTP/1.1 500 Internal Server Error
Server: server
Date: Fri, 14 Oct 2022 12:30:15 GMT
Content-Length: 61
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

Internal server error [18819567-ab17-4342-86eb-f8a45966e8c8]
```

Transfer from RPI to PAN

Request Example

```
POST /paynet/api/v4/transfer/39509 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↪ "J5wfdvnj45sDN2TELY2nIHcpDgIPsEa5", oauth_signature=
  ↪ "VhvXrMZNTTrRxBCd7A0tRWcq67KLyD6wQpJoVuJP0I4
  ↪ %2BWseXd5bAfMZtuy1QBhVGMlGiVfL97tmiB1LzSTnWcA
  ↪ %2BL58bCyEXgoE0XL4rn3U2rXV2JWgPTRv1EHCaAt%2FJhZ1JPd6kzGG3ZDJXAZg3YMoRdL00iK66Z1AK
  ↪ %2BhuL1M9mziEkcvZZR3eLREmz%2F223aVXRr%2BXc6dWFTHQm9BDTsVarsHrYHv9TkBjDbYXp%2F
  ↪ %2BX9ltfdUV7YXGemd6jy25CbJC9zbTj1wHsrpT2bkbBCpych8xtANFAeQgyTpR4IU40glzPSunXeZa09o1
  ↪ %2FwzWEvDmHN4g0WRnyl%2Bmgy7qu3wvMyyDf27UCRXFgEmfybk%2Fa3bwouJUqQbYUMif%2FGULsPmdSmz
  ↪ %2FWhmPnyIq9lvNkExE4xIXB1j7C8aYGoSRavvdhmEcNIbyXEs6hur
  ↪ %2FmXkIJH5RHDJTAcfu7RHkRUd3jBQjEEI1q9HnhTmdHK0EovQs%2BzR3K7Fv6
  ↪ %2FLW71cmbMMpqYbkIaBVN0
  ↪ %2Fdx4jLwZwqbK3B6PxxAyf6f48vJDFoNqVrLpaGXWTwZaeLWLE0sCgk3Cscss64%2B
  ↪ %2Bt8eHaVRS2u09uwlh9iCSqXeZjikHB782u7xc6f8PwP70xK0DsBsN1QyN1
  ↪ %2FLMH1ozr1HsYvZrucC8C7T2BykksEKmitpZ0cRUsj7Hb59nh39xk%2B8VklVenfUyz70
  ↪ %2FnMs0euKna0RHLI35DqgnvPp0yjU2SqM32N6%2B7Dtaa135N7oC1z8hG1zCB2C%2F%2Bk9Bmt0y
  ↪ %2Fad6yiEMN%2F%2FslGob7U5x82Sf7ph92EFCPu1ccW%2FMbxb9xu8csaMHDbh7E%2FydQm3W0ym0U3
  ↪ %2FPSckea23Y0TScrr0mQsG0jBwzgehUsJyCyLiWehFob6aEfuphXfdiYb5vXUKZuTAWPofgusJByr0HERhfpQoUJ96YXfGGF
  ↪ %2FPfZxTCQxjm%2FI00kSY48PTjYVq36V7x2v
  ↪ %2FDzJeux1Txu4TpewUod1ZDhxzu8kUEgKgxtJaLpCFuZL8dwTynb4LCoKEK8mM15MRXWpHJzJ4k9yH1Rrn8bfEcQWUI5bdgsE
  ↪ %2B2ZM05XihaKA1eCxBtKtXlvNShKjqS5fYdUfRHf5yVRwE67EZx9zAj1lHvTPSB1q1foWbU5MhN9vNc36hKLLsCjiKZsRctGF
  ↪ %2BYD%2BQsMiiEIPmlHLWf2Z1w8g7NK%2BigKPI13hyLe0avPDEzV1ngy1XqkIdQdLS1mxxpD8sw9
  ↪ %2F0h4iN7kDgRC3eSwM97%2B0%2FdaN0d14JQ%3D%3D", oauth_signature_method="RSA-SHA256",
  ↪ oauth_timestamp="1661496079", oauth_version="1.0"
Content-Length: 314
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=10.42
&client_orderid=1
&destination-card-no=4998406231653309
&currency=USD
&cvv2=123
&deposit2card=false
&card_recurring_payment_id=1491828
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200
```

Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Fri, 26 Aug 2022 06:41:34 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
```

(continues on next page)

(continued from previous page)

```
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=async-response
&serial-number=00000000-0000-0000-0000-000002ddafb6
&merchant-order-id=1
&paynet-order-id=6863043
&end-point-id=39509
```

Fail Response Example

```
HTTP/1.1 500 Internal Server Error
Server: server
Date: Fri, 14 Oct 2022 12:34:35 GMT
Content-Length: 61
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

Internal server error [ea7cab67-2af5-408a-aa76-79bba38688cf]
```

Transfer from RPI to RPI

Request Example

```
POST /paynet/api/v4/transfer/39509 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↪ "AC4qZJfR40ZW8xiNX9SrWj5upjPZ74kU", oauth_signature="mF%2BUtmLP7hbIMLfHCbuxDiq
  ↪ %2F3nx6l04zuWzpkVdnw030HiyNBGdEg2rIsg1SZAg3Eu8VUXGBepYK0wksnyIgcYtigP5apTLrz%2FgpE4o
  ↪ %2BHtP%2FfJ%2B04qQ4fV5WH0RqPjd9xBEaD2J0%2Foe6xYg3tx4u1QZ1MyrQjcJTrpbn%2B0xnia8
  ↪ %2FFruu4Crs3KHLmiUBSSVc9T9YlDu%2BuZfDBCxSqHs1j3T4JoCskfH8VHJ
  ↪ %2FI8igkh5e9qUYD4MrVunhsbcv8KHS0N%2Fb1f5b%2FcoH0QK9ERX7Smo7Ns9ErvbGjJtu%2BwFMffH8
  ↪ %2B4EextzJbRKQRMdy6PxNiu66wzGuMVKCRdTvc
  ↪ %2FisUnJ0LssFRwnFLamxrytoPj2I45rPkh08ydA2jYCAErR8FYAmk%2BiLrY9pLkAL21BgKLHw
  ↪ %2BJgxUeQAIqLwWHIaNA8AnzP9nvYGu9gRVYfei%2FZ%2Bue%2BqXBAldJIGYB0JTGHTef8t2l0piI0AEIM
  ↪ %2Fd%2FKF5%2BbPp52Xdde6aKV1FT2A5i8%2FCJEKxuh
  ↪ %2FMWwcSqa88T7m4swnzH879cnkqbgzmSl2PS49poiccDaUwC78ZCwGVD73FUSsgBn068kZVpYwBIat543EH1HCvG1sH1xvF
  ↪ %2B
  ↪ %2BMn3L1PPss2G2qai5ITLunNfyFXzofotvR7KrpRM3EuZmB47eanQ0nxmpyvBI2Bsv0b08bJ97EPSxjhIyjdYoTq8F2GEumMs
  ↪ %2Fd7iSjvjfxHLA%2FIQjew%2BguK%2BTL%2Bw1lPCDq9monQ8EX70T1kh91DVjlcxglqxpWt
  ↪ %2BWCY56ZXuEKZ6kqYjT6N2ab2bjGTiBrcgmToq7gvcbkDMCqgzPJZR5tK%2Fc%2Fu2hep02L0%2BU
  ↪ %2FzG9IB55iFog2c2qG32T%2Bg8K54WGPAYzubkW3mKy9B8VwLzUu%2FWSKNQgtbf2pqV0LUZ
  ↪ %2FTCalood9CqfNF%2B2ECO8x1ubyyzbBaUIW86fA1cgXptgmJp6G5xapky877ND9XXBRlXv0S6
  ↪ %2BlcItiqB%2FfQEI57awflpyRlGQQJttGrJTqjYXiFbvmLA1Z98l8yZ2aZmDi%2BfxcGkICtnjUYZnlYU
  ↪ %2B07Eo%2F1V%2BrcbQMqZxP6zYHd7hPTwuNZ0fS
  ↪ %2FGAzd1bfVXkBTszqB8PZuuQEs50bBfwjGbJGKhY8KuFgcR7B6lhamzfhokCMGcx0MJP6YY9ZWqU4ba9WT04ZU53pgEJmUTuR
  ↪ %2FgT1GlzlKQJN80gZ1Wb9t13yVRva%2FTRn
  ↪ %2FOXzfBhPIrxJv0KfXLUx8qHDweBwo5U2r8CG0N4xw5GJTgs2ncDBNXjWSjLE3dhG9G1louVP3ydryCQ%3D
  ↪ %3D", oauth_signature_method="RSA-SHA256", oauth_timestamp="1661496711", oauth_
  ↪ version="1.0"
```

(continues on next page)

(continued from previous page)

```
Content-Length: 312
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=10.42
&card_recurring_payment_id=1491830
&client_orderid=1
&currency=USD
&cvv2=123
&deposit2card=false
&destination_card_recurring_payment_id=1491829
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200
```

Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Fri, 26 Aug 2022 06:52:10 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=async-response
&serial-number=000000000-0000-0000-0000-000002ddafbb
&merchant-order-id=1
&paynet-order-id=6863044
&end-point-id=39509
```

Fail Response Example

```
HTTP/1.1 500 Internal Server Error
Server: server
Date: Fri, 14 Oct 2022 12:38:47 GMT
Content-Length: 61
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

Internal server error [317a3dcd-a96e-4b3a-801c-c5cfe7d05dad]
```


Transfer from Deposit to PAN

Request Example

```
POST /paynet/api/v4/transfer/39509 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↪ "Fw108eTrgovCuHhmHJ35utckB11MkfwD", oauth_signature="DRmRSQPQPw
  ↪ %2BkDcpNi0boL98MDQzvH730fM%2F0ylUFoUbm165GsbkhT0TJhFD%2BjpSLiEXnzinRaiDtTc
  ↪ %2FoEDLVuCF7EjNfIsKT6xLEkp39xf08yrr2QWeqeAu6mpmwzq7%2BVNVobo0UCzQXMCZbkZ0h
  ↪ %2B4US2pbJbZ0QrZFwGo%2Fd73weyecFBe4Z3ua4Q94MvMUKioH%2FD%2FvjFvLB6Mf8ua6b3%2Fmo5
  ↪ %2BiRwXykic1mwUBf7TB6MXgpRcu9eET3BbGc8%2FS0Xn9Lv2KE%2FoUYN4SmE
  ↪ %2BWgZXE3WR5DALkVrKYzX05p0guqsy3IAp%2Blu%2F1XaU%2BxqxcLAF%2B4KBGLrQt2RINSe
  ↪ %2FgCHTh9qwTiNJcCURdVZ%2Bid6jqGWo4IswEwRgCAHv0XbsoZTj2qW6YqpCm7U1v58FSbLhMLQy
  ↪ %2Fp6cpArnLGU1KDRjVo4nzFnsgIJaPwlim2PuM13X1ueaPsb7wEPWdJbpzs5cLViI7pkPSXSVWCAPrsCRu
  ↪ %2B25%2FzSqrz%2FPw04%2F3dhbkP9CDw9BM0AtonBJzHJtFDKsM2x0U56qv1xaJgEYGUVa0wyy
  ↪ %2FZn0rde4kPX2tkaBF6fFzZkoaMLI1CqBWBqHvWTXZ1vgWe6NIQmFve9NPjA69
  ↪ %2FEhT02SREA8R2GkS0hF1UBtc1cRg8z6dY4jXbqr7UfwSsu%2BI3JB7
  ↪ %2BDTLME0VSpfTEA8pbuqHwQL9YGV2BYPUTkVTMIfuetz2byaNafv9TsdV1QBi5uqbjT
  ↪ %2FTXeZ3214zL7rIU1UpqAHXkAcEraZDQV3h5X0i37yTmC0vMkfBl
  ↪ %2B0DnhZbSHkgCLd5aqDc4Hm0pvTfDFa92PcqAq0rK4TMERKQKujMJFucoWP3fjRS7dQFsBLpWTbtKoNkycALHtd
  ↪ %2BI0PIzglrpJo09AoHmbQJGZd40XouBnkjo0KjjJqP4ZRGdtfMH4ADSj%2FyC1w00aWcZGNkNEABS
  ↪ %2BRVpGD9VC0rYoe695Xt9XV4UEGygCREJ92WqhbnaqfrGVu6eXCeIuV7KaUH7aLps5YzIySpDeJLKQTU5J7TuCzf0niLh5C2t
  ↪ %2Bl0h0BnqHX5FsU6RzQnPF9QkU0i1A2K
  ↪ %2B1nfqYOHUFq8wJNo4BIQwosUo7FzYRVzgsXqMnF5gQ7NYMkwxCaUz96XNTDj1vlzQWSyo3AgS6Jb5oIy5yr
  ↪ %2B%2FJPDzEchf370t7kUAtmE3fLZRh
  ↪ %2BcrUd2jZ7tRHxcndQiTodbt0pjht2qv4WcbXqPaWIWURBB80xdH7QELxPyG%2BUUvPaiUGscDZ76aJnEk
  ↪ %2BT5jBC2Hak8%2B9Zy4qB%2Ft2QCTPwoAkuC2oHUafzaA8JzytjJLVQpTFvLh7Lj
  ↪ %2BAWg7KZGqKfYX1tmDsYUrdBRb9PfgmCPZ6owEkBDPQ%3D%3D", oauth_signature_method="RSA-
  ↪ SHA256", oauth_timestamp="1662384499", oauth_version="1.0"
Content-Length: 197
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=10.42
&client_orderid=1
&currency=USD
&deposit2card=true
&destination-card-no=4402746838361941
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200
```

Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 05 Sep 2022 13:28:35 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
```

(continues on next page)

(continued from previous page)

```

Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=async-response
&serial-number=00000000-0000-0000-0000-000002ddb02e
&merchant-order-id=1
&paynet-order-id=6863078
&end-point-id=39509

```

Fail Response Example

```

HTTP/1.1 500 Internal Server Error
Server: server
Date: Fri, 14 Oct 2022 12:58:56 GMT
Content-Length: 61
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

Internal server error [3fc83866-7aac-4c92-b6e9-828069e5a9d2]

```

Transfer from Deposit to RPI

Request Example

```

POST /paynet/api/v4/transfer/39509 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↪ "F2ab2fvpltWloWmn7xV1coohcnHXAR0c", oauth_signature="V5ufFoDRY
  ↪ %2BMR8pUP6PgCOHg2M8t53EepdzSvqzzMdQVQ3oEoyPW0TPQK1daviExQ%2FF
  ↪ %2Ft1Lc77n1RKopD0SXbm2UBXe
  ↪ %2FE9RNDERH04blb01bKCTm9N2IAUTW5PWKcre0LYuiCt2nh5cwY2hFbvxF15EPoUk06AIoA9wFmrNEICBQJtDhtj7bULmAZhq
  ↪ %2FXDLZ0%2Fg2jJ8kaamcL3EMewZcIwyB1F0iKUBniwX%2BXattNeXbniXTBqPG
  ↪ %2B653RpqEMG3YwAIbAobxLk
  ↪ %2B2q6plW5EDApY8rXXv61WBzVJ2pfBakWHUGslB15SynzzhE14EFDt6eL4dl22qm
  ↪ %2FM6i7fJGea2d5zFboVgZ0uIXCh0gtoN8gJa3mMht2isfRzuCg8K22hkgNy3CMYeUh7TFYxgsAr14pBXKjMS6MgMvXnPsxWoM
  ↪ %2FcrECPfnIcjk8HvT%2Bsux4ytps%2BKrbBfXnq%2Bjg%2BtT7LjR1mQnvaWw5HRMDlzlC%2Bsm04J
  ↪ %2BLJQBmpKknFNBv3oxGejhzZTtVcEGddCMNeh4KXUCMucJdb1I8ssq03yFS%2FPrDZbbdK
  ↪ %2Fy7VonLxWg0j0zw6l46M9YLvJE9clUBvhgpguG6ClMecpgoQvDMnR%2BIx3zl1qy6E8eDyaXJ
  ↪ %2Bh4opohJrSRGDiTcVtGRbcZUr%2FWyLDUR625LjavHv9zT1aGGieSTsNM
  ↪ %2BeLgxsSzRlyXR4E4zjqDAB3J9rwKKbZyx8gTduMhGydUGDapNwKVhC5hb0F
  ↪ %2ByYjwiz2M05Qr40mdeBfejdAJx4%2ButBsxx0l0ucz1IjR8L8NAn5LVrmuf33nLT63J8mTgXU
  ↪ %2F1bnQQHvmnTA4nX2NkpXbUDHfry8H%2F0oyCbs6dChX4RrJ0QkS49tjxzJuiFIxAsUHkrldS9cHuluG5
  ↪ %2FV%2Fp6IwYHcWhUG2Pxw6PJAcuin32sM66p%2Ff63CAcu46V9cIyoXpeSBvYZM8jt6Guowy5uHDIImUFLi
  ↪ %2FHCVopWLS%2FRFyY2iJUmqKypGh7Po
  ↪ %2FJyVhAYdpXrpaLTEM0d40pLU00GaxjaPTmcKuDgmXDQEFmC00zAT6
  ↪ %2F57tvZ38bhqSbReM5dWZC6mog9iC2wjY4gpDf
  ↪ %2FUTX7gPGmQkefTysZ84QdRdoDNi0BDDfBD5l0PSPzMs67obwXgnsrJ0dlGNRyEjQ
  ↪ %2B8hZgxXH2mWh2wX4skHq43kxaJJ2bdIVab9BNtVj5YjnUJxJWJoe0a0
  ↪ %2B7slFQwfxIm8lxKwtJdEgK6r0lU0Lg%3D%3D", oauth_signature_method="RSA-SHA256", oauth_
  ↪ timestamp="1662384669", oauth_version="1.0"

```

(continues on next page)

(continued from previous page)

```
Content-Length: 206
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=10.42
&client_orderid=1
&currency=USD
&deposit2card=true
&destination_card_recurring_payment_id=1491830
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200
```

Success Response Example

```
HTTP/1.1 200 OK
Server: server
Date: Mon, 05 Sep 2022 13:31:42 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 137

type=async-response
&serial-number=00000000-0000-0000-0000-000002ddb02f
&merchant-order-id=1
&paynet-order-id=6863079
&end-point-id=39509
```

Fail Response Example

```
HTTP/1.1 500 Internal Server Error
Server: server
Date: Fri, 14 Oct 2022 12:55:09 GMT
Content-Length: 61
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

Internal server error [51a48bc5-a061-40c5-b88b-f2c084024b77]
```

7.43.7 Postman Collection

7.43.8 Request Builder

Enter your private key in PKCS#1 container to use debug. See [RSA-SHA256](#) for details.

7.44 /api/v4/transfer-form

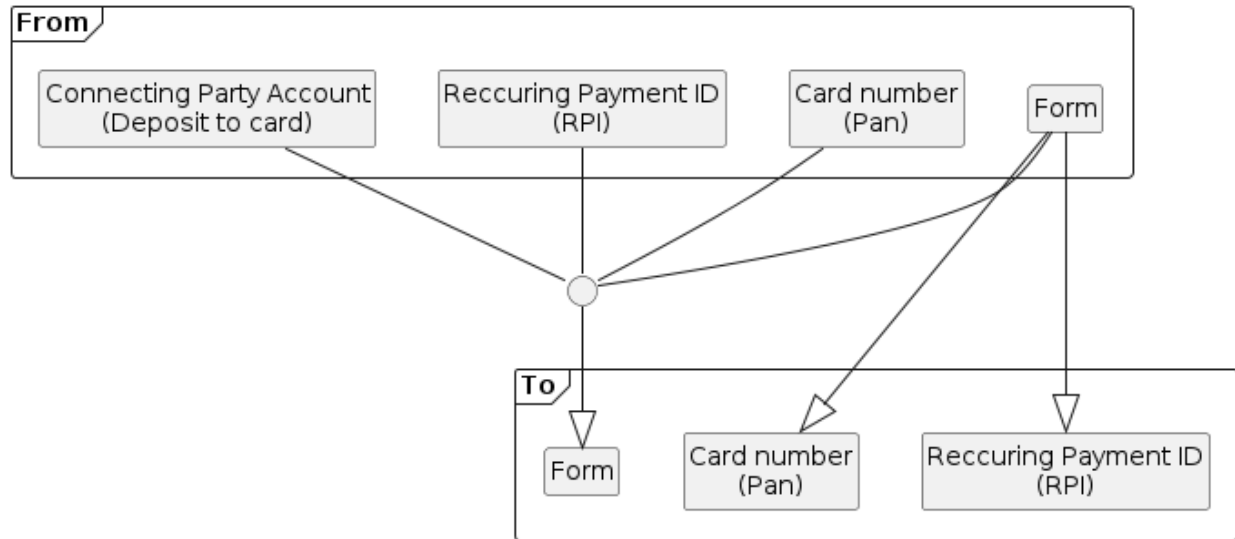
- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Additional Parameters](#)
 - [Sender Customer Data](#)
 - [Receiver Customer Data](#)
- [Response Parameters](#)
- [Request & Response Examples](#)
 - [Transfer from Form to PAN](#)
 - [Transfer from Form to RPI](#)
 - [Transfer from PAN to Form](#)
 - [Transfer from RPI to Form](#)
 - [Transfer from Form to Form](#)
 - [Transfer from Deposit to Form](#)
- [Postman Collection](#)
- [Request Builder](#)

7.44.1 Introduction

Transfer is a type of complex transaction that allows to move funds between bank cards (Primary Account Number - PAN), card tokens (Recurring Payment ID - RPI) and Connecting Party bank account (deposits).

Transfer Form integration is relevant for Connecting Party who are not able to accept Connecting Party order ID. card details (Connecting Party's website must complete PCI DSS certification). Using Transfer Form integration Connecting Party is released of accepting payment details. In this case the accepting and processing of payment details is carried out on the Doc2.0 side (if PAN is not sent). This type of integration also allows the Sender (for transfers between cards) or Receiver (for deposit to card) to submit their PAN on Doc2.0 side.

Transfer is initiated through HTTPS POST request to the [URL](#) with the [required parameters](#) depending on a certain transfer Use-Case (see below). Use [RSA-SHA256](#) for authentication. Transfer Form Use-Cases diagram:



Form to PAN	Transfer from Form to PAN occurs when transferring funds from payment details indicated in form to bank card for which its number is indicated.
Form to RPI	Transfer from Form to RPI occurs when funds are transferred from payment details indicated in form to bank card for which its recurring payment id is indicated.
PAN to Form	Transfer from RPI to Form occurs when funds are transferred from one bank card for which its number is indicated to payment details indicated in form.
RPI to Form	Transfer from RPI to Form occurs when funds are transferred from one bank card for which its recurring payment id is indicated to payment details indicated in form.
Form to Form	Transfer from Form to Form occurs when funds are transferred from payment details indicated in form to another payment details indicated in form.
Deposit to Form	Transfer from deposit to Form occurs when funds are transferred from Connecting Party account to payment details indicated in form.

7.44.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/transfer-form/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/transfer-form/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v4/transfer-form/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v4/transfer-form/group/ENDPOINTGROUPID

7.44.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Form to PAN

Parameter	Description	Value
destination-card-no, alias destination	Receiver`s card PAN.	Necessity: Required Type: Numeric Length: 19
deposit2card	Marker of deposit to card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: False Type: Boolean
ipaddress	The Sender or Receiver`s IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.100.101	Necessity: Required Type: String Length: 45
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Order description.	Necessity: Required Type: String Length: 64k

continues on next page

Table 166 – continued from previous page

Parameter	Description	Value
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
first_name	Sender or Receiver's first name.	Necessity: Optional Type: String Length: 50
last_name	Sender or Receiver's last name.	Necessity: Optional Type: String Length: 50
ssn	Last four digits of the Sender or Receivers's social security number.	Necessity: Optional Type: Numeric Length: 4
birthday	Sender or Receiver's birthday.	Necessity: Optional Type: Numeric Length: 8
address1	Sender or Receiver's address. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Optional Type: String Length: 256
city	Sender or Receiver's city.	Necessity: Optional Type: String Length: 50
state	Sender or Receiver's state (two-letter state code). Please see Mandatory State Codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Optional Type: String Length: 2

continues on next page

Table 166 – continued from previous page

Parameter	Description	Value
zip_code	Sender or Receiver's zip code.	Necessity: Optional Type: String Length: 10
country	Sender or Receiver's country (two letter abbreviation).	Necessity: Optional Type: String Length: 2
phone	Sender or Receiver's full international phone number, including country suffix.	Necessity: Optional Type: String Length: 15
cell_phone	Sender or Receiver's full international cell phone number, including country suffix.	Necessity: Optional Type: String Length: 15
purpose	Destination to where the payment goes. It is useful for the merchants who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999;mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128
email	Sender or Receiver's email address.	Necessity: Optional Type: String Length: 50
site_url	URL the original transfer is made from.	Necessity: Optional Type: String Length: 128

continues on next page

Table 166 – continued from previous page

Parameter	Description	Value
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined. Optional for direct integration (non-form) deposit2card. This parameter must be used if there is neither redirect_success_url nor redirect_fail_url.	Necessity: Optional Type: String Length: 1024
redirect_success_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is approved. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024
redirect_fail_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is declined or filtered. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024
preferred_language	Sender or Receiver's two-letter language code for multi-language transfer forms.	Necessity: Optional Type: String Length: 2

continues on next page

Table 166 – continued from previous page

Parameter	Description	Value
merchant_form_data	Parameters sent in merchant_form_data API parameter are parsed into macros with the same name, the parameter is url-encoded, example test-param%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters [a-zA-Z0-9], parameter value characters [a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 166 – continued from previous page

Parameter	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

Form to RPI

Parameter	Description	Value
destination_card_recurring_payment_id	Receiver's recurring payment ID. To create destination_card_recurring_payment_id please see Card Reference ID Creation.	Necessity: Required Type: Long
deposit2card	Marker of deposit to card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: False Type: Boolean
ipaddress	The Sender or Receiver's IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.1.101	Necessity: Required Type: String Length: 45
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10

continues on next page

Table 167 – continued from previous page

Parameter	Description	Value
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Order description.	Necessity: Required Type: String Length: 64k
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
first_name	Sender or Receiver's first name.	Necessity: Optional Type: String Length: 50
last_name	Sender or Receiver's last name.	Necessity: Optional Type: String Length: 50
ssn	Last four digits of the Sender or Receivers's social security number.	Necessity: Optional Type: Numeric Length: 4
birthday	Sender or Receiver's birthday.	Necessity: Optional Type: Numeric Length: 8
address1	Sender or Receiver's address. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Optional Type: String Length: 256

continues on next page

Table 167 – continued from previous page

Parameter	Description	Value
city	Sender or Receiver's city.	Necessity: Optional Type: String Length: 50
state	Sender or Receiver's state (two-letter state code). Please see Mandatory State Codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Optional Type: String Length: 2
zip_code	Sender or Receiver's zip code.	Necessity: Optional Type: String Length: 10
country	Sender or Receiver's country (two letter abbreviation).	Necessity: Optional Type: String Length: 2
phone	Sender or Receiver's full international phone number, including country suffix.	Necessity: Optional Type: String Length: 15
cell_phone	Sender or Receiver's full international cell phone number, including country suffix.	Necessity: Optional Type: String Length: 15
purpose	Destination to where the payment goes. It is useful for the merchants who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999;mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128

continues on next page

Table 167 – continued from previous page

Parameter	Description	Value
email	Sender or Receiver's email address.	Necessity: Optional Type: String Length: 50
site_url	URL the original transfer is made from.	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined. Optional for direct integration (non-form) deposit2card. This parameter must be used if there is neither redirect_success_url nor redirect_fail_url.	Necessity: Optional Type: String Length: 1024
redirect_success_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is approved. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 167 – continued from previous page

Parameter	Description	Value
redirect_fail_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is declined or filtered. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024
preferred_language	Sender or Receiver's two-letter language code for multi-language transfer forms.	Necessity: Optional Type: String Length: 2
merchant_form_data	Parameters sent in merchant_form_data API parameter are parsed into macros with the same name, the parameter is url-encoded, example testparam%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters [a-zA-Z0-9], parameter value characters [a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128

continues on next page

Table 167 – continued from previous page

Parameter	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

PAN to Form

Parameter	Description	Value
credit_card_number	Sender`s credit card number.	Necessity: Required Type: Numeric Length: 19
cvv2	Sender`s CVV2 code. CVV2 (Card verification value) is the three of four digit number farthest to the right on the flip side of a credit card.	Necessity: Required Type: String Length: 3-4
expire_month	Sender credit card`s month of expiration.	Necessity: Required Type: String Length: 2
expire_year	Sender credit card`s year of expiration.	Necessity: Required Type: String Length: 2-4
card_printed_name	Sender`s card printed name.	Necessity: Required Type: String Length: 128
deposit2card	Marker of deposit to card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: False Type: Boolean
ipaddress	The Sender or Receiver`s IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.1.101	Necessity: Required Type: String Length: 45
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10

continues on next page

Table 168 – continued from previous page

Parameter	Description	Value
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Order description.	Necessity: Required Type: String Length: 64k
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
first_name	Sender or Receiver's first name.	Necessity: Optional Type: String Length: 50
last_name	Sender or Receiver's last name.	Necessity: Optional Type: String Length: 50
ssn	Last four digits of the Sender or Receiver's social security number.	Necessity: Optional Type: Numeric Length: 4
birthday	Sender or Receiver's birthday.	Necessity: Optional Type: Numeric Length: 8
address1	Sender or Receiver's address. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Optional Type: String Length: 256

continues on next page

Table 168 – continued from previous page

Parameter	Description	Value
city	Sender or Receiver's city.	Necessity: Optional Type: String Length: 50
state	Sender or Receiver's state (two-letter state code). Please see Mandatory State Codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Optional Type: String Length: 2
zip_code	Sender or Receiver's zip code.	Necessity: Optional Type: String Length: 10
country	Sender or Receiver's country (two letter abbreviation).	Necessity: Optional Type: String Length: 2
phone	Sender or Receiver's full international phone number, including country suffix.	Necessity: Optional Type: String Length: 15
cell_phone	Sender or Receiver's full international cell phone number, including country suffix.	Necessity: Optional Type: String Length: 15
purpose	Destination to where the payment goes. It is useful for the merchants who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999;mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128

continues on next page

Table 168 – continued from previous page

Parameter	Description	Value
email	Sender or Receiver's email address.	Necessity: Optional Type: String Length: 50
site_url	URL the original transfer is made from.	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined. Optional for direct integration (non-form) deposit2card. This parameter must be used if there is neither redirect_success_url nor redirect_fail_url.	Necessity: Optional Type: String Length: 1024
redirect_success_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is approved. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 168 – continued from previous page

Parameter	Description	Value
redirect_fail_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is declined or filtered. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024
preferred_language	Sender or Receiver's two-letter language code for multi-language transfer forms.	Necessity: Optional Type: String Length: 2
merchant_form_data	Parameters sent in merchant_form_data API parameter are parsed into macros with the same name, the parameter is url-encoded, example testparam%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters [a-zA-Z0-9], parameter value characters [a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128

continues on next page

Table 168 – continued from previous page

Parameter	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

RPI to Form

Parameter	Description	Value
cvv2	Sender`s CVV2 code. CVV2 (Card verification value) is the three of four digit number farthest to the right on the flip side of a credit card.	Necessity: Required Type: String Length: 3-4
card_recurring_payment_id	Sender`s recurring payment ID. To create card_recurring_payment_id please see Card Reference ID Creation.	Necessity: Required Type: Long
deposit2card	Marker of deposit to card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: False Type: Boolean
ipaddress	The Sender or Receiver`s IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.1.1	Necessity: Required Type: String Length: 45
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Order description.	Necessity: Required Type: String Length: 64k
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128

continues on next page

Table 169 – continued from previous page

Parameter	Description	Value
first_name	Sender or Receiver's first name.	Necessity: Optional Type: String Length: 50
last_name	Sender or Receiver's last name.	Necessity: Optional Type: String Length: 50
ssn	Last four digits of the Sender or Receiver's social security number.	Necessity: Optional Type: Numeric Length: 4
birthday	Sender or Receiver's birthday.	Necessity: Optional Type: Numeric Length: 8
address1	Sender or Receiver's address. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Optional Type: String Length: 256
city	Sender or Receiver's city.	Necessity: Optional Type: String Length: 50
state	Sender or Receiver's state (two-letter state code). Please see Mandatory State Codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Optional Type: String Length: 2
zip_code	Sender or Receiver's zip code.	Necessity: Optional Type: String Length: 10

continues on next page

Table 169 – continued from previous page

Parameter	Description	Value
country	Sender or Receiver's country (two letter abbreviation).	Necessity: Optional Type: String Length: 2
phone	Sender or Receiver's full international phone number, including country suffix.	Necessity: Optional Type: String Length: 15
cell_phone	Sender or Receiver's full international cell phone number, including country suffix.	Necessity: Optional Type: String Length: 15
purpose	Destination to where the payment goes. It is useful for the merchants who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999;mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128
email	Sender or Receiver's email address.	Necessity: Optional Type: String Length: 50
site_url	URL the original transfer is made from.	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined. Optional for direct integration (non-form) deposit2card. This parameter must be used if there is neither redirect_success_url nor redirect_fail_url.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 169 – continued from previous page

Parameter	Description	Value
redirect_success_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is approved. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024
redirect_fail_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is declined or filtered. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024
preferred_language	Sender or Receiver's two-letter language code for multi-language transfer forms.	Necessity: Optional Type: String Length: 2

continues on next page

Table 169 – continued from previous page

Parameter	Description	Value
merchant_form_data	Parameters sent in merchant_form_data API parameter are parsed into macros with the same name, the parameter is url-encoded, example testparam%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters [a-zA-Z0-9], parameter value characters [a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 169 – continued from previous page

Parameter	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

Form to Form

Parameter	Description	Value
deposit2card	Marker of deposit to card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: False Type: Boolean
ipaddress	The Sender or Receiver's IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.1.101	Necessity: Required Type: String Length: 45
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10

continues on next page

Table 170 – continued from previous page

Parameter	Description	Value
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Order description.	Necessity: Required Type: String Length: 64k
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
first_name	Sender or Receiver's first name.	Necessity: Optional Type: String Length: 50
last_name	Sender or Receiver's last name.	Necessity: Optional Type: String Length: 50
ssn	Last four digits of the Sender or Receiver's social security number.	Necessity: Optional Type: Numeric Length: 4
birthday	Sender or Receiver's birthday.	Necessity: Optional Type: Numeric Length: 8
address1	Sender or Receiver's address. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Optional Type: String Length: 256

continues on next page

Table 170 – continued from previous page

Parameter	Description	Value
city	Sender or Receiver's city.	Necessity: Optional Type: String Length: 50
state	Sender or Receiver's state (two-letter state code). Please see Mandatory State Codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Optional Type: String Length: 2
zip_code	Sender or Receiver's zip code.	Necessity: Optional Type: String Length: 10
country	Sender or Receiver's country (two letter abbreviation).	Necessity: Optional Type: String Length: 2
phone	Sender or Receiver's full international phone number, including country suffix.	Necessity: Optional Type: String Length: 15
cell_phone	Sender or Receiver's full international cell phone number, including country suffix.	Necessity: Optional Type: String Length: 15
purpose	Destination to where the payment goes. It is useful for the merchants who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999;mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128

continues on next page

Table 170 – continued from previous page

Parameter	Description	Value
email	Sender or Receiver's email address.	Necessity: Optional Type: String Length: 50
site_url	URL the original transfer is made from.	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined. Optional for direct integration (non-form) deposit2card. This parameter must be used if there is neither redirect_success_url nor redirect_fail_url.	Necessity: Optional Type: String Length: 1024
redirect_success_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is approved. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 170 – continued from previous page

Parameter	Description	Value
redirect_fail_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is declined or filtered. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024
preferred_language	Sender or Receiver's two-letter language code for multi-language transfer forms.	Necessity: Optional Type: String Length: 2
merchant_form_data	Parameters sent in merchant_form_data API parameter are parsed into macros with the same name, the parameter is url-encoded, example testparam%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters [a-zA-Z0-9], parameter value characters [a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128

continues on next page

Table 170 – continued from previous page

Parameter	Description	Value
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

Deposit to Form

Parameter	Description	Value
deposit2card	Marker of deposit to card transfer. If true, no sender cardholder data is needed to process the transaction. If false, sender cardholder data is needed.	Value: True Type: Boolean
ipaddress	The Sender or Receiver's IP address, include for fraud screening purposes. NB: 45 is for IPv4 tunneling like 0000:0000:0000:0000:0000:0000:192.168.1.100.191	Necessity: Required Type: String Length: 45
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar, EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Order description.	Necessity: Required Type: String Length: 64k
client_orderid	Connecting Party order ID.	Necessity: Required Type: String Length: 128
first_name	Sender or Receiver's first name.	Necessity: Optional Type: String Length: 50
last_name	Sender or Receiver's last name.	Necessity: Optional Type: String Length: 50

continues on next page

Table 171 – continued from previous page

Parameter	Description	Value
ssn	Last four digits of the Sender or Receiver's social security number.	Necessity: Optional Type: Numeric Length: 4
birthday	Sender or Receiver's birthday.	Necessity: Optional Type: Numeric Length: 8
address1	Sender or Receiver's address. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Optional Type: String Length: 256
city	Sender or Receiver's city.	Necessity: Optional Type: String Length: 50
state	Sender or Receiver's state (two-letter state code). Please see Mandatory State Codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Optional Type: String Length: 2
zip_code	Sender or Receiver's zip code.	Necessity: Optional Type: String Length: 10
country	Sender or Receiver's country (two letter abbreviation).	Necessity: Optional Type: String Length: 2
phone	Sender or Receiver's full international phone number, including country suffix.	Necessity: Optional Type: String Length: 15

continues on next page

Table 171 – continued from previous page

Parameter	Description	Value
cell_phone	Sender or Receiver's full international cell phone number, including country suffix.	Necessity: Optional Type: String Length: 15
purpose	Destination to where the payment goes. It is useful for the merchants who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999;mail@example.com etc. This value will be used by fraud monitoring system	Necessity: Optional Type: String Length: 128
email	Sender or Receiver's email address.	Necessity: Optional Type: String Length: 50
site_url	URL the original transfer is made from.	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined. Optional for direct integration (non-form) deposit2card. This parameter must be used if there is neither redirect_success_url nor redirect_fail_url.	Necessity: Optional Type: String Length: 1024

continues on next page

Table 171 – continued from previous page

Parameter	Description	Value
redirect_success_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is approved. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024
redirect_fail_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is declined or filtered. This parameter should not be used to retrieve results from Doc2.0 gateway, because all parameters go through client's browser and can be lost during transmission. To deliver the correct payment result to the backend server_callback_url must be used instead. gate.doc2.com should be passed if non-3DS schema for transactions processing is used and there is no need to redirect Sender or Receiver anywhere. This parameter must be used if there is no redirect_url.	Necessity: Optional Type: String Length: 1024
preferred_language	Sender or Receiver's two-letter language code for multi-language transfer forms.	Necessity: Optional Type: String Length: 2

continues on next page

Table 171 – continued from previous page

Parameter	Description	Value
merchant_form_data	Parameters sent in merchant_form_data API parameter are parsed into macros with the same name, the parameter is url-encoded, example testparam%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters [a-zA-Z0-9], parameter value characters [a-zA-Z0-9], control characters [= &], 2MB max size. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 1024

continues on next page

Table 171 – continued from previous page

Parameter	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 1024

7.44.4 Additional Parameters

Sender Customer Data

Parameters below can be required for specific integrations. For more information, please contact Support manager.

Parameter Name	Description	Value
sender_first_name	Sender's first name.	Type: String Length: 128
sender_last_name	Sender's last name.	Type: String Length: 128
sender_middle_name	Sender's middle name/patronym.	Type: String Length: 128
sender_ssn	Last four digits of the Sender's social security number.	Type: String Length: 11

continues on next page

Table 172 – continued from previous page

Parameter Name	Description	Value
sender_birth_place	Sender`s birth place.	Type: String Length: 128
sender_birthday	Sender`s birthday.	Type: String Length: 30
sender_address1	Sender`s address.	Type: String Length: 256
sender_city	Sender`s city.	Type: String Length: 128
sender_state	Sender`s US states (two letter abbreviation). Not applicable outside the US.	Type: String Length: 4
sender_zip_code	Sender`s zip code.	Type: String Length: 32
sender_citizenship	Sender`s citizenship.	Type: String Length: 128
sender_country_code	Sender`s country (two letter abbreviation).	Type: String Length: 2
sender_phone	Sender`s full international phone number, including country suffix.	Type: String Length: 128
sender_cell_phone	Sender`s full international cell phone number, including country suffix.	Type: String Length: 128
sender_email	Sender`s email address.	Type: String Length: 128
sender_resident	Is sender a resident?	Type: Boolean

continues on next page

Table 172 – continued from previous page

Parameter Name	Description	Value
sender_identity_document_id	Sender`s identity document name.	Type: String Length: 128
sender_identity_document_series	Sender`s identity document series.	Type: String Length: 12
sender_identity_document_number	Sender`s identity document number.	Type: String Length: 16
sender_identity_document_issuer_name	Sender`s identity document issuer.	Type: String Length: 128
sender_identity_document_issuer_department_code	Sender`s identity document issuer department code.	Type: String Length: 32
sender_identity_document_issue_date	Sender`s identity document issue date.	Type: Date

Receiver Customer Data

Parameters below can be required for specific integrations. For more information, please contact Support manager.

Parameter Name	Description	Value
receiver_first_name	Receiver`s first name.	Type: String Length: 128
receiver_last_name	Receiver`s last name.	Type: String Length: 128
receiver_middle_name	Receiver`s middle name/patronym.	Type: String Length: 128
receiver_birth_place	Receiver`s birth place.	Type: String Length: 128

continues on next page

Table 173 – continued from previous page

Parameter Name	Description	Value
receiver_birthday	Receiver's birthday.	Type: String Length: 128
receiver_address1	Receiver's address.	Type: String Length: 256
receiver_city	Receiver's city.	Type: String Length: 128
receiver_state	Receiver's state.	Type: String Length: 2-4
receiver_zip_code	Receiver's zip code.	Type: String Length: 32
receiver_region	Receiver's region.	Type: String Length: 30
receiver_area	Receiver's area.	Type: String Length: 50
receiver_citizenship	Receiver's citizenship.	Type: String Length: 128
receiver_country_code	Receiver's country (two letter abbreviation).	Type: String Length: 2
receiver_phone	Receiver's full international phone number, including country suffix.	Type: String Length: 128
receiver_email	Receiver's email address.	Type: String Length: 128
receiver_resident	Is receiver a resident?	Type: Boolean

continues on next page

Table 173 – continued from previous page

Parameter Name	Description	Value
receiver_identity_document_id	Receiver`s identity document name.	Type: String Length: 128
receiver_identity_document_series	Receiver`s identity document series.	Type: String Length: 12
receiver_identity_document_number	Receiver`s identity document number.	Type: String Length: 16
receiver_identity_document_issuer_name	Receiver`s identity document issuer.	Type: String Length: 128
receiver_identity_document_issuer_department_code	Receiver`s identity document issuer department code.	Type: String Length: 32
receiver_identity_document_issue_date	Receiver`s identity document issue date.	Type: Date

7.44.5 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Parameter	Description.
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, ex:error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order ID.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

continues on next page

Table 174 – continued from previous page

Parameter	Description.
redirect-url	The URL to the page where the Connecting Party should redirect the client's browser. Connecting Party should send HTTP 302 redirect, see General Transfer Form Process Flow.

7.44.6 Request & Response Examples

Transfer from Form to PAN

Request Example

```

POST /paynet/api/v4/transfer-form/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↳ "eZCXwEp9f017vGFwPUDStKj0KIP9u0nu", oauth_signature=
  ↳ "2qrf7DY0fiy88doBrIGafZBqLLX12ePYWDGLqNxBfU9T3RzwsYtEj92EPuiJuInZxbPY6av4F2jT1JS02D4KqPATadsEYb8A
  ↳ %2Fbh0JJBN3aMvKJ%2FnsVRChY%2FFpRhdufdBxZLKpAXYfJQgk42%2FB9pu
  ↳ %2FsBSRWJwTnVZxfavHYMWmrY2rvq2x1nedSJq60%2Fxfz90U3CvH0lK%2FGfdI4WfvpHIXpw1Mlz
  ↳ %2F6AnQRdMmYHnoDsFu79GI57NeoBRsZazJBYh%2FE%2FFcS9sHU5BF0yvLQYz
  ↳ %2FCXtb7lDbKeDJTHDDWakyyZPQc0QReIMHpNwNr9mjC%2FucIJitFEldsWWGaOnTc2Tqtrdjzzwukm
  ↳ %2BP7Bi5siJ5Pl%2BxWq0MAEFpNCXfThBKIJrxQXbBwR4YfPaBb70nePybCH8JCX0vakoah
  ↳ %2BwFJfR0WNg0BFYxQfUYNcEXDmrk8hLbwz%2FXJ6bfanFaE8N30wijJmlzY
  ↳ %2B4ZrEqBjnlQcBnNuvuxKSb6dKtnRo5UVXEszEiCxEvWQifggohkU1CTHZFpA2qmrjIcgJoD0hQPv6Lk92LiDtVSE2drxZ
  ↳ %2BjuI69ZzGlnKlM0lVCnFfdahrhtMM9GYNN%2B24YaKd5UkCMIMru%2Bf0fvqwrJu00
  ↳ %2Fims8VXwsm9aP8pP4Lr3I%3D", oauth_signature_method="RSA-SHA256", oauth_timestamp=
  ↳ "1670421183", oauth_version="1.0"
Content-Length: 303
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=100
&client_orderid=34T43R77N
&currency=USD
&deposit2card=false
&destination-card-no=4268736646656312
&destination_card_printed_name=Test%20Test
&destination_expire_month=12
&destination_expire_year=2023
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200

```

Success Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 07 Dec 2022 13:56:03 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding

```

(continues on next page)

(continued from previous page)

```

X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 285

type=async-form-response
&serial-number=00000000-0000-0000-0000-000002ddbfa2
&merchant-order-id=34T43R77N
&paynet-order-id=6863810
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
↪%2FBB587546567A31587163597A684535634A775969614A577953626976766D4B494A66796F55434370393032673D

```

Fail Response Example

```

HTTP/1.1 403
Server: server
Date: Mon, 12 Dec 2022 08:25:15 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>403</title>
  <style type="text/css">
    body {
      font-family: Arial, sans-serif;
      font-size: 130%;
      background-color: #eee;
    }
    p {
      margin: 10em auto 0;
      width: 500px;
      border: 1px solid gray;
      text-align: center;
      vertical-align: middle;
      padding: 40px 20px;
      background-color: #fff;
      -webkit-border-radius: 20px;
      -moz-border-radius: 20px;
      border-radius: 20px;
    }
  </style>
</head>
<body>
  <p>Access is denied</p>
</body>
</html>

```

Transfer from Form to RPI

Request Example

```
POST /paynet/api/v4/transfer-form/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↳ "yQFnx1Zerg4pH28VmLQL3Hwc7dQJlAqg", oauth_signature=
  ↳ "G2ZRjH0xzhsQ5xIurkM7Bo7REnoBfG3HIDDzwG9XfFGZdFC0815
  ↳ %2FPVS4ESdLPYVer7iT2K3Ltl02mhMzmIWuRV3YTVKKCbfbuJkPk10wJ0x66ZaX4q0DI5Zej%2FadXiKuJ
  ↳ %2FLe%2BLL8Tp%2Ftzrylk6fiTAoX2AT471hp02xezERVvdEwkhOWB1mqwYE4Vx0L8wDk0W0oqR
  ↳ %2Bk1XgWXPcj3E8WV3kEVFwxYwgi6daQyimqReEc5q8fqbul7sANmIozrSRWqjG2K
  ↳ %2FBLs0W2UFLMcvWdHi20N%2B%2BnTzaLCExB2991ozSE2jPp4eahkd69Zz
  ↳ %2BEwLENVIn8KDwvA5j25XaGgbqLKK%2BBt5Csh0bkvZZZ%2FuWqujp%2Fzb0BGMuucuwYXe4yW8Ter1YC
  ↳ %2F9zoR%2BMwIckBd5gaV4m0xQCZ5pe9GMD2805BLM2XDtcB9UT%2FeMs3MR3WIXN51EVCn
  ↳ %2BcazxwEI4LmZ5966aaUGu4GF41JWRQAZBchx4F%2Bd%2BRadwCceBJJSiQg82hwRk4QYnyDEx
  ↳ %2BWqcwErVZGCH8yJhph1Etn3e0G8z0PLBLG2hqTqGPfYivzyFHI1LNW8jl7%2BKJq2HZgXjtcU8Ji
  ↳ %2BVw2JVr86nnM4nPbpIRL7qr%2BSYUpXjWw%2B3e%2F6Gt0L%2B0WzVFMo5jKf
  ↳ %2FfyJSd0gbycolsBHplZRZ8Gsts8UGiIuQ8%3D", oauth_signature_method="RSA-SHA256",
  ↳ oauth_timestamp="1670847388", oauth_version="1.0"
Content-Length: 213
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=100
&client_orderid=34T43R77N
&currency=USD
&deposit2card=false
&destination_card_recurring_payment_id=1491854
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200
```

Success Response Example

```
HTTP/1.1 200
Server: server
Date: Mon, 12 Dec 2022 12:17:14 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 285

type=async-form-response
&serial-number=00000000-0000-0000-0000-000002ddc2d7
&merchant-order-id=34T43R77N
&paynet-order-id=6863948
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
  ↳ %2FBB587546567A31587163597A684535634A775969614A66354B455A714F6F32684D61414E78767241354872343D
```

Fail Response Example

```

HTTP/1.1 200
Server: server
Date: Mon, 12 Dec 2022 12:08:14 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 132

type=validation-error
&serial-number=00000000-0000-0000-0000-000002ddc2b8
&error-message=Recurring+payment+not+found
&error-code=122

```

Transfer from PAN to Form

Request Example

```

POST /paynet/api/v4/transfer-form/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↳ "BpGsiUR8CbIjGAD1Qxu8MTtGHnlIvzcF", oauth_signature=
  ↳ "ozovWc0wjugTlGfrS8TeSxqQtnSEQ7UrGh7z6B5px5BoKgTWF44Ej6VCAxCWmL
  ↳ %2B1Qxg5iZ5ExOqTWZ2GTx70ly%2B%2BgHkSmz3t02PEr9XJsJ2
  ↳ %2F0m4XFoF0LnbVPDje7AmZZBuF2l0iandlUBjJVA
  ↳ %2BarD6dWVv8LNrv3RVyheuhBreAr7dLcSD4Iu44MH76t0ZS
  ↳ %2BuMnqfJkWC37owAdC7xtJw0IF8wrFM272qHnzxbfvZRaJruoG6W0%2F%2BMGJLPnaNbrhfJ99
  ↳ %2BtvCybb0CsVMQqJ
  ↳ %2FsUWM8Z89ci6a4MCPE41v1i9lXyvkZDm6qfoVgKVPuHVVsjbvuu5Ub7MeMgBs176ZrRsi9yLiHmPLQGqhzxydy14GpCNH9FF
  ↳ %2F%2F7z6lLE3S0%2Bgca6NkuhfLU7lvAl0CguRAvU3
  ↳ %2F1Vh3R1p8C9S3BXQak6eM0xYp7h01wQKXMq4uMM2FjmtTUYe6ZiqyJGbKTrrIEkD%2Fo
  ↳ %2FX1NXxylMbWxuNyoTnQXx%2BQzDsUrY3Mhtb%2FslwCGDY4pu0lwiVqGg31EBz0wFWzJFX7n2uQ076g5q
  ↳ %2FnTJqf3dq3AdTC6xpaGC2kek0fxc6FEDdaoVeQ3PwJWYi1kcPbwkcvo4gIi03rc03JUJZF3TfomkFs
  ↳ %2BFnh%2FwLJHs%3D", oauth_signature_method="RSA-SHA256", oauth_timestamp="1670569138
  ↳ ", oauth_version="1.0"
Content-Length: 275
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=100
&card_printed_name=Test%20Test
&client_orderid=34T43R77N
&credit_card_number=4047753518764320
&currency=USD
&cvv2=123
&deposit2card=false
&expire_month=12
&expire_year=2023

```

(continues on next page)

(continued from previous page)

```
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200
```

Success Response Example

```
HTTP/1.1 200
Server: server
Date: Fri, 09 Dec 2022 06:59:20 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 285

type=async-form-response
&serial-number=00000000-0000-0000-0000-000002ddc004
&merchant-order-id=34T43R77N
&paynet-order-id=6863822
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
↪%2FBB587546567A31587163597A684535634A775969614A61365A5A5637456C4545743867705438697A512F30633D
```

Fail Response Example

```
HTTP/1.1 403
Server: server
Date: Fri, 09 Dec 2022 06:45:14 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>403</title>
  <style type="text/css">
    body {
      font-family: Arial, sans-serif;
      font-size: 130%;
      background-color: #eee;
    }
    p {
      margin: 10em auto 0;
      width: 500px;
      border: 1px solid gray;
      text-align: center;
      vertical-align: middle;
```

(continues on next page)

(continued from previous page)

```

padding: 40px 20px;
background-color: #fff;
-webkit-border-radius: 20px;
-moz-border-radius: 20px;
border-radius: 20px;
}
</style>
</head>
<body>
  <p>Access is denied</p>
</body>
</html>

```

Transfer from RPI to Form

Request Example

```

POST /paynet/api/v4/transfer-form/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↳ "y03UAVaLM9NxEwAHD0R9YDCQQFzRWx8b", oauth_signature="Zgx05s2b
  ↳ %2Fr4wSbo0sb2tGXbhPeDYlfJfnziSH57mkYfEMNtGoWCeULanQLKq1ddusIBhGUFwpwoh5f22y9tqHRsekiAQ92s9ixQxmV
  ↳ %2FHoTyktrxn0vhPxuwSmwD%2FhX%2F0MJgDaCo0ZK%2BxeXfPRVRwlNM8LsDd29i
  ↳ %2FnIUsexWCBUjs3M5ThYGz58UJdZm0xrWw3poH2Lb6Fw9b6XMwV2DzvyomhMzZfyQYYX%2BhP22rCmXTg
  ↳ %2BEyTVnJDBiWI3pF6WYQFjd05tPbck4bxw4IYNIKKEi78dw0%2FZcb65gfHlaUtqHd3ZXiFiob4YFeV
  ↳ %2B1XKI2wSVBVemEd3jVj0FXDZv2GLzrZ0Ukxcns6BqkRxMqaWc9
  ↳ %2FbpsL65XdFxB4LUuP2h1PVk6A5yjjanSjp1XqzyAx8EGpxYGAzIPNSsYPVN9x5TuyIqtYVXhz2y0z8RPx5VFmSYr
  ↳ %2BaCr6ruhRNkyonNmk47oPkIkG9PpLivBaq8PnIzUNpExTRIYJXJ0wD69CjCPxT
  ↳ %2Fhr3udDQJD4ZHFE3TNA9g1H0VRL
  ↳ %2Ff60jPhZeanf9MgcFqz6jckPQghBix0aTBf9Dmp9jmv12IQZ3PR3J3n53nDEUIw%2Fhiga3
  ↳ %2BdZswkJMYS57VB282SVfDXQP0%2F400C%2B3annev%2BMftiSfChhRgazDRT0CnbETXqoKc%3D",
  ↳ oauth_signature_method="RSA-SHA256", oauth_timestamp="1670850681", oauth_version="1.
  ↳ 0"
Content-Length: 266
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=100
&card_recurring_payment_id=1491854
&client_orderid=34T43R77N
&currency=USD
&cvv2=123
&deposit2card=false
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200

```

Success Response Example

```

HTTP/1.1 200
Server: server
Date: Mon, 12 Dec 2022 13:12:21 GMT

```

(continues on next page)

(continued from previous page)

```

Content-Type: text/html;charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 285

type=async-form-response
&serial-number=00000000-0000-0000-0000-000002ddc34e
&merchant-order-id=34T43R77N
&paynet-order-id=6863959
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
%2FBB587546567A31587163597A684535634A775969614A5A716755336669466574615739747A737059777A50493D

```

Fail Response Example

```

HTTP/1.1 200
Server: server
Date: Mon, 12 Dec 2022 12:25:05 GMT
Content-Type: text/html;charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 176

type=validation-error
&serial-number=00000000-0000-0000-0000-000002ddc2ec
&merchant-order-id=34T43R77N
&error-message=Src+card+cvv2+should+be+filled+in+transfer.
&error-code=29

```

Transfer from Form to Form

Request Example

```

POST /paynet/api/v4/transfer-form/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
"3FUX1QCRBPF7am99U04bty1qD7FK08I8", oauth_signature="CCvlgjfVBZ0woMS
%2FBUXv1LiRlc0Nu30wf3FwNmPnP49RWi3ve3VE9Jt61QpRx14pmCz3xFKQt
%2Bi4UvXJYlSn206TSZ9H7TGCSHhQQP5vD0mm3Pnq0YgtI2oPpn2PhgM%2FlgwwWcBjBqmCyzM
%2FGGqK24zn5C20bxvHL9LfuMgF4bEpIYDGXVh3%2BbhU5Z2H25fjjttZylm1SLhAi9l0ct2%2BtH00TJ
%2FgjdWR%2FCnIroLVPWqnYbE8f%2F
%2Bz1HuynqW684UUaAiB8WgEuLH1BBM6HqYiFAVyfwLLNzDv08cKZew7JJPss017XBdqYyE2QWkREM61FPSQpQsuAnRpezxjRZn
%2B%2BVeEEP7Z0IFqUreDgSYYSntrgtvZhhX6oNuEn4V8jWA6T00wPNPKqkEXTk7w53nF0Hn
%2FLgAa3FfYSHtCnFP1uELmPnVr003IWaPPw0%2BcCJzNMua7g%2FYjLCsdV3tlwUwne1R0V3togR5k

```

(continues on next page)

(continued from previous page)

```

↪%2FTnG%2Bub4txjaFeq0m9p517ppKZQn8Fe0jtIIJMCYVRVmsYkeBZyyRgcdIB0An09fCKZ7y74gI12py
↪%2FvVzagFCjZo05T20gY60pCkrP4HhWEkKJhrqSCBe7NLC1xZDszfkLFjF9Z4dWvk2PIcNfI3SGYZRks0YFPECycfYzYIJfuKp
↪%3D", oauth_signature_method="RSA-SHA256", oauth_timestamp="1670835177", oauth_
↪version="1.0"
Content-Length: 167
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=100
&client_orderid=34T43R77N
&currency=USD
&deposit2card=false
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200

```

Success Response Example

```

HTTP/1.1 200
Server: server
Date: Mon, 12 Dec 2022 08:53:26 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 285

type=async-form-response
&serial-number=00000000-0000-0000-0000-000002ddc1f8
&merchant-order-id=34T43R77N
&paynet-order-id=6863917
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
↪%2FBB587546567A31587163597A684535634A775969614A613368326D746C2B72645043314744502B537A6E57343D

```

Fail Response Example

```

HTTP/1.1 403
Server: server
Date: Mon, 12 Dec 2022 09:18:33 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>403</title>

```

(continues on next page)

(continued from previous page)

```

<style type="text/css">
  body {
    font-family: Arial, sans-serif;
    font-size: 130%;
    background-color: #eee;
  }
  p {
    margin: 10em auto 0;
    width: 500px;
    border: 1px solid gray;
    text-align: center;
    vertical-align: middle;
    padding: 40px 20px;
    background-color: #fff;
    -webkit-border-radius: 20px;
    -moz-border-radius: 20px;
    border-radius: 20px;
  }
</style>
</head>
<body>
  <p>Access is denied</p>
</body>
</html>

```

Transfer from Deposit to Form

Request Example

```

POST /paynet/api/v4/transfer-form/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↳ "SDrbaim155azbf2UthcVDrhwi0A6liE4", oauth_signature="CFUUC1Y6QT5Gj7LA%2FdW6gkGN
  ↳ %2Fh73J8HAXhvYtFx%2F%2FChlQIU7f4uvVmKp0ossrRD05wdx0ENHJZLZ8SI%2B610ygcWP%2B3K
  ↳ %2FqEYDI7qjdq8HgKLT3BnjSoGU6EjGjvcMwQSY0AwNJ4adYYZG6G%2Ble9S0vbPcQalkfxIjpTl
  ↳ %2FkyrIVtNS4h9ExkD%2Fg9UUXZ%2F2YA0CyhTy26QwOL0o5NmNeh%2FX7wpqPEyjHfEE
  ↳ %2FWbc3iSNSjlJA9b%2BfX5eFAVxzzZ2X76HRxy
  ↳ %2FuHqah9NsZgXsagMHPX0pvovNRLEh7NFBqVDWhoHFfDS%2Bwd6vcLqXxwLK0Vecmq06GN1
  ↳ %2BiZX4C9QebaNw6JSx72jMon3hnn37pUSLhzWXz%2BwlMX7XN
  ↳ %2FcTwsZ0kL51DzFXwPAJPVAeNevgxlaFtbELMGcZN0U19u72c0ZKMu7%2FsxLAfCgUITC02NhvZw
  ↳ %2BR8xUZUcxvwu4fvoTXr6rf8r0iloJIA9RoakobzmPUDcGx3
  ↳ %2F4Hswjbqk7Y0g6jxQJD6ZoKscHpLiQLvBWPIlsVMpQUpnUqzBNGexdCd8zaRzqiszWz4hEg6teTMFOLA
  ↳ %2FoQ8Wu%2FxtSs8JMuXsWtestY9h1y5KuXWDy809hpN45pBF%2BvtoDfyr6ktKrkZZTZG49FBKk
  ↳ %2FcKi5FjzL5Bsbwv9fgk1%2BRYDexMthCqX%2Fdrz%2BzMM%3D", oauth_signature_method="RSA-
  ↳ SHA256", oauth_timestamp="1670851416", oauth_version="1.0"
Content-Length: 166
Content-Type: application/x-www-form-urlencoded
Connection: close

amount=100
&client_orderid=34T43R77N
&currency=USD
&deposit2card=true

```

(continues on next page)

(continued from previous page)

```
&ipaddress=1.1.1.1
&order_desc=Your%20order%20description
&redirect_url=http%3A%2F%2Fwww.example.com
&server_callback_url=https://httpstat.us/200
```

Success Response Example

```
HTTP/1.1 200
Server: server
Date: Mon, 12 Dec 2022 13:23:58 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 285

type=async-form-response
&serial-number=00000000-0000-0000-0000-000002ddc35a
&merchant-order-id=34T43R77N
&paynet-order-id=6863960
&redirect-url=https%3A%2F%2Fsandbox.doc2.com%2Fpaynet%2Fform%2Finit
%2FBB587546567A31587163597A684535634A775969614A5878476D6E717A31324167486F61674C6830743070343D
```

Fail Response Example

```
HTTP/1.1 403
Server: server
Date: Mon, 12 Dec 2022 13:19:46 GMT
Content-Type: text/html
Content-Length: 735
Connection: close
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>403</title>
  <style type="text/css">
    body {
      font-family: Arial, sans-serif;
      font-size: 130%;
      background-color: #eee;
    }
    p {
      margin: 10em auto 0;
      width: 500px;
      border: 1px solid gray;
      text-align: center;
      vertical-align: middle;
```

(continues on next page)

(continued from previous page)

```
padding: 40px 20px;
background-color: #fff;
-webkit-border-radius: 20px;
-moz-border-radius: 20px;
border-radius: 20px;
}
</style>
</head>
<body>
  <p>Access is denied</p>
</body>
</html>
```

7.44.7 Postman Collection

7.44.8 Request Builder

Enter your private key in PKCS#1 container to use debug. See [RSA-SHA256](#) for details.

7.45 /api/v4/transfer-by-ref

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Request Parameters for Insurance](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.45.1 Introduction

Deposit to card transfer (D2C) is a money transfer from Connecting Party bank account (Deposit) and Receiver bank card number (PAN) or tokenized card data (Card Reference ID). Deposit to card is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth RSA-SHA256](#) for authentication.

7.45.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/v4/transfer-by-ref/ENDPOINTID	https://gate.doc2.com/paynet/api/v4/transfer-by-ref/ENDPOINTID
https://sandbox.doc2.com/paynet/api/v4/transfer-by-ref/group/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/v4/transfer-by-ref/group/ENDPOINTGROUPID

7.45.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameters marked as Conditional* can be mandatory for specific integrations. For more information, please contact your manager in Doc2.0.

Parameter Name	Description	Value
client_orderid	Connecting Party order identifier.	Necessity: Required Type: String Length: 128
login	Connecting Party's login. Must be used as oauth_consumer_key in OAuth Authorization, not included in request as login parameter.	Necessity: Required Type: String Length: 20
destination-card-no	Card number of destination card. Mandatory if destination-card-ref-id omitted. For the scenario of payment to a card inside the system, this card will be considered as a destination, and all processing limits, lists and fraud scoring will be applied to it as a destination card.	Necessity: Conditional Type: String Length: 16-19

continues on next page

Table 176 – continued from previous page

Parameter Name	Description	Value
destination-card-ref-id	Card reference id to destination card, obtained at Card Registration step. Mandatory if destination-card-no omitted. For the scenario of payment to a card inside the system, this card will be considered as a destination, and all processing limits, lists and fraud scoring will be applied to it as a destination card.	Necessity: Conditional Type: Numeric Length: 20
destination_expire_month	Expire month of destination card.	Necessity: Conditional Type: String Length: 1-2
destination_expire_year	Expire year of destination card.	Necessity: Conditional Type: String Length: 2-4
amount	Amount to be transferred. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents.	Necessity: Required Type: Numeric Length: 10
currency	Currency the transaction is charged in (three-letter currency code). Example of valid parameter values are: USD for US Dollar EUR for European Euro.	Necessity: Required Type: String Length: 3
order_desc	Order description.	Necessity: Required Type: String Length: 64k
receiver_identity_document_id	Receiver's identity document series (e.g. 4 digits for domestic passport).	Necessity: Optional Type: String Length: 512

continues on next page

Table 176 – continued from previous page

Parameter Name	Description	Value
re- ceiver_identity_document_id	Receiver's identity document number (e.g. 0123456789 for domestic passport).	Necessity: Optional Type: String Length: 512
re- ceiver_identity_document_id	Receiver's identity document id. Possible values: 21 for domestic passport or 31 for international passport.	Necessity: Optional Type: String Length: 512
receiver_address1	Receiver's address.	Necessity: Conditional* Type: String Length: 512
receiver_city	Receiver's city.	Necessity: Conditional* Type: String Length: 512
receiver_first_name	Receiver first name.	Necessity: Required Type: String Length: 128
re- ceiver_middle_name	Receiver middle name.	Necessity: Optional Type: String Length: 128
receiver_last_name	Receiver last name.	Necessity: Required Type: String Length: 128
receiver_phone	Receiver full international cell phone number, including country code.	Necessity: Optional Type: String Length: 128

continues on next page

Table 176 – continued from previous page

Parameter Name	Description	Value
receiver_resident	Is receiver a resident?	Necessity: Optional Type: Boolean Length: true/false
ipaddress	Customer's IP address, included for fraud screening purposes.	Necessity: Optional Type: String Length: 45
first_name	Sender first name.	Necessity: Optional Type: String Length: 128
middle_name	Sender middle name.	Necessity: Optional Type: String Length: 128
last_name	Sender last name.	Necessity: Optional Type: String Length: 128
ssn	Last four digits of the Sender's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Sender date of birth, in the format MMD-DYY.	Necessity: Optional Type: Numeric Length: 8
address1	Sender address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Optional Type: String Length: 256

continues on next page

Table 176 – continued from previous page

Parameter Name	Description	Value
city	Sender city.	Necessity: Optional Type: String Length: 50
state	Sender's state. Please see Appendix A for a list of valid state codes. Mandatory for USA, Canada and Australia.	Necessity: Optional Type: String Length: 2-3
zip_code	Sender ZIP code.	Necessity: Optional Type: String Length: 10
receiver_zip_code	Receiver ZIP code.	Necessity: Conditional* Type: String Length: 10
country	Sender country(two-letter country code). Please see Appendix B for a list of valid country codes.	Necessity: Optional Type: String Length: 2
re- ceiver_country_code	Receiver country(two-letter country code). Please see Appendix B for a list of valid country codes.	Necessity: Optional Type: String Length: 2
phone	Sender full international phone number, including country code.	Necessity: Optional Type: String Length: 15
cell_phone	Sender full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15

continues on next page

Table 176 – continued from previous page

Parameter Name	Description	Value
email	Sender email address.	Necessity: Optional Type: String Length: 50
purpose	Destination to where the payment goes. It is useful for the Connecting Parties who let their clients to transfer money from a credit card to some type of client's account, e.g. game or mobile phone account. Sample values are: +9999999999; mail@example.com etc. This value will be used by fraud monitoring system.	Necessity: Optional Type: String Length: 128
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters . This parameter can be sent instead of notify_url. If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc).	Necessity: Optional Type: String Length: 128

continues on next page

Table 176 – continued from previous page

Parameter Name	Description	Value
notify_url	URL, where the transaction status is sent to. Connecting Party may use notify URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to notify_url refer to Connecting Party callback parameters . This parameter can be sent instead of server_callback_url. If notify_url is sent, Payment Gateway sends callback notification once the original transaction receives final status, and about every future update for this original transaction (reversal, chargeback, etc). If server_callback_url is sent, Payment Gateway sends callback notification only when original transaction receives final status.	Necessity: Optional Type: String Length: 128
redirect_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined. See more details at Statuses .	Necessity: Optional Type: String Length: 250
redirect_success_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is approved.	Necessity: Optional Type: String Length: 1024
redirect_fail_url	URL the cardholder will be redirected to upon completion of the transaction. Please note that the cardholder will be redirected only in case if the transaction is declined or filtered.	Necessity: Optional Type: String Length: 1024
merchant_data	Any additional information for this transaction which may be useful in Connecting Party's external systems, e.g. VIP customer, TV promo campaign lead. Will be returned in Status response and Connecting Party Callback.	Necessity: Optional Type: String Length: 64k

7.45.4 Request Parameters for Insurance

Note:

The acquirer can override the mandatory nature of some fields to make them mandatory instead of optional.

Leading and trailing whitespace in input parameters will be trimmed.

Parameter Name	Description	Value
insurance_amount	The insurance amount. The amount must be specified in minimum units with a . delimiter. For example, 100.5 in USD means 100 US dollars and 50 cents.	Type: Numeric Length: 10
insured_person_first_name	Insured person's first name.	Type: String Length: 256
insured_person_last_name	Insured person's last name.	Type: String Length: 256
insured_person_middle_name	Insured person's middle name.	Type: String Length: 256
insured_person_birthday	Insured person's date of birth.	Type: String Length: 256
insured_person_document_series	Insured person's document series.	Type: String Length: 256

continues on next page

Table 177 – continued from previous page

Parameter Name	Description	Value
insured_person_document_number	Insured person's document number.	Type: String Length: 256
insured_person_document_issue_date	Date the insured person's document was issued.	Type: String Length: 256
insured_person_document_issuer_name	Name of the authority that issued the insured person's document.	Type: String Length: 256
insured_person_document_issuer_code	Code of the authority that issued the insured person's document.	Type: String Length: 256
insured_person_registration_address	Insured person's registration address.	Type: String Length: 256
insured_person_phone	Full international phone number of the insured person.	Type: String Length: 256
insured_person_email	Insured person's email address.	Type: String Length: 256
card_insurance_agreement_number	Agreement number.	Type: String Length: 256

continues on next page

Table 177 – continued from previous page

Parameter Name	Description	Value
card_insurance_agreement_sell_date	Sale date of the agreement.	Type: String Length: 256
card_insurance_agreement_start_date	Agreement start date.	Type: String Length: 256
card_insurance_agreement_end_date	Agreement end date.	Type: String Length: 256
card_insurance_agreement_amount	Insurance amount.	Type: String Length: 256
card_insurance_agreement_bonus	Insurance premium.	Type: String Length: 256

7.45.5 Response Parameters

Note:

Response has Content-Type: text/html;charset=utf-8 header. All fields are x-www-form-urlencoded, with (0xA) character at the end of each parameter's value.

Response parameter	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id.

continues on next page

Table 178 – continued from previous page

Response parameter	Description
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.45.6 Request Example

```

POST /paynet/api/v4/transfer-by-ref/39915 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth oauth_consumer_key="TestMerchant", oauth_nonce=
  ↳ "w1rmMzx2Sq2k5Hi7KAZmdhHHD6BygKcE", oauth_signature="rXn0DwL0laV%2Ft1q3s
  ↳ %2FJdLybJTdxKKN0XeFtyVNZ0U5bG0Qfcc6iyHqQG660hb852CYKjhrxvG3wf33RlIQd09f230Sg91JkdrUgl8QDrMqntC5myW
  ↳ %2BMjpV821fAY8AS%2BBByNUK0uClmwFgLYwr5YinbhjDa6P8aEdHgbf
  ↳ %2FVw5vg90qjwKxJhSXPGB5mIg7T9gTNXF4n4YTUJA0l
  ↳ %2BWyeGcIjuTSsiCUnbsrNjB0GYc0PLUSuHirwJ0NiKvDhK3qaZrGaEw8SZEQ%2F8x0naoaa0XUq
  ↳ %2FkppPfTJT%2FjTl%2FbGyX1x%2FwTl6EYgIElavHYj2psIGmFnyUk
  ↳ %2FYnaGZHike00rSf4IkG5Ye2M4hyl7lKQ0CXMZMvvGtfvLm0RsAod8o30KuDD5Tw
  ↳ %2BRmDiQNCqN8GeZawHsZipgwZuy2IdZ3sHiot3U6NEt10VH9TsWdu
  ↳ %2FstfBJOWBBStTRTTh4hn3Zvf5jLuTfTcepKz4CIgdQGGMmRSz6dcBnkYJaa7VRTh27dz
  ↳ %2BnEuC0laEYAytVuzQA43MvUXNLmAuh9JlmqQquAZwB%2BRFFLPg0j
  ↳ %2FVVsfSqX35UWBYQtWlWdEvc3jWBVxCjEW0s8cAcowlb4R2g4V48HC2Rz3ggfINzP9
  ↳ %2FWtXlknNnIlzQAeYbMREguYqFrK%2BYK6Lqyu%2BM1je0j9CWhhk%3D", oauth_signature_method=
  ↳ "RSA-SHA256", oauth_timestamp="1697442690", oauth_version="1.0"
Content-Length: 805
Content-Type: application/x-www-form-urlencoded
Connection: close

address1=10020Main%20st
&amount=100
&birthday=19820115
&cell_phone=%2B19023384543
&city=Seattle
&client_orderid=34T43R77N
&country=US
&currency=USD
&destination-card-ref-id=1461819
&card-exp-month=07
&card-exp-year=2028
&email=john.smith%40gmail.com
&first_name=John
&ipaddress=65.153.12.232
&last_name=Smith
&merchant_data=VIP%20customer
&middle_name=M
&order_desc=Test%20order%20Description
&phone=%2B12063582043
&purpose=user_account1
&receiver_address1=Red%20Sq%2C%201a
&receiver_city=Moscow

```

(continues on next page)

(continued from previous page)

```
&receiver_first_name=Jane
&receiver_identity_document_id=21
&receiver_identity_document_number=222222
&receiver_identity_document_series=1111
&receiver_last_name=Doe
&receiver_middle_name=L
&receiver_phone=%2B79031110022
&receiver_resident=true
&redirect_url=http%3A%2F%2Fwww.example.com%2F
&server_callback_url=https%3A%2F%2Fhttpstat.us%2F200
&ssn=1267&state=WA
&zip_code=98102
```

7.45.7 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Mon, 16 Oct 2023 13:20:25 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 145

type=async-response
&serial-number=000000000-0000-0000-0000-000002ee3a57
&merchant-order-id=34T43R77N
&paynet-order-id=7234671
&end-point-id=39915
```

7.45.8 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Mon, 16 Oct 2023 07:54:11 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-XSS-Protection: 1
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
Content-Language: en-US
Strict-Transport-Security: max-age=31536000
Content-Length: 162

type=validation-error
&serial-number=000000000-0000-0000-0000-000002ee3a4f
&merchant-order-id=34T43R77N
```

(continues on next page)

(continued from previous page)

```
&error-message=Rebill+1461819+was+not+found
&error-code=104
```

7.45.9 Postman Collection

7.45.10 Request Builder

Enter your private key in PKCS#1 container to use debug. See [RSA-SHA256](#) for details.

7.46 /api/mfo/scoring

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Signature computation](#)
- [Request Builder](#)

7.46.1 Introduction

Card scoring is initiated through HTTPS POST request to the [URLs](#) and the [parameters](#) specified below. Each request should have {client_order_ID}, generated by Connecting Party. {endpointId} in request URL is provided by Payment Gateway. Each request should be signed and the signature should be put to [X-Authorization](#) request header.

7.46.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/mfo/scoring/{client_order_id}/{endpoint_id}	https://api.doc2.com/paynet/api/mfo/scoring/{client_order_id}/{endpoint_id}

7.46.3 Request Parameters

Parameter name	Description	Format
first6PanDigits	First 6 digits of the card number. Send either cardRefId or cardNumber or a combination of first6PanDigits and last4PanDigits, not all.	N6
last4PanDigits	Last 4 digits of the card number. Send either cardRefId or cardNumber or a combination of first6PanDigits and last4PanDigits, not all.	N4
cardRefId	The card-ref-id (Card Reference ID) created earlier with /api/v2/create-card-ref/ request that can be used to find the specific card by its token. Send either cardRefId or cardNumber or a combination of first6PanDigits and last4PanDigits, not all.	N{1-*}
uniqueCardRefId	The unq-card-ref-id (Unique Card Reference ID) to each PAN created earlier with /api/v2/create-card-ref/ request that can be used by Connecting Party for loyalty programs or fraud control.	N{1-*}
cardNumber	Full card number. Send either cardNumber or cardRefId or a combination of first6PanDigits and last4PanDigits, not all.	N{13-19}
cardExpiryMonth	Card expiration month, if first6PanDigits and last4PanDigits are used (optional).	N{1-2}
cardExpiryYear	Card expiration year, if first6PanDigits and last4PanDigits are used (optional).	N4

7.46.4 Response Parameters

Parameter name	Description	Format
error	If error occurs, here you may see brief error description.	Type: String
bankBin	Card's bank identification number (As a rule, first 6 digits of the card number).	Type: Integer Example: 444444

continues on next page

Table 181 – continued from previous page

Parameter name	Description	Format
cardFound	This parameter shows if the system has card scoring information on this card or not.	Type: Boolean Example: true
countIssuedFor180Days	The number of transactions for the issuance of MFI loans for the last 180 days.	Type: Integer Example: 1
countIssuedFor30Days	The number of transactions for the issuance of MFI loans for the last 30 days.	Type: Integer Example: 1
countIssuedFor90Days	The number of transactions for the issuance of MFI loans for the last 90 days.	Type: Integer Example: 1
expiredMonth	Card expiration month.	Type: Integer Example: 12
expiredYear	Card expiration year.	Type: Integer Example: 2099
incomingTransferAmount-For30Days	Incoming transfer amount for 30 days.	Type: Decimal Example: 100.25
incomingTransferAmount-For365Days	Incoming transfer amount for 365 days.	Type: Decimal Example: 100.25
incomingTransferAmount-For60Days	Incoming transfer amount for 60 days.	Type: Decimal Example: 100.25
incomingTransferAmount-For90Days	Incoming transfer amount for 90 days.	Type: Decimal Example: 100.25
lastDischargeAmount	Amount of last repayment.	Type: Decimal Example: 111.400
lastDischargeDate	Date of last repayment.	Type: YYYY.MM.DD Example: 2015.07.24

continues on next page

Table 181 – continued from previous page

Parameter name	Description	Format
lastFourDigits	Last 4 digits of the card number.	Type: Integer Example: 1234
lastSuccessfulDis-chargeAmount	The amount of the last successful repayment.	Type: Decimal Example: 1.400
lastSuccessfulDischargeDate	The date of the last successful repayment.	Type: YYYY.MM.DD Example: 2015.07.24
mfoCountFor180Days	Number of unique MFIs conducting transactions in the last 180 days.	Type: Integer Example: 1
mfoCountFor30Days	Number of unique MFIs conducting transactions in the last 30 days.	Type: Integer Example: 1
mfoCountFor90Days	Number of unique MFIs conducting transactions in the last 90 days.	Type: Integer Example: 1
mfoIssuedFor180Days	Number of unique MFIs that issued loans in the last 180 days.	Type: Integer Example: 1
mfoIssuedFor30Days	Number of unique MFIs that issued loans in the last 30 days.	Type: Integer Example: 1
mfoIssuedFor90Days	Number of unique MFIs that issued loans in the last 90 days.	Type: Integer Example: 1
orderId	Number of the order generated in the system.	Type: Integer Example: 3623277
outgoingTransferAmount-For30Days	Outgoing transfer amount for 30 days.	Type: Decimal Example: 100.25
outgoingTransferAmount-For365Days	Outgoing transfer amount for 365 days.	Type: Decimal Example: 100.25

continues on next page

Table 181 – continued from previous page

Parameter name	Description	Format
outgoingTransferAmount-For60Days	Outgoing transfer amount for 60 days.	Type: Decimal Example: 100.25
outgoingTransferAmount-For90Days	Outgoing transfer amount for 90 days.	Type: Decimal Example: 100.25
totalDischargeAmount	Amount of all issued funds.	Type: Decimal Example: 15.410
totalIssuedAmount	Amount of all disbursed funds.	Type: Decimal Example: 10.420
totalRecurrentAmount	Amount of all forcibly written-off funds.	Type: Decimal Example: 9.000
transfersFromMFO	Indicator of MFI loans.	Type: true/false Example: true

7.46.5 Request example

Request with cardNumber parameter:

```
POST /paynet/api/mfo/scoring/3772/123456
HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.88.1
Accept: */*
X-Authorization: 3ea5d48e9354496c2576ef5ecee0324b2aabca6c
Content-Length: 27
Content-Type: application/x-www-form-urlencoded
Connection: close

cardNumber=4731543184191796
```

Request with cardRefId parameter:

```
POST /paynet/api/mfo/scoring/3772/123456 HTTP/2
Host: sandbox.doc2.com
User-Agent: curl/7.88.1
Accept: */*
X-Authorization: a3774b3911a833425f635e45474ecd9fe2714d3a
Content-Length: 15
Content-Type: application/x-www-form-urlencoded

cardRefId=12345
```

Request with first6PanDigits and last4PanDigits parameters (expiration date is optional):

```
POST /paynet/api/mfo/scoring/3772/123456
HTTP/2
Host: sandbox.doc2.com
User-Agent: curl/7.88.1
Accept: */*
X-Authorization: b548c006285fde742d608238be18fa0e5b559a08
Content-Length: 81
Content-Type: application/x-www-form-urlencoded

cardExpiryMonth=12&cardExpiryYear=2099&first6PanDigits=499939&last4PanDigits=5721
```

7.46.6 Success Response Example

```
HTTP/2 200 OK
Date: Mon, 01 Dec 2020 11:47:18 GMT
Content-Type: application/json; charset=UTF-8

{
  "bankBin":499939,
  "cardFound":true,
  "countIssuedFor180Days":1,
  "countIssuedFor30Days":1,
  "countIssuedFor90Days":1,
  "incomingTransferAmountFor30Days":30.000,
  "incomingTransferAmountFor365Days":100.000,
  "incomingTransferAmountFor60Days":40.000,
  "incomingTransferAmountFor90Days":50.000,
  "lastDischargeAmount":10.000,
  "lastDischargeDate":"2023.03.26",
  "lastFourDigits":"5721",
  "lastSuccessfulDischargeAmount":10.000,
  "lastSuccessfulDischargeDate":"2023.03.26",
  "mfoCountFor180Days":1,
  "mfoCountFor30Days":1,
  "mfoCountFor90Days":1,
  "mfoIssuedFor180Days":1,
  "mfoIssuedFor30Days":1,
  "mfoIssuedFor90Days":1,
```

(continues on next page)

(continued from previous page)

```
"orderId":2709907,  
"outgoingTransferAmountFor30Days":10.000,  
"outgoingTransferAmountFor365Days":60.000,  
"outgoingTransferAmountFor60Days":20.000,  
"outgoingTransferAmountFor90Days":30.000,  
"totalDischargeAmount":200.000,  
"totalIssuedAmount":150.000,  
"totalRecurrentAmount":200.000  
"transfersFromMF0":false  
}
```

7.46.7 Fail Response Example

```
HTTP/2 200 OK  
Date: Mon, 01 Dec 2020 11:47:18 GMT  
Content-Type: application/json; charset=UTF-8  
  
{ "orderId":1234567, "cardFound":false }
```

7.46.8 Signature computation

To compute the signature, do the following:

1. Sort POSTed parameters by their keys, alphabetically.
2. Concatenate values of these parameters in that order inserting the semicolon character ; between the values (only values for which there is an actual value presents need to be considered). This will produce a base string that will be signed.
3. Remove any gyphens - from your connecting party control key, then decode it using HEX encoding; this will produce a HMAC key.
4. Compute `hmac_sha1(base_string, hmac_key)`.
5. Encode the result as HEX this will produce the signature.

Computed signature needs to be put to the X-Authorization request header.

7.46.9 Request Builder

To get random data on staging environment sandbox.doc2.com use cardNumber = 4455555555555544 or cardRefId = 555555.

7.47 /api/mfo/v2/scoring-form

- [Introduction](#)
- [Form API URLs](#)
- [Request parameters](#)
- [Response parameters](#)
- [Request example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.47.1 Introduction

Card scoring form is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#). {clientOrderId} in request URL is generated by Connecting Party. {endpointId} in request URL is provided by Payment Gateway.

7.47.2 Form API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/mfo/v2/scoring-form/{endpointId}/{clientOrderId}	https://gate.doc2.com/paynet/api/mfo/v2/scoring-form/{endpointId}/{clientOrderId}

7.47.3 Request parameters

Parameter name	Description	Value
redirectUrl	URL the customer will be redirected to upon completion of the card scoring request	Type: String Length: 128

7.47.4 Response parameters

Parameter name	Description	Format
error	Error description (if error occurs).	Type: String
orderId	Order id on payment gateway.	Type: String
redirectUrl	URL the customer will be redirected to fill the card scoring request data.	Type: String

7.47.5 Request example

```
POST /paynet/api/mfo/v2/scoring-form/3772/123 HTTP/2
Host: sandbox.doc2.com
User-Agent: curl/7.88.1
Accept: */*
Authorization: 0Auth realm="",oauth_version="1.0",oauth_consumer_key="test_merch",
  ↳oauth_timestamp="1709030996",oauth_nonce="RrB6HIwPqtg",oauth_signature_method="HMAC-
  ↳SHA1",oauth_signature="kQsRHApN8f%2FKfM42TjPGHHUK3xE%3D"
Content-Length: 52
Content-Type: application/x-www-form-urlencoded

redirectUrl=http%3A%2F%2Fconnectingparty.com%2Fexample%2Fredirect%2F
```

7.47.6 Success Response Example

```
HTTP/2 200 OK
Date: Mon, 01 Dec 2020 11:47:18 GMT
Content-Type: application/json; charset=UTF-8

{"orderId":2793619,"redirectUrl":"https://sandbox.doc2.com/paynet/form/mfo-scoring/
  ↳init/
  ↳BB474B5A46346173433239614477534F576533495256794B523362374F65756C73677A5162376C6E365933493D
  ↳"}
  ↳}
```

7.47.7 Fail Response Example

```
HTTP/2 403 Forbidden
Date: Mon, 01 Dec 2020 11:47:18 GMT
Content-Type: application; charset=UTF-8

type=error
&serial-number=000000000-0000-0000-0000-000003d94fed
&error-message=Forbidden
&error-code=-1
```

7.47.8 Request Builder

7.48 /api/mfo/v2/scoring-result

- [Introduction](#)
- [Form Request Results API](#)
- [Request parameters](#)
- [Response Parameters](#)
- [Request example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.48.1 Introduction

Card scoring result is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication. See [Statuses](#). {orderId} in request URL is returned in Scoring Form response. {endpointId} in request URL is provided by Payment Gateway.

7.48.2 Form Request Results API

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/mfo/v2/scoring-result/{endpointId}/{orderId}	https://gate.doc2.com/paynet/api/mfo/v2/scoring-result/{endpointId}/{orderId}

7.48.3 Request parameters

This type of request has no parameters.

7.48.4 Response Parameters

Parameter name	Description	Format
error	If error occurs, here you may see brief error description.	Type: String
bankBin	Card`s bank identification number (As a rule, first 6 digits of the card number).	Type: Integer Example: 444444
cardFound	This parameter shows if the system has card scoring information on this card or not.	Type: Boolean Example: true
countIssuedFor180Days	The number of transactions for the issuance of MFI loans for the last 180 days.	Type: Integer Example: 1
countIssuedFor30Days	The number of transactions for the issuance of MFI loans for the last 30 days.	Type: Integer Example: 1
countIssuedFor90Days	The number of transactions for the issuance of MFI loans for the last 90 days.	Type: Integer Example: 1
expiredMonth	Card expiration month.	Type: Integer Example: 12
expiredYear	Card expiration year.	Type: Integer Example: 2099
incomingTransferAmount-For30Days	Incoming transfer amount for 30 days.	Type: Decimal Example: 100.25
incomingTransferAmount-For365Days	Incoming transfer amount for 365 days.	Type: Decimal Example: 100.25

continues on next page

Table 186 – continued from previous page

Parameter name	Description	Format
incomingTransferAmount-For60Days	Incoming transfer amount for 60 days.	Type: Decimal Example: 100.25
incomingTransferAmount-For90Days	Incoming transfer amount for 90 days.	Type: Decimal Example: 100.25
lastDischargeAmount	Amount of last repayment.	Type: Decimal Example: 111.400
lastDischargeDate	Date of last repayment.	Type: YYYY.MM.DD Example: 2015.07.24
lastFourDigits	Last 4 digits of the card number.	Type: Integer Example: 1234
lastSuccessfulDischargeAmount	The amount of the last successful repayment.	Type: Decimal Example: 1.400
lastSuccessfulDischargeDate	The date of the last successful repayment.	Type: YYYY.MM.DD Example: 2015.07.24
mfoCountFor180Days	Number of unique MFIs conducting transactions in the last 180 days.	Type: Integer Example: 1
mfoCountFor30Days	Number of unique MFIs conducting transactions in the last 30 days.	Type: Integer Example: 1
mfoCountFor90Days	Number of unique MFIs conducting transactions in the last 90 days.	Type: Integer Example: 1
mfoIssuedFor180Days	Number of unique MFIs that issued loans in the last 180 days.	Type: Integer Example: 1

continues on next page

Table 186 – continued from previous page

Parameter name	Description	Format
mfoIssuedFor30Days	Number of unique MFIs that issued loans in the last 30 days.	Type: Integer Example: 1
mfoIssuedFor90Days	Number of unique MFIs that issued loans in the last 90 days.	Type: Integer Example: 1
orderId	Number of the order generated in the system.	Type: Integer Example: 3623277
outgoingTransferAmount-For30Days	Outgoing transfer amount for 30 days.	Type: Decimal Example: 100.25
outgoingTransferAmount-For365Days	Outgoing transfer amount for 365 days.	Type: Decimal Example: 100.25
outgoingTransferAmount-For60Days	Outgoing transfer amount for 60 days.	Type: Decimal Example: 100.25
outgoingTransferAmount-For90Days	Outgoing transfer amount for 90 days.	Type: Decimal Example: 100.25
totalDischargeAmount	Amount of all issued funds.	Type: Decimal Example: 15.410
totalIssuedAmount	Amount of all disbursed funds.	Type: Decimal Example: 10.420
totalRecurrentAmount	Amount of all forcibly written-off funds.	Type: Decimal Example: 9.000
transfersFromMFO	Indicator of MFI loans.	Type: true/false Example: true

7.48.5 Request example

```
POST /paynet/api/mfo/v2/scoring-result/3772/2763506
HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.88.1
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="test_merch",
  ↳oauth_timestamp="1707860432",oauth_nonce="Ng6vHfIZohT",oauth_signature_method="HMAC-
  ↳SHA1",oauth_signature="xRipq64VebWslXFL1kVBy36dEmg%3D"
Content-Length: 142
Content-Type: application/x-www-form-urlencoded
Connection: close

oauth_consumer_key=test_merch&oauth_nonce=Ng6vHfIZohT&oauth_signature_method=HMAC-
  ↳SHA1&oauth_timestamp=1707860432&oauth_version=1.0
```

7.48.6 Success Response Example

```
HTTP/2 200 OK
Date: Mon, 01 Dec 2020 11:47:18 GMT
Content-Type: application/json; charset=UTF-8

{
  "bankBin":499939,
  "cardFound":true,
  "countIssuedFor180Days":1,
  "countIssuedFor30Days":1,
  "countIssuedFor90Days":1,
  "incomingTransferAmountFor30Days":30.000,
  "incomingTransferAmountFor365Days":100.000,
  "incomingTransferAmountFor60Days":40.000,
  "incomingTransferAmountFor90Days":50.000,
  "lastDischargeAmount":10.000,
  "lastDischargeDate":"2023.03.26",
  "lastFourDigits":"5721",
  "lastSuccessfulDischargeAmount":10.000,
  "lastSuccessfulDischargeDate":"2023.03.26",
  "mfoCountFor180Days":1,
  "mfoCountFor30Days":1,
  "mfoCountFor90Days":1,
  "mfoIssuedFor180Days":1,
  "mfoIssuedFor30Days":1,
  "mfoIssuedFor90Days":1,
  "orderId":2709907,
  "outgoingTransferAmountFor30Days":10.000,
  "outgoingTransferAmountFor365Days":60.000,
  "outgoingTransferAmountFor60Days":20.000,
  "outgoingTransferAmountFor90Days":30.000,
  "totalDischargeAmount":200.000,
  "totalIssuedAmount":150.000,
  "totalRecurrentAmount":200.000
  "transfersFromMFO":false
}
```


7.48.7 Fail Response Example

```
HTTP/2 403 Forbidden
Date: Mon, 01 Dec 2020 11:47:18 GMT
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

type=error
&serial-number=00000000-0000-0000-0000-000003da9cb1
&error-message=Forbidden
&error-code=-1
```

7.48.8 Request Builder

7.49 /api/scoring/v1/score

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.49.1 Introduction

Card scoring request is initiated through HTTPS POST (with UTF-8 encoding) request to the [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication.

7.49.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/scoring/v1/score/{endpoint}	https://api.doc2.com/paynet/api/scoring/v1/score/{endpoint}

7.49.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Parameter name	Description	Format
card-ref-id	The card-ref-id created earlier using create-card-ref call that can be used to find the specific card that is needed. To create create-card-ref see /api/v2/create-card-ref/ . Send either card-ref-id or card-number, not all.	N{1-*}
card-number	Full card number. Send either card-number or card-ref-id, not all.	N{13-19}
client-order-id	Unique order identifier assigned by Connecting Party.	N{1-*}
first-name	First Name.	N{64}
last-name	Last Name.	N{64}
middle-name	Middle Name.	N{64}
birth-date	Customer's birth date.	1982-07-31
rating	For example 0000.	N{4}
phone	Customer's full international phone number, including country code.	N{11}

7.49.4 Response Parameters

Parameter name	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Connecting Party order id
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.49.5 Request example

POST /paynet/api/scoring/v1/score/2404

Parameters:

```
birth-date: 2000-01-01
card-ref-id: 12345
client-order-id: score123
create-card-ref: false
first-name: John
last-name: Smith
middle-name: M
phone: 19019019191
rating: 5
```

Headers:

```
accept: */*
accept-encoding: gzip,deflate
authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key=
↪ "connectingparty",oauth_timestamp="1702073235",oauth_nonce="OPASWJfawifJHG",oauth_
↪ signature_method="HMAC-SHA1", oauth_signature="8FpDccqJGDwppPUnLwokfJl53Mc%3D"
content-length: 419
content-type: application/x-www-form-urlencoded
```

7.49.6 Success Response Example

HTTP/2 200 OK

Date: Mon, 01 Dec 2020 11:47:18 GMT

```
type=async-response
&serial-number=00000000-0000-0000-0000-00010b87db96
&merchant-order-id=score123
&paynet-order-id=9550
&end-point-id=2404
```

7.49.7 Fail Response Example

HTTP/2 403 Forbidden

Date: Mon, 01 Dec 2020 11:47:18 GMT

```
type=error
&serial-number=00000000-0000-0000-0000-000003da9cb1
&error-message=Forbidden
&error-code=-1
```

7.49.8 Request Builder

7.50 /api/scoring/v1/status

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

7.50.1 Introduction

Card scoring status is initiated through HTTPS GET request to the [URLs](#) and the [parameters](#) specified below. Use [OAuth HMAC-SHA1](#) for authentication.

7.50.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/scoring/v1/status/{client_order_id}	https://api.doc2.com/paynet/api/scoring/v1/status/{endpoint_id}

7.50.3 Request Parameters

Note:

Request must have content-type=application/x-www-form-urlencoded and [Authorization headers](#).

Status call parameters	Description
client_orderid	Connecting Party order identifier of the transaction for which the status is requested.
orderid	Order id assigned to the order by Doc2.0.

7.50.4 Response Parameters

Parameter name	Description
type	The type of response. May be status-response.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
status	The status code of the initial transaction. May be approved, declined, processing, error or filtered. If type equals filtered it means the transaction was considered fraudulent by Doc2.0 Server.
client-order-id	Client order id
paynet-order-id	Order id assigned to the order by Doc2.0.
card-ref-id	ID generated by card-ref.
bin	First 6 digits of the card.
last-four-digits	Last 4 digits of the card.
bank-name	Card Bank Name.
scoring-service-request-id	Scoring service request ID.
scoring-service-response-id	Scoring service response ID.
score1	Scoring score 1.
score2	Scoring score 2.
score3	Scoring score 3.
scoring-service-response	Scoring service text response.
error-message	If status is declined, error, filtered this parameter contains the reason for decline.
error-code	The error code is case status is declined, error, filtered.

7.50.5 Request example

```
GET /paynet/api/scoring/v1/status/2404?client-order-id=score123&orderid=9550
Parameters:
  client-order-id: score123
  orderid: 9550
Headers:
  accept-encoding: gzip,deflate
  authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key=
  "connectingparty",oauth_timestamp="1702073235",oauth_nonce="PSWvnfwMDH",oauth_
  "signature_method="HMAC-SHA1",oauth_signature="lYpN%2BRi4u3tcCXJknQGVxlIGKQk%3D"
```

7.50.6 Success Response Example

```
HTTP/2 200 OK
Date: Mon, 01 Dec 2020 11:47:18 GMT

type=scoring-status-response
&serial-number=00000000-0000-0000-0000-00010b87dbc2
&status=approved
&client-order-id=score123
```

(continues on next page)

(continued from previous page)

```
&paynet-order-id=9550
&bin=555555
&last-four-digits=4444
&bank-name=WORLDBANK
&scoring-service-request-id=9550
&scoring-service-response=<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <response><id>9550</id><cd>0</cd><results><result><name>score</name><value>0.
  <5820386168936834</value></result></results></response>
```

7.50.7 Fail Response Example

```
HTTP/2 403 Forbidden
Date: Mon, 01 Dec 2020 11:47:18 GMT

type=error
&serial-number=00000000-0000-0000-0000-000003da9cb1
&error-message=Forbidden
&error-code=-1
```

7.50.8 Request Builder

7.51 /mapi/v1/sale

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Request Authentication Example](#)
- [Response Parameters](#)
- [Google Pay Request Example](#)
- [Google Pay Response Example](#)

7.51.1 Introduction

Application initiates a transaction by sending HTTPS POST request by using [URL](#) and the [parameters](#) specified below. Use [HMAC-SHA1](#) for authentication.

7.51.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/mapi/v1/sale/ENDPOINTID	https://api.doc2.com/paynet/mapi/v1/sale/ENDPOINTID

7.51.3 Request Parameters

Note:

* leading and trailing whitespace in input parameters will be omitted

Please note the following characters must be escaped in the parameter values: & + \.

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Mandatory Type: String Length: 128
order_desc	Brief order description	Necessity: Mandatory Type: String Length: 64k
first_name	Payer's first name.	Necessity: Optional Type: String Length: 50

continues on next page

Table 194 – continued from previous page

Parameter Name	Description	Value
last_name	Payer's last name	Necessity: Optional Type: String Length: 50
ssn	Last four digits of the payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYY-MM-DD.	Necessity: Optional Type: Numeric Length: 8
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Mandatory Type: String Length: 256
city	Payer's city.	Necessity: Mandatory Type: String Length: 50
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Optional Type: String Length: 2
zip_code	Payer's ZIP code	Necessity: Mandatory Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Mandatory Type: String Length: 2

continues on next page

Table 194 – continued from previous page

Parameter Name	Description	Value
phone	Payer's full international phone number, including country code.	Necessity: Optional Type: String Length: 15
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
email	Payer's e-mail address.	Necessity: Mandatory Type: String Length: 50
purpose	Destination to where the payment goes. It is useful for the Connecting Partys who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents	Necessity: Mandatory Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro	Necessity: Mandatory Type: String Length: 3
cvv2	Customer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card. It used if acquirer provides only E-Commerce connection	Necessity: Optional Type: Numeric Length: 3-4

continues on next page

Table 194 – continued from previous page

Parameter Name	Description	Value
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Mandatory Type: String Length: 20
encrypted-type	Type of the encryption. This depends on mPOS model	Necessity: Mandatory Type: String Length: 32
encrypted-data	Card tracks data encrypted using encrypted-type encryption and encoded using HEX encoding. For more info contact support	Necessity: Mandatory Type: String Length: -
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters .	Necessity: Optional Type: String Length: 1024

7.51.4 Request Authentication Example

To make sure that request is authorized by Connecting Party, all requests must be signed. MAPI works with OAuth 1.0a HMAC-SHA1 signature as it is a well-established and supported in the client libraries.

MAPI supports only the signature part and does not support a full-blown OAuth. Connecting Party control key is used as a secret key.

Below is an example of how to make a signed request in Java with help of [scribe library](#)¹⁸:

```
import org.scribe.builder.api.DefaultApi10a;
import org.scribe.model.*;
import org.scribe.oauth.OAuth10aServiceImpl;
import org.scribe.oauth.OAuthService;
import org.scribe.services.HMACSha1SignatureService;
import org.scribe.services.SignatureService;
```

(continues on next page)

¹⁸ <https://github.com/scribejava/scribejava>

(continued from previous page)

```

import java.util.Map;

public class App {
    public String doPost(String url, Map<String, String>parameters) {
        OAuthConfig config = new OAuthConfig(apiToken, merchantControlKey,
        ↪ OAuthConstants.OUT_OF_BAND,
            SignatureType.Header, null, null);
        OAuthService service = new OAuth10aServiceImpl(new HmacSha1Mapi(), config);
        OAuthRequest request = new OAuthRequest(Verb.POST, url);
        for (Map.Entry < String, String > entry : parameters.entrySet()){
            request.addBodyParameter(entry.getKey(), entry.getValue());
        } // empty token for 'two-legged'
        Token token = new Token("", "");
        service.signRequest(token, request);
        Response response = request.send();
        return response.getBody();
    }

    private static class HmacSha1Mapi extends DefaultApi10a {
        @Override
        public String getRequestTokenEndpoint() {
            return null; // not used
        }

        @Override
        public String getAccessTokenEndpoint() {
            return null; //not used
        }

        @Override
        public String getAuthorizationUrl(Token requestToken) {
            return null; // not used
        }

        @Override
        public SignatureService getSignatureService() {
            return new HMACSha1SignatureService();
        }
    }
}

```

7.51.5 Response Parameters

Parameter name	Description
type	The type of response. May be async-form-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
status	See Status List for details
paynet-order-id	Order id assigned to the order by gate.doc2.com
merchant-order-id	Connecting Party order id
serial-number	Unique number assigned by gate.doc2.com server to particular request from the Connecting Party.

continues on next page

Table 195 – continued from previous page

Parameter name	Description
error-message	If status is declined or error this parameter contains the reason for decline or error details
error-code	The error code in case of declined or error status

7.51.6 Google Pay Request Example

Once a Google Pay token is retrieved, it may look like follows:

```
{
  "signature": "MEUCIQCOH1uW2LaQheaE4bg074mqd3QpFIx/BJxG/
  OGMehhDqwIgPb6XzlwDwBM+SLzk2HYnAy0h/Pkps8l8Bn13AVc9hQQ\u003d",
  "intermediateSigningKey": {
    "signedKey": "{\"keyValue":
    MFkwEwYHKOZiZj0CAQYIKoZiZj0DAQcDQgAEiZi3I+5g0xH3k0XMxiNyo0MMRwU7sD+/
    qks15AdgSU5+xH2PTFwe9yCQLzn4DSdjo3QKIyc+c5fUG1dkasBvKw\\u003d\\u003d", "keyExpiration
    ": "1649766119579"}",
    "signatures": [

    MEYCIQDLUKwcyf9AAqyJ6WSLrT5jG+Hd4iKvoXxaayvFuerX8wIhAMzHqN1xyAIRhznjLkkUA0Qgbg7cUltK0FH1hvRL7CAN
    ]
  },
  "protocolVersion": "ECv2",
  "signedMessage": "{\"encryptedMessage":
  EOpg0NNX90pyZlR5FEe6CMNf4pkeJvEh000G94cBD+y1x22LdGE55Le2gBRI3mFRCusQDaQtyCS5PvoLp0/
  urpWNr8xTxW/EwxjnqkYgUBRFG0mbJQACEA/7MA1Pu9w4gs0C0Y2akgurdCbLQkLasTSc5e/

  i6GWmdiXqWAWM8JrNBVklpYnFek3FCeK3W23mYHntedCLyTTPVmbUeMQ4n8rhgzo+8fiWNjKywLjVZLJ2o+oCNxhMVfhG74GdW

  W4EoWGTE6A+pEyI8jTbA4Y0box39exJbNthQsThqXg50KJ0EDcMNYBPg/lbyNQfY8NjvBMA7fB//
  QiZqIu8WgZFfSiMJ+j4uXHdCDfLT9biw4pde7bkRuH/Uofd1CabiFhQSu64SeEEKj0r87wEY55sNr8Kz
  SCAww0gQP0XYlt/t0A5KT9k172I84WtiWQYr/
  UNuIVURiH8x08e8Ihx9NvkGx+Gz/ZyUVQH9RXjEXDRNvm63sSiUMB1TLy9PiMK00451Uw2yw30c/
  8SzyKByPYtNdHwmd6iAhfoQSpJAKMTwhYMmKFZC+b8K

  HKbx8Ge+NvKv2GpCLWY0WL4wvqWNPk21Jzc0bFcnzztafdTPbbEB1pfY", "ephemeralPublicKey":
  BGVn0SZ1J5pDZlInHve5nj1Ybj+C0oLfcTygynSqU8YqKaZs8ZdHQjVLI5rRI10PmskgZSMH
  YXdkM0k+/D8/o0o\\u003d", "tag":
  "II0dhc0ZtYl26Yln3GNSXTK4yalRylKsKj71RyjZfFM\\u003d"}"
}
```

Make HEX from the token:

```
7B0A2020227369676E6174757265223A20224D4555434951434F48317557324C6151686561453462673037346D7164335170
6C38426E3133415663396851515C7530303364222C0A202022696E7465726D6564696174655369676E696E674B6579223A20
3043415159494B6F5A497A6A30444151634451674145695A6933492B3567307848336B4F584D78694E796F4F4D4D52775537
646B617342764B775C5C75303033645C5C75303033645C222C5C226B657945787069726174696F6E5C223A5C223136343937
4B7763796639414171794A3657534C7254356A472B486434694B766F587861617976465565725838774968414D7A48714E31
20202270726F746F636F6C56657273696F6E223A20224543763222C0A2020227369676E65644D657373616765223A20227B
6B654A7645684F4F4F4739346342442B79317832324C64474535354C653267425249336D4652437573514461517479435335
7734677330434F5932616B6775726443624C516B4C617354536335652F693647576D646958715741574D384A724E42566B6C
4B79774C6A565A6C4A326F2B6F434E78684D566668473734476457766F623254664531756F65584F43584F4830327538746F
386A546241345930626F78333965784A624E746851735468715867354F4B4A4F4544634D4E594250672F6C62794E51665938
5275482F556F6664314361626966685153753634536545456B6A307238377745593535734E72384B7A53434177774F675150
```

(continues on next page)

(continued from previous page)

```
67782B477A2F5A79555651483952586A455844524E766D3633735369554D4231546C593950694D4B4F303435315577327977
783847652B4E564B76324770434C777930574C34777671574E506B32314A7A63306246636E7A7A7461666454506262454231
356E6A3159626A2B43306F6C4663547967796E5371553859714B615A73385A6448516A566C49537252493130506D736B675A
36596C6E33474E5358544B3479616C52796C4B734B6A373152796A5A66464D5C5C75303033645C227D220A7D
```

Form a request setting encrypted-data equals token HEX and encrypted-type=googlepay:

```
POST paynet/mapi/v1/sale/27764 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="test_merchant14",
↳oauth_signature_method="HMAC-SHA1",oauth_signature="hSXMSUPP%2FIGY00WpU4LVx0mu7SA%3D
↳"
Content-Length: 108
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=123098
&cvv2=XXX
&amount=114.94
&ipaddress=115.135.52.242
&state=
&currency=USD
&phone=+6072344354
&zip_code=81200
&order_desc=Super product 1
&email=francislusaikun@yahoo.com
&country=MY
&city=Johor Bahru
&address1=11 Jalan Lurah 6 Kg. Kempas Baru
&redirect_url=http://MERCHANT_SITE/payment_update/123098/
&encrypted-type=googlepay
&encrypted-
↳data=7B0A2020227369676E6174757265223A20224D4555434951434F48317557324C6151686561453462673037346D716
30682F506B7073386C38426E3133415663396851515C7530303364222C0A202022696E7465726D6564696174655369676E69
59484B6F5A497A6A3043415159494B6F5A497A6A30444151634451674145695A6933492B3567307848336B4F584D78694E79
632B633566554731646B617342764B775C5C75303033645C5C75303033645C222C5C226B657945787069726174696F6E5C22
4559434951444C554B7763796639414171794A3657534C7254356A472B486434694B766F5878616179764655657258387749
205D0A20207D2C0A20202270726F746F636F6C56657273696F6E223A202245437632222C0A2020227369676E65644D657373
6536434D4E6634706B654A7645684F4F4F4739346342442B79317832324C64474535354C653267425249336D465243757351
2F374D416C5075397734677330434F5932616B6775726443624C516B4C617354536335652F693647576D646958715741574D
6F2B386669574E6A4B79774C6A565A6C4A326F2B6F434E78684D566668473734476457766F623254664531756F65584F4358
4536412B70457949386A546241345930626F78333965784A624E746851735468715867354F4B4A4F4544634D4E594250672F
773470646537626B5275482F556F6664314361626966685153753634536545456B6A307238377745593535734E72384B7A53
38496878394E766B67782B477A2F5A79555651483952586A455844524E766D3633735369554D4231546C593950694D4B4F30
432B62384B484B62783847652B4E564B76324770434C777930574C34777671574E506B32314A7A63306246636E7A7A746166
445A6C496E487665356E6A3159626A2B43306F6C4663547967796E5371553859714B615A73385A6448516A566C4953725249
68634F5A74596C3236596C6E33474E5358544B3479616C52796C4B734B6A373152796A5A66464D5C5C75303033645C227D22
```

7.51.7 Google Pay Response Example

```
type=async-response
&serial-number=00000000-0000-0000-0000-00000000624e8
&merchant-order-id=59e1e3ca-5d44-11e1-b3d6-002522b853b4
&paynet-order-id=94935
```

7.52 /mapi/v1/preauth

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Request Authentication Example](#)
- [Response Parameters](#)
- [Google Pay Request Example](#)
- [Google Pay Response Example](#)

7.52.1 Introduction

Application initiates a transaction by sending HTTPS POST request by using [URL](#) and the [parameters](#) specified below. Use [HMAC-SHA1](#) for authentication.

7.52.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/mapi/v1/preauth/ENDPOINT	https://api.doc2.com/paynet/mapi/v1/preauth/ENDPOINT

7.52.3 Request Parameters

Note:

* leading and trailing whitespace in input parameters will be omitted

Please note the following characters must be escaped in the parameter values: & + \.

Parameter Name	Description	Value
client_orderid	Unique order identifier assigned by Connecting Party.	Necessity: Mandatory Type: String Length: 128
order_desc	Brief order description	Necessity: Mandatory Type: String Length: 64k
first_name	Payer's first name.	Necessity: Optional Type: String Length: 50
last_name	Payer's last name	Necessity: Optional Type: String Length: 50
ssn	Last four digits of the payer's social security number.	Necessity: Optional Type: Numeric Length: 32
birthday	Payer's date of birth, in the format YYYYMMDD.	Necessity: Optional Type: Numeric Length: 8
address1	Payer's address line 1. (Please note that in some cases it is not possible to send address length more than 50 characters. Please contact your manager for more details.)	Necessity: Mandatory Type: String Length: 256
city	Payer's city.	Necessity: Mandatory Type: String Length: 50

continues on next page

Table 197 – continued from previous page

Parameter Name	Description	Value
state	Payer's state. Please see Mandatory State codes for a list of valid state codes. Required for USA, Canada and Australia.	Necessity: Optional Type: String Length: 2
zip_code	Payer's ZIP code	Necessity: Mandatory Type: String Length: 10
country	Payer's country. Please see Country codes for a list of valid country codes.	Necessity: Mandatory Type: String Length: 2
phone	Payer's full international phone number, including country code.	Necessity: Optional Type: String Length: 15
cell_phone	Payer's full international cell phone number, including country code.	Necessity: Optional Type: String Length: 15
email	Payer's e-mail address.	Necessity: Mandatory Type: String Length: 50
purpose	Destination to where the payment goes. It is useful for the Connecting Party who let their payers to top up their accounts with bank card (Mobile phone accounts, game accounts etc.). Sample values are: +9999999999; mail@example.com etc. This value can be used by the fraud monitoring system.	Necessity: Optional Type: String Length: 128

continues on next page

Table 197 – continued from previous page

Parameter Name	Description	Value
amount	Amount to be charged. The amount has to be specified in the highest units with . delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents	Necessity: Mandatory Type: Numeric Length: 10
currency	Currency the transaction is charged in (See: Currency codes). Sample values are: USD for US Dollar EUR for European Euro	Necessity: Mandatory Type: String Length: 3
cvv2	Customer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card. It used if acquirer provides only E-Commerce connection	Necessity: Optional Type: Numeric Length: 3-4
ipaddress	Payer's IP address, included for fraud screening purposes.	Necessity: Mandatory Type: String Length: 20
encrypted-type	Type of the encryption. This depends on mPOS model	Necessity: Mandatory Type: String Length: 32
encrypted-data	Card tracks data encrypted using encrypted-type encryption and encoded using HEX encoding. For more info contact support	Necessity: Mandatory Type: String Length: -
server_callback_url	URL, where the transaction status is sent to. Connecting Party may use server callback URL for custom processing of the transaction completion, e.g. to collect payment data in the Connecting Party's information system. For the list of parameters which come along with server callback to server_callback_url refer to Connecting Party callback parameters .	Necessity: Optional Type: String Length: 1024

7.52.4 Request Authentication Example

To make sure that request is authorized by Connecting Party, all requests must be signed. MAPI works with OAuth 1.0a HMAC-SHA1 signature as it is a well-established and supported in the client libraries.

MAPI supports only the signature part and does not support a full-blown OAuth. Connecting Party control key is used as a secret key.

Below is an example of how to make a signed request in Java with help of [scribe library](#)¹⁹:

```
import org.scribe.builder.api.DefaultApi10a;
import org.scribe.model.*;
import org.scribe.oauth.OAuth10aServiceImpl;
import org.scribe.oauth.OAuthService;
import org.scribe.services.HMACSha1SignatureService;
import org.scribe.services.SignatureService;
import java.util.Map;

public class App {
    public String doPost(String url, Map<String, String>parameters) {
        OAuthConfig config = new OAuthConfig(apiToken, merchantControlKey,
        ↪ OAuthConstants.OUT_OF_BAND,
            SignatureType.Header, null, null);
        OAuthService service = new OAuth10aServiceImpl(new HmacSha1Mapi(), config);
        OAuthRequest request = new OAuthRequest(Verb.POST, url);
        for (Map.Entry <String, String> entry : parameters.entrySet()) {
            request.addBodyParameter(entry.getKey(), entry.getValue());
        } // empty token for 'two-legged'
        Token token = new Token("", "");
        service.signRequest(token, request);
        Response response = request.send();
        return response.getBody();
    }

    private static class HmacSha1Mapi extends DefaultApi10a {
        @Override
        public String getRequestTokenEndpoint() {
            return null; // not used
        }

        @Override
        public String getAccessTokenEndpoint() {
            return null; //not used
        }

        @Override
        public String getAuthorizationUrl(Token requestToken) {
            return null; // not used
        }

        @Override
        public SignatureService getSignatureService() {
            return new HMACSha1SignatureService();
        }
    }
}
```

(continues on next page)

¹⁹ <https://github.com/scribejava/scribejava>

(continued from previous page)

```
}
}
```

7.52.5 Response Parameters

Parameter name	Description
type	The type of response. May be async-form-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
status	See Status List for details
paynet-order-id	Order id assigned to the order by gate.doc2.com
merchant-order-id	Connecting Party order id
serial-number	Unique number assigned by gate.doc2.com server to particular request from the Connecting Party.
error-message	If status is declined or error this parameter contains the reason for decline or error details
error-code	The error code in case of declined or error status

7.52.6 Google Pay Request Example

Once a Google Pay token is retrieved, it may look like follows:

```
{
  "signature": "MEUCIQCOH1uW2LaQheaE4bg074mqd3QpFIx/BJxG/
  ↳ 0GMEhhDqwIgPb6XzlwDwBM+SLzk2HYnAy0h/Pkps8l8Bn13AVc9hQQ\u003d",
  "intermediateSigningKey": {
    "signedKey": "{\"keyValue":
    ↳ "MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEiZi3I+5g0xH3k0XMxiNyo0MMRwU7sD+/
    ↳ qks15AdgSU5+xH2PTFwe9yCQLzn4DSdjo3QKIyc+c5fUG1dkasBvKw\\u003d\\u003d", "keyExpiration
    ↳ ": "1649766119579"}",
    "signatures": [
      ↳ "MEYCIQDLUKwcyf9AAqyJ6WSLrT5jG+Hd4iKvoXxaayvFUerX8wIhAMzHqN1xyAIRhznjLkkUA0Qgbg7cUltK0FH1hvRL7CAN
      ↳ "
    ]
  },
  "protocolVersion": "ECv2",
  "signedMessage": "{\"encryptedMessage":
  ↳ "E0pG0NNX90pyYzlR5FEe6CMNf4pkeJvEh000G94cBD+y1x22LdGE55Le2gBRI3mFRCusQDaQtyCS5PvoLp0/
  ↳ urpWNr8xTxW/EwxjnqkYgUBRFG0mbJQAceA/7MA1Pu9w4gs0C0Y2akgurdCbLQkLastSc5e/
  ↳ i6GwmdiXqAWM8JrNBVklpYnFek3FCeK3W23mYHntedCLyTTPVmbUeMQ4n8rhgzo+8fiWNjKywLjVZLJ2o+oCNxhMVfhG74Gdw
  ↳ W4EoWGTE6A+pEyI8jTbA4Y0box39exJbNthQsThqXg50KJ0EDcMNYBPg/lbyNQfY8NjvBMA7fB//
  ↳ QiZqIu8WgZFfSiMJ+j4uXHdCDfLT9biw4pde7bkRuH/Uofd1Cabi fhQSu64SeEEKj0r87wEY55sNr8Kz
  ↳ SCAww0gQP0XYlt/t0A5KT9k172I84WtiwQYr/
  ↳ UNuIVURiH8x08e8Ihx9NvkGx+Gz/ZyUVQH9RXjEXDRNvm63sSiUMB1TLY9PiMK00451Uw2yw30c/
  ↳ 8SzyKByPYtNdHwmd6iAhfoQSpJAKMTwhYMmKFZC+b8K
  ↳ HKbx8Ge+NVKv2GpCLWY0WL4wvqWNPk21Jzc0bFczztafdTPbbEB1pfY", "ephemeralPublicKey":
```

(continues on next page)

(continued from previous page)

```

↪ "BGVn0SZ1J5pDZlInHve5nj1Ybj+C0o1FcTygynSqU8YqKaZs8ZdHQjVlISrRI10PmskgZSMH
    YXdkM0k+/D8/o0o\\u003d", "tag":
↪ "II0dhc0ZtYl26Yln3GNSXTK4yalRylKsKj71RyjZfFM\\u003d"}"
}

```

Make HEX from the token:

```

7B0A2020227369676E6174757265223A20224D4555434951434F48317557324C6151686561453462673037346D7164335170
6C38426E3133415663396851515C7530303364222C0A202022696E7465726D6564696174655369676E696E674B6579223A20
3043415159494B6F5A497A6A30444151634451674145695A6933492B3567307848336B4F584D78694E796F4F4D4D52775537
646B617342764B775C5C75303033645C5C75303033645C222C5C226B657945787069726174696F6E5C223A5C223136343937
4B7763796639414171794A3657534C7254356A472B486434694B766F587861617976465565725838774968414D7A48714E31
20202270726F746F636F6C56657273696F6E223A20224543763222C0A2020227369676E65644D657373616765223A20227B
6B654A7645684F4F4F4739346342442B79317832324C64474535354C653267425249336D4652437573514461517479435335
7734677330434F5932616B6775726443624C516B4C617354536335652F693647576D646958715741574D384A724E42566B6C
4B79774C6A565A6C4A326F2B6F434E78684D566668473734476457766F623254664531756F65584F43584F4830327538746F
386A546241345930626F78333965784A624E746851735468715867354F4B4A4F4544634D4E594250672F6C62794E51665938
5275482F556F6664314361626966685153753634536545456B6A307238377745593535734E72384B7A53434177774F675150
67782B477A2F5A79555651483952586A455844524E766D3633735369554D4231546C593950694D4B4F303435315577327977
783847652B4E564B76324770434C777930574C34777671574E506B32314A7A63306246636E7A7A7461666454506262454231
356E6A3159626A2B43306F6C4663547967796E5371553859714B615A73385A6448516A566C49537252493130506D736B675A
36596C6E33474E5358544B3479616C52796C4B734B6A373152796A5A66464D5C5C75303033645C227D220A7D

```

Form a request setting encrypted-data equals token HEX and encrypted-type=googlepay:

```

POST paynet/mapi/v1/preauth/27764 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="test_merchant14",
↪ oauth_signature_method="HMAC-SHA1",oauth_signature="hSXMSUPP%2FIGY00WpU4LVx0mu7SA%3D
↪ "
Content-Length: 108
Content-Type: application/x-www-form-urlencoded
Connection: close

client_orderid=123098
&cvv2=XXX
&amount=114.94
&ipaddress=115.135.52.242
&state=
&currency=USD
&phone=+6072344354
&zip_code=81200
&order_desc=Super product 1
&email=francislusaikun@yahoo.com
&country=MY
&city=Johor Bahru
&address1=11 Jalan Lurah 6 Kg. Kempas Baru
&redirect_url=http://MERCHANT_SITE/payment_update/123098/
&encrypted-type=googlepay
&encrypted-
↪ data=7B0A2020227369676E6174757265223A20224D4555434951434F48317557324C6151686561453462673037346D716
30682F506B7073386C38426E3133415663396851515C7530303364222C0A202022696E7465726D6564696174655369676E69
59484B6F5A497A6A3043415159494B6F5A497A6A30444151634451674145695A6933492B3567307848336B4F584D78694E79
632B633566554731646B617342764B775C5C75303033645C5C75303033645C222C5C226B657945787069726174696F6E5C22

```

(continues on next page)

(continued from previous page)

```

4559434951444C554B7763796639414171794A3657534C7254356A472B486434694B766F5878616179764655657258387749
205D0A20207D2C0A20202270726F746F636F6C56657273696F6E223A202245437632222C0A2020227369676E65644D657373
6536434D4E6634706B654A7645684F4F4F4739346342442B79317832324C64474535354C653267425249336D465243757351
2F374D416C5075397734677330434F5932616B6775726443624C516B4C617354536335652F693647576D646958715741574D
6F2B386669574E6A4B79774C6A565A6C4A326F2B6F434E78684D566668473734476457766F623254664531756F65584F4358
4536412B70457949386A546241345930626F78333965784A624E746851735468715867354F4B4A4F4544634D4E594250672F
773470646537626B5275482F556F6664314361626966685153753634536545456B6A307238377745593535734E72384B7A53
38496878394E766B67782B477A2F5A79555651483952586A455844524E766D3633735369554D4231546C593950694D4B4F30
432B62384B484B62783847652B4E564B76324770434C777930574C34777671574E506B32314A7A63306246636E7A7A746166
445A6C496E487665356E6A3159626A2B43306F6C4663547967796E5371553859714B615A73385A6448516A566C4953725249
68634F5A74596C3236596C6E33474E5358544B3479616C52796C4B734B6A373152796A5A66464D5C5C75303933645C227D22

```

7.52.7 Google Pay Response Example

```

type=async - response
&serial-number=00000000-0000-0000-0000-00000000624e8
&merchant-order-id=59e1e3ca-5d44-11e1-b3d6-002522b853b4
&paynet-order-id=94935

```

7.53 /registry/upload

- [Introduction](#)
- [Adjustments parameters](#)
- [File example](#)
- [Request Builder](#)

7.53.1 Introduction

This API allows Manager to upload a file with adjustments for Merchant balance in Payment Gateway.

7.53.2 Adjustments parameters

Adjustments must be sent in archived tsv file format.

Parameter	Description
Credit amount	Credit amount.
Debit amount	Debit amount.
External ID	Adjustment external ID.
External Merchant ID	External merchant ID. Set in properties of merchant/manager.

continues on next page

Table 199 – continued from previous page

Parameter	Description
Balance name	Balance name. If parameter is not set, default balance is automatically selected.

7.53.3 File example

tsv file example:

10.45	eid1	THIS_IS_EXT_MERCH_ID	Global merchant balance
90.05	eid2	THIS_IS_EXT_MERCH_ID	Global merchant balance

7.53.4 Request Builder

In order to make an adjustment, tsv file must be archived and sent. 7-Zip or similar archiving software is needed in order to use this debugger.

7.54 /services/transaction-report

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.54.1 Introduction

Remote transactions report provide an opportunity to get complete information about transactions with their statuses for verification and further analysis.

Remote transactions report is initiated through HTTPS GET request by using [URLs](#) and the [parameters](#) specified below. Use [HMAC-SHA1](#) for authentication.

7.54.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet-ui/services/transaction-report	https://gate.doc2.com/paynet-ui/services/transaction-report

7.54.3 Request Parameters

Parameter Name	Mandatory	Description
createdFrom-Date	Yes	Filters transactions on creation date. Must be in the following format: YYYY-MM-DD HH:MM:SS.
createdToDate	Yes	Filters transactions on creation date. Must be in the following format: YYYY-MM-DD HH:MM:SS.
bankFromDate	No	Filters transactions on date when transaction was uploaded to bank. Must be in the following format: YYYY-MM-DD HH:MM:SS.
bankToDate	No	Filters transactions on date when transaction was uploaded to bank. Must be in the following format: YYYY-MM-DD HH:MM:SS.
transaction-Types	No	Comma-separated list of transaction types to return. If this parameter is omitted, transactions of all types. Following types are recognized: sale, reversal, preauth, capture, cancel, transfer, payout, fraud, chargeback, retrieval, dispute, refund.
transactionStatuses	No	Comma-separated list of transaction statuses to return. If this parameter is omitted, transactions in all statuses are returned. Following statuses are recognized: approved, declined, filtered.
endpoints	No	Comma-separated list of endpoint IDs to which returned transactions must belong. If this parameter is omitted, transactions belonging to any endpoint of Connecting Party who makes a request are returned.

Note:

There are some restrictions:

* If endpoints parameter is specified and some endpoints in the list do not belong to the Connecting Party who makes a request, or some endpoints from the list do not exist, an error message is returned.

* Period specified by createdFromDate and createdToDate parameters must not exceed 31 days in length.

* Cannot make more than 10 requests per hour to this report.

7.54.4 Response Parameters

Note:

The returned report has wellformed CSV format. In accordance with [RFC 4180](https://datatracker.ietf.org/doc/html/rfc4180)²⁰, fields containing line breaks (CRLF, CR, LF), double quotes, and commas are enclosed in double-quotes.

Parameter Name	Description	Value
Txid	Transaction ID in Doc2.0, also referred as Order ID.	Type: Int Length: 10
Created Date	Creation date in the server timezone.	Type: Datetime Length: -
Created Date (no time)	Creation date in the server timezone, no time.	Type: Datetime Length: -
Bank Date	Creation date in bank.	Type: Datetime Length: -
Merchant	Connecting Party's name.	Type: Varchar Length: 128
End-Point ID	Endpoint ID.	Type: Int Length: 10
Project ID	Connecting Party's project ID.	Type: Int Length: 10
Project name	Name of the project.	Type: Varchar Length: 128

continues on next page

²⁰ <https://datatracker.ietf.org/doc/html/rfc4180>

Table 202 – continued from previous page

Parameter Name	Description	Value
Currency	Transaction currency.	Type: Varchar Length: 3
Site URL	Domain name of the site.	Type: Varchar Length: 100
Card type	Card type of international payment systems.	Type: Varchar Length: 32
Ip	IP address.	Type: Varchar Length: 128
Type	Transaction type.	Type: Varchar Length: 32
Status	Transaction status.	Type: Varchar Length: 32
Error Number	Error code.	Type: Int Length: 10
Error Description	Error Description.	Type: Varchar Length: 256
Reason code	Reason code (if information is available).	Type: Varchar Length: 16
Reason	Reason (if information is available).	Type: Varchar Length: 256
Comment	Comment (if information is available).	Type: Varchar Length: 256
Amount	Transaction amount.	Type: Decimal Length: 15,3

continues on next page

Table 202 – continued from previous page

Parameter Name	Description	Value
Processor ID	Processor / acquirer transaction identification number.	Type: Varchar Length: 128
Card No	Masked PAN.	Type: Varchar Length: 20
Name	Card holder name.	Type: Varchar Length: 128
Routing Number	Routing step ID.	Type: Varchar Length: 16
Account Number	Connecting Party's balance identification number.	Type: Varchar Length: 24
Email	Customer email.	Type: Varchar Length: 128
Phone	Customer phone number.	Type: Varchar Length: 128
Address	Address passed by API.	Type: Varchar Length: 256
Zip	Zip passed by API.	Type: Varchar Length: 32
City	City passed by API.	Type: Varchar Length: 128
State	State passed by API.	Type: Varchar Length: 128
Country	Country passed by API.	Type: Varchar Length: 128

continues on next page

Table 202 – continued from previous page

Parameter Name	Description	Value
First name	Customer first name. For transfer transactions: sender first name (if sent), or receiver first name.	Type: Varchar Length: 128
Last name	Customer last name. For transfer transactions: sender last name (if sent), or receiver last name.	Type: Varchar Length: 128
Order Description	Payment description passed by API.	Type: Varchar Length: 256
Merchant OID	ID of transaction on Connecting Party's side.	Type: Varchar Length: 256
Auxiliary transaction merchant OID	ID of master transaction in Merchant order ID field, if there are no master transactions, it will be the same as Merchant OID field.	Type: Varchar Length: 256
Reversal Amount	Refund amount.	Type: Decimal Length: 15,3
Capture Amount	Transaction confirmation amount after DMS authorization.	Type: Decimal Length: 15,3
Approval Code	Authorization code.	Type: Varchar Length: 10
Auth credit code	Auth Credit Code is indicated as auth code when performing AFT and when performing OCT.	Type: Varchar Length: 10
Auth verification value	Cardholder authentication check values. Indicated only after payment for the order and in case of appropriate permission.	Type: Varchar Length: 64
Transaction XID	Electronic commercial transaction identifier. Indicated only after payment for the order and in case of appropriate permission.	Type: Varchar Length: 64
Transaction RRN	RRN - Retrieval Reference Number.	Type: Varchar Length: 20

continues on next page

Table 202 – continued from previous page

Parameter Name	Description	Value
Transaction ARN	ARN - Acquirers Reference Number.	Type: Varchar Length: 64
Card MPI status	A separate tool for checking card participation in 3-D Secure, may be Y / N.	Type: Varchar Length: 3
Electronic Commerce Indicator	ECI.	Type: Varchar Length: 3
Is 3DS	A tool for checking if transaction is 3DS-validated, may be True / False.	Type: Varchar Length: 3
Is enrolled	A tool for checking if transaction is enrolled, may be True / False.	Type: Varchar Length: 3
Support doc Cb Reason Code	These fields are related to dispute management.	Type: Varchar Length: 256
Support doc Cb Reason Name	These fields are related to dispute management.	Type: Varchar Length: 256
Described Services/Merchandise	These fields are related to dispute management.	Type: Varchar Length: 256
What is not as described	Dispute details.	Type: Varchar Length: 256
Merchant Response	These fields are related to dispute management.	Type: Text Length: -
Tried to Resolve with Merchant	These fields are related to dispute management.	Type: Varchar Length: 256
Last Contact Date	These fields are related to dispute management.	Type: Varchar Length: 256

continues on next page

Table 202 – continued from previous page

Parameter Name	Description	Value
Contact Name	These fields are related to dispute management.	Type: Varchar Length: 256
Contact Method	These fields are related to dispute management.	Type: Varchar Length: 256
Cb Comments	These fields are related to dispute management.	Type: Text Length: -
Cb Documents	These fields are related to dispute management.	Type: Text Length: -
Cb Additional Files	These fields are related to dispute management.	Type: Text Length: -
Gate ID	Gate ID.	Type: Int Length: 10
Gate Name	Gate Name.	Type: Varchar Length: 128
Gate descriptor	Descriptor specified on the gate.	Type: Varchar Length: 128
Gate method	Gate properties.	Type: Varchar Length: 255
Gate method 2	Gate properties.	Type: Varchar Length: 255
Gate business type	Gate properties.	Type: Varchar Length: 256
Gate MID	Gate MID.	Type: Varchar Length: 256

continues on next page

Table 202 – continued from previous page

Parameter Name	Description	Value
View Receipt URL	Parameter responsible for displaying the URL of the receipt.	Type: Varchar Length: 100
Destination	The parameter passed to the Purpose field.	Type: Varchar Length: 128
Client commission	Total transaction fee.	Type: Decimal Length: 15,3
Destination Card No	6 + 4 of destination card.	Type: Varchar Length: 20
Destination bank	Destination card issuer.	Type: Varchar Length: 128
Bank name	Issuer of source card.	Type: Varchar Length: 128
DAPI IMEI	Custom string provided by Connecting Party in initial request, also used as unique device 32 symbol string identifier in DAPI.	Type: Varchar Length: 32
BIN country code	ISO 3166-1 alpha-2 country code of source card issuer (determined by BIN).	Type: Varchar Length: 3
BIN country	Country name of source card issuer (determined by BIN).	Type: Varchar Length: 128
Destination BIN country code	ISO 3166-1 alpha-2 country code of destination card issuer (determined by BIN).	Type: Varchar Length: 3
Destination BIN country name	Country name of destination card issuer (determined by BIN).	Type: Varchar Length: 128
Processor	Processor Name.	Type: Varchar Length: 128

continues on next page

Table 202 – continued from previous page

Parameter Name	Description	Value
Processor ID	Processor ID.	Type: Int Length: 10
Processor Error Code	Processor Error Code.	Type: Int Length: 10
CardRefId	Card reference ID obtained at Card Registration Stage .	Type: Int Length: 10
Initial tx bank date (Server TZ)	Date received from Bank on Server time zone.	Type: Datetime Length: -
Initial tx bank date (User TZ)	Date received from Bank on Connecting Party zone.	Type: Datetime Length: -
Initial tx Amount	The amount received from Connecting Party.	Type: Decimal Length: 15,3
External merchant identifier	External merchant identifier.	Type: Varchar Length: 128
Manager rate plan	Manager rate plan name.	Type: Varchar Length: 128
Bank name attr	Bank name.	Type: Varchar Length: 128
Bank branch	Bank branch name.	Type: Varchar Length: 128
Account name	Bank account name.	Type: Varchar Length: 128
Status change date	Date of status change.	Type: Datetime Length: -

continues on next page

Table 202 – continued from previous page

Parameter Name	Description	Value
Manager rate plan id	Manager rate plan id.	Type: Int Length: 10
Currency for processing	Currency used for transaction.	Type: Varchar Length: 3
Amount in currency for processing	Amount in currency used for transaction.	Type: Decimal Length: 15,3
Capture amount in currency for processing	Capture amount in currency used for transaction.	Type: Decimal Length: 15,3
Reversal amount in currency for processing	Reversal amount in currency used for transaction.	Type: Decimal Length: 15,3
Currency rate for processing	Currency rate used for transaction.	Type: Decimal Length: 15,3
Provider currency rate	Name of rate provider for conversion.	Type: Decimal Length: 15,3
Fee in currency for processing	Fee in currency used for transaction.	Type: Decimal Length: 15,3
Transaction ID	Internal ID in Doc2.0 which helps to distinguish several processing attempts within one transaction (Txid).	Type: Varchar Length: 128
Reseller commission	Reseller transaction fee.	Type: Decimal Length: 15,3
Manager commission	Manager transaction fee.	Type: Decimal Length: 15,3
Dealer commission	Dealer transaction fee.	Type: Decimal Length: 15,3

continues on next page

Table 202 – continued from previous page

Parameter Name	Description	Value
Processor commission	Processor transaction fee.	Type: Decimal Length: 15,3
Card expiration date	Expiration date for source card.	Type: Varchar Length: 10
Destination card expiration date	Expiration date for destination card.	Type: Varchar Length: 10
Last routing type	Last routing type used in balancing for this transaction.	Type: Varchar Length: 128
Last routing value	Last routing value used in balancing for this transaction.	Type: Varchar Length: 128
IPs region	Region of Connecting Party relative to card BIN region. For MasterCard possible values are: Domestic, European (EEA), European (SEPA NON EEA), European (NON SEPA NON EEA), Interregional (Inter-European), Interregional. For Visa possible values are: Domestic, European (EEA), European (Non-EEA GB, GI), European (Non-EEA CH), European (Non-EEA AD, MC, SM, VA), Interregional (Non-EEA IL, TR), Interregional. If BIN country cannot be determined, processor region value will be: Unknown.	Type: Varchar Length: 128
Processor region	Card BIN region. Possible values are: EU, NON EU. If BIN country cannot be determined, processor region value will be: Unknown.	Type: Varchar Length: 128
Manager ID	Manager ID.	Type: Int Length: 10
Manager name	Manager name.	Type: Varchar Length: 128
Creation time	Transaction creation time in the system.	Type: Datetime Length: -

continues on next page

Table 202 – continued from previous page

Parameter Name	Description	Value
Customer level	Customer level in CMS.	Type: Varchar Length: 128
Customer IP address country	Customer country identified by IP.	Type: Varchar Length: 2

For Faster Payments System (CBII) integration

Parameter Name	Description	Value
SBP Source Account Number	Sender's bank account	Type: Varchar Length: 24
SBP Source Phone	Sender's phone	Type: Varchar Length: 128
SBP Source Bank ID	Sender's bank ID in the NSPK	Type: Varchar Length: 128
SBP Source Name	Sender's full name	Type: Varchar Length: 128
SBP Source Address	Sender's address	Type: Varchar Length: 256
SBP Destination Phone	Receiver's phone	Type: Varchar Length: 128
SBP Destination Bank ID	Receiver's bank ID	Type: Varchar Length: 128
SBP Destination Account Number	Receiver's bank account	Type: Varchar Length: 24

continues on next page

Table 203 – continued from previous page

Parameter Name	Description	Value
SBP Destination Name	Receiver's full name	Type: Varchar Length: 128
SBP Destination Address	Receiver's address	Type: Varchar Length: 256
SBP Reversal Phone	Receiver's phone	Type: Varchar Length: 128
SBP Reversal Bank ID	Receiver's bank ID	Type: Varchar Length: 128
SBP Reversal Account Number	Receiver's bank account	Type: Varchar Length: 24
SBP Reversal Name	Receiver's full name	Type: Varchar Length: 128
SBP Reversal Address	Receiver's address	Type: Varchar Length: 256

Note:

The order of parameters in transaction report might change in future releases. New parameters might be added. Please implement response handling based on parameters names, not parameters order.

7.54.5 Request Example

```
GET /paynet-ui/services/transaction-report?createdFromDate=2023-01-01%2000%3A00%3A00&
↳createdToDate=2023-02-30%2000%3A00%3A00&oauth_consumer_key=logic&oauth_
↳nonce=3gbXZF13uAn&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1690546956&oauth_
↳version=1.0 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.88.1
Accept: */*
Authorization: 0Auth realm="",oauth_version="1.0",oauth_consumer_key="logic",oauth_
↳timestamp="1690546956",oauth_nonce="3gbXZF13uAn",oauth_signature_method="HMAC-SHA1",
↳oauth_signature="G46r4h99dwB9opEYSfUsgHLJeGg%3D"
Connection: close
```

7.54.6 Success Response Example

```
Txid;Created Date (Server TZ);Created Date (User TZ);Created Date (Server TZ, no
↳time);Bank Date (Server TZ);Bank Date (User TZ);Merchant;End-Point ID;Project ID;
↳Project Name;Currency;Site URL;Card type;Ip;Type;Status;Error Number;Error
↳Description;Reason code;Reason;Comment;Amount;Processor tx ID;Card No;Name;Routing
↳Number;Account Number;Email;Phone;Address;Zip;City;State;Country;First Name;Last
↳Name;Order Description;Merchant OID;Auxiliary transaction merchant OID;Reversal
↳Amount;Capture amount;Approval Code;Auth Credit Code;Auth Verification Value;
↳Transaction XID;Transaction RRN;Transaction ARN;Card MPI Status;Electronic Commerce
↳Indicator;Is 3D;Is Enrolled;Support doc Cb reason code;Support doc Cb reason;
↳Described services/merchandise;What is not as described;Merchant response;Tried to
↳resolve with merchant;Last contact date;Contact name;Contact method;Cb Comments;Cb
↳Documents;Cb additional files;Gate ID;Gate Name;Gate Descriptor;Gate Method;Gate
↳Method 2;Gate Business type;Gate MID;View Receipt URL;Destination;Client commission;
↳Destination Card No;Destination bank name;Bank name;DAPI IMEI;BIN country code;BIN
↳country;Destination BIN country code;Destination BIN country;Processor;Processor
↳Error Code;CardRefId;Initial tx bank date (Server TZ);Initial tx bank date (User
↳TZ);Initial tx Amount;External merchant identifier;Manager rate plan;Bank name attr;
↳Bank branch;Account name;Status change date;Manager rate plan id;Currency for
↳processing;Amount in currency for processing;Capture amount in currency for
↳processing;Reversal amount in currency for processing;Currency rate for processing;
↳Provider currency rate;Fee in currency for processing;Transaction id;Reseller
↳commission;Manager commission;Dealer commission;Processor commission;Card
↳expiration date;Destination card expirationDate;Last routing type;Last routing
↳value;IPs region;Processor region;Manager ID;Manager name;Creation time;Customer
↳level
1810032;2023-05-01T17:38:43.00;2023-05-01T17:38:43.00;01.05.2023;;;Demo merchant A;
↳7829;2251;Test Rub Project;RUB;;MasterCard;85.132.102.1;sale;filtered;10137;Billing
↳country blacklisted for merchant;;;100.000;;555544XXXXX1111;TEST HOLDER;;;
↳j***b@brown.demo;8***01122;Red*** 1;***;***;RUS;Jacob;Brown;Test payment;ylt2sut-
↳hqXHxTFspRVUd9Uvcsg;ylt2sut-hqXHxTFspRVUd9Uvcsg;;;No;;;3197;Test
↳RUB;DEM02;;;https://sandbox.doc2.com/paynet/view-receipt/7829/b6123559-fc64-32fd-
↳9c91-af49a92aa2b9;purpose?;0.000;;NATIONAL BANK OF RAS AL-KHAIMAH (RAKBANK);;AE;
↳United Arab Emirates;;;Test processor;;;2023-05-01T17:38:43.00;2023-05-01T17:38:43.
↳00;100.000;;zrt;;;2023-05-01T17:38:43.00;398;RUB;100.000;;;1537212;0.000;0.000;0.
↳000;0.000;203604;;route by Source Credit Card Issuer Country;OTHERS;Unknown;Unknown;
↳76;test_manager;;
1812039;2023-05-04T19:11:29.00;2023-05-04T19:11:29.00;04.05.2023;;;Demo merchant A;
↳6655;1712;Demo Ams EUR;EUR;;Visa;34.129.65.121;preauth;declined;1015;SECURE_3D_
↳TIMEOUT;;;5.000;;444455XXXXX1111;TEST HOLDER;;;a***v@example.com;+3***12233;
↳Wes***HX;***;Ams***en;;NLD;Firstname;Lastname;Test payment;902B4FF5;902B4FF5;;;3,
↳12424E+11;;;Yes;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https://sandbox.
↳doc2.com/paynet/view-receipt/6655/a4b048ad-47ce-33ed-af73-68c7646cee0c;0.000;;
↳UNKNOWN;;US;United States;;;Test processor;timeout;;2023-05-04T19:11:29.00;2023-05-
↳04T19:11:29.00;5.000;;2% cyprus;;;2023-05-04T19:11:29.00;582;USD;5.500;;;1.
↳099663612901000;1.099663612901000;;1539210;0.000;0.000;0.000;0.000;202401;;route by
↳Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;;
1812040;2023-05-04T19:11:29.00;2023-05-04T19:11:29.00;04.05.2023;;;Demo merchant A;
↳6655;1712;Demo Ams EUR;EUR;;Visa;34.129.65.121;preauth;declined;1015;SECURE_3D_
↳TIMEOUT;;;5.000;;444455XXXXX1111;TEST HOLDER;;;a***v@example.com;+3***12233;
↳Wes***HX;***;Ams***en;;NLD;Firstname;Lastname;Test payment;902B4FF5;902B4FF5;;;3,
↳12424E+11;;;Yes;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https://sandbox.
↳doc2.com/paynet/view-receipt/6655/270932f6-6687-3127-a32a-2baa6eb1b3f6;0.000;;
↳UNKNOWN;;US;United States;;;Test processor;timeout;;2023-05-04T19:11:29.00;2023-05-
↳04T19:11:29.00;5.000;;2% cyprus;;;2023-05-04T19:11:29.00;582;USD;5.500;;;1.
↳099663612901000;1.099663612901000;;1539211;0.000;0.000;0.000;0.000;202401;;route by
```

(continues on next page)

(continued from previous page)

```

Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;;
1812037;2023-05-04T19:11:30.00;2023-05-04T19:11:30.00;04.05.2023;;;Demo merchant A;
6655;1712;Demo Ams EUR;EUR;;Visa;34.129.65.121;sale;declined;1015;SECURE_3D_TIMEOUT;
;;;5.000;;444455XXXXXX1111;TEST HOLDER;;;a***v@example.com;+3***12233;Wes***HX;***;
Ams***en;;NLD;Firstname;Lastname;Rebill order description;902B4FF5;902B4FF5;;;;;;3,
12425E+11;;;Yes;;;;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https://sandbox.
doc2.com/paynet/view-receipt/6655/711a99be-e32f-3abe-9740-e4bda0c2d372;;0.000;;;
UNKNOWN;;US;United States;;;Test processor;timeout;;2023-05-04T19:11:30.00;2023-05-
04T19:11:30.00;5.000;;2% cyprus;;;2023-05-04T19:11:30.00;582;USD;5.500;;;1.
099663612901000;1.099663612901000;;1539212;0.000;0.000;0.000;0.000;202401;;route by
Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;;
1812038;2023-05-04T19:11:30.00;2023-05-04T19:11:30.00;04.05.2023;;;Demo merchant A;
6655;1712;Demo Ams EUR;EUR;;Visa;34.129.65.121;sale;declined;1015;SECURE_3D_TIMEOUT;
;;;5.000;;444455XXXXXX1111;TEST HOLDER;;;a***v@example.com;+3***12233;Wes***HX;***;
Ams***en;;NLD;Firstname;Lastname;Rebill order description;902B4FF5;902B4FF5;;;;;;3,
12454E+11;;;Yes;;;;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https://sandbox.
doc2.com/paynet/view-receipt/6655/52e5301a-55a7-3939-9f9c-6a0114e32c15;;0.000;;;
UNKNOWN;;US;United States;;;Test processor;timeout;;2023-05-04T19:11:30.00;2023-05-
04T19:11:30.00;5.000;;2% cyprus;;;2023-05-04T19:11:30.00;582;USD;5.500;;;1.
099663612901000;1.099663612901000;;1539213;0.000;0.000;0.000;0.000;202401;;route by
Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;;
1812041;2023-05-04T19:16:26.00;2023-05-04T19:16:26.00;04.05.2023;;;Demo merchant A;
6655;1712;Demo Ams EUR;EUR;;Visa;65.153.12.213;sale;declined;1015;SECURE_3D_TIMEOUT;
;;;10.420;;444455XXXXXX1111;TEST HOLDER;;;j***h@gmail.com;1***82043;Wes***HX;***;
Ams***en;;NLD;Firstname;Lastname;Test Order Description;902B4FF5;902B4FF5;;;;;;3,
1243E+11;;;Yes;;;;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https://sandbox.
doc2.com/paynet/view-receipt/6655/06636793-308d-33eb-8a21-77f92b00cd5d;;0.000;;;
UNKNOWN;;US;United States;;;Test processor;timeout;190926;2023-05-04T19:16:26.00;
2023-05-04T19:16:26.00;10.420;;2% cyprus;;;2023-05-04T19:16:26.00;582;USD;11.460;;;
1.099663612901000;1.099663612901000;;1539214;0.000;0.000;0.000;0.000;202401;;route
by Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;;
1812042;2023-05-04T19:16:26.00;2023-05-04T19:16:26.00;04.05.2023;;;Demo merchant A;
6655;1712;Demo Ams EUR;EUR;;Visa;65.153.12.213;sale;declined;1015;SECURE_3D_TIMEOUT;
;;;10.420;;444455XXXXXX1111;TEST HOLDER;;;j***h@gmail.com;1***82043;Wes***HX;***;
Ams***en;;NLD;Firstname;Lastname;Test Order Description;902B4FF5;902B4FF5;;;;;;3,
12453E+11;;;Yes;;;;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https://sandbox.
doc2.com/paynet/view-receipt/6655/d9a800f3-d915-3099-a769-01f9fcb7d356;;0.000;;;
UNKNOWN;;US;United States;;;Test processor;timeout;;2023-05-04T19:16:26.00;2023-05-
04T19:16:26.00;10.420;;2% cyprus;;;2023-05-04T19:16:26.00;582;USD;11.460;;;1.
099663612901000;1.099663612901000;;1539215;0.000;0.000;0.000;0.000;202401;;route by
Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;;
1812688;2023-05-05T17:18:52.00;2023-05-05T17:18:52.00;05.05.2023;;;Demo merchant A;
6655;1712;Demo Ams EUR;EUR;;Visa;89.147.253.161;transfer;filtered;10204;Customer
data validation failed (firstname, lastname, cardholder, email, phone);;;10.420;;
421070XXXXXX5721;John Doe;;;;;;Your order description;1;1;;;;;;No;;;;;;
;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https://sandbox.doc2.com/paynet/view-
receipt/6655/402efad3-7e4a-3497-8bd4-792a11074f7f;;0.000;423261XXXXXX1636;JPMORGAN
CHASE BANK N.A.;DEMIRBANK OJSC;;AZ;Azerbaijan;US;United States;Test processor;;
190924;2023-05-05T17:18:52.00;2023-05-05T17:18:52.00;10.420;;2% cyprus;;;2023-05-
05T17:18:52.00;582;EUR;10.420;;;;1539816;0.000;0.000;0.000;0.000;209912;;route by
Customer Loyalty;NEW FOR MERCHANT;Unknown;Unknown;76;test_manager;;
1817526;2023-05-16T16:15:13.00;2023-05-16T16:15:13.00;16.05.2023;;;Demo merchant A;
6655;1712;Demo Ams EUR;EUR;;Visa;85.132.52.251;sale;approved;;;100.000;PNTEST-
1817526;444455XXXXXX1111;TEST HOLDER;;;a***v@example.com;+3***12233;Wes***HX;***;
Ams***en;;NLD;Firstname;Lastname;Test payment;test;1817525;100.000;;504362;502894;;;
3,1367E+11;179416949;;Yes;;;;;;2470;A VISA CC Ams EUR;DEM01;;;test1;

```

(continues on next page)

(continued from previous page)

```

→https://sandbox.doc2.com/paynet/view-receipt/6655/fdf7267b-876b-36b8-84a7-
→2fea7a3685e4;;2.200;;;UNKNOWN;;US;United States;;;Test processor;;;2023-05-
→16T16:15:13.00;2023-05-16T16:15:13.00;100.000;;2% cyprus;;;2023-05-16T16:16:11.00;
→582;USD;108.740;;;1.087320537745000;1.087320537745000;;1544050;0.000;2.200;0.000;1.
→000;202701;;route by Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;;
1817526;2023-05-16T16:15:16.00;2023-05-16T16:15:16.00;16.05.2023;;;Demo merchant A;
→6655;1712;Demo Ams EUR;EUR;;Visa;85.132.52.251;reversal;approved;;;;;Auto-return;100.
→000;1,68424E+12;444455XXXXXX1111;TEST HOLDER;;;a***v@example.com;+3***12233;
→Wes***HX;***;Ams***en;;NLD;Firstname;Lastname;Test payment;test;1817525;100.000;;
→504362;502894;;;3,13665E+11;179416949;;;Yes;;;;;;2470;A VISA CC Ams EUR;
→DEM01;;;test1;https://sandbox.doc2.com/paynet/view-receipt/6655/fdf7267b-876b-36b8-
→84a7-2fea7a3685e4;;0.000;;;UNKNOWN;;US;United States;;;Test processor;;;2023-05-
→16T16:15:13.00;2023-05-16T16:15:13.00;100.000;;2% cyprus;;;2023-05-16T16:16:11.00;
→582;USD;108.740;;;1.087320537745000;1.087320537745000;;1544051;0.000;0.000;0.000;0.
→000;202701;;route by Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;;
1817526;2023-05-16T16:16:11.00;2023-05-16T16:16:11.00;16.05.2023;2023-05-16T00:00:00.
→00;2023-05-16T00:00:00.00;Demo merchant A;6655;1712;Demo Ams EUR;EUR;;Visa;85.132.52.
→251;chargeback;approved;;;11,1;Card Recovery Bulletin>manual;100.000;1,68424E+12;
→444455XXXXXX1111;TEST HOLDER;;;a***v@example.com;+3***12233;Wes***HX;***;Ams***en;;
→NLD;Firstname;Lastname;Test payment;test;1817525;100.000;;504362;502894;;;3,
→13665E+11;179416949;;;Yes;;;;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https:/
→/sandbox.doc2.com/paynet/view-receipt/6655/fdf7267b-876b-36b8-84a7-2fea7a3685e4;;0.
→000;;;UNKNOWN;;US;United States;;;Test processor;;;2023-05-16T16:15:13.00;2023-05-
→16T16:15:13.00;100.000;;2% cyprus;;;2023-05-16T16:16:11.00;582;USD;108.740;;;1.
→087320537745000;1.087320537745000;;1544053;0.000;0.000;0.000;0.000;202701;;route by
→Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;;
1817507;2023-05-16T16:17:24.00;2023-05-16T16:17:24.00;16.05.2023;;;Demo merchant A;
→9068;1712;Demo Ams EUR;EUR;;Any;85.132.52.251;sale;declined;920084;Request failed:
→Timer expired. User not paid. Request failed: Timer expired. User not paid.;;;100.
→000;;999999XXXXXX9999;Unknown holder;;;a***v@example.com;+3***12233;Wes***HX;***;
→Ams***en;;NLD;Firstname;Lastname;Test payment;test;1817503;;;;;;No;;;;;;
→4441;Test demo EUR;DEM03;;;https://sandbox.doc2.com/paynet/view-receipt/9068/
→7f44aae1-10fe-3f05-8271-d94f539fdf6a;;0.000;;;;;;Test processor;920084;;2023-05-
→16T16:17:24.00;2023-05-16T16:17:24.00;100.000;;2% cyprus;;;2023-05-16T16:17:24.00;
→582;EUR;100.000;;;1544054;0.000;0.000;0.000;0.000;202412;;route by Source Credit
→Card Type;ANY_CREDIT_CARD;Unknown;Unknown;76;test_manager;;
1817527;2023-05-16T16:28:20.00;2023-05-16T16:28:20.00;16.05.2023;;;Demo merchant A;
→9068;1712;Demo Ams EUR;EUR;;Any;85.132.52.251;sale;declined;920084;Request failed:
→Timer expired. User not paid. Request failed: Timer expired. User not paid.;;;100.
→000;;999999XXXXXX9999;Unknown holder;;;a***v@example.com;+3***12233;Wes***HX;***;
→Ams***en;;NLD;Firstname;Lastname;Test payment;test;1817525;;;;;;No;;;;;;
→4441;Test demo EUR;DEM03;;;https://sandbox.doc2.com/paynet/view-receipt/9068/
→33e28fb1-6443-30a1-bac2-d5d1648e38cd;;0.000;;;;;;Test processor;920084;;2023-05-
→16T16:28:20.00;2023-05-16T16:28:20.00;100.000;;2% cyprus;;;2023-05-16T16:28:20.00;
→582;EUR;100.000;;;1544078;0.000;0.000;0.000;0.000;202412;;route by Source Credit
→Card Type;ANY_CREDIT_CARD;Unknown;Unknown;76;test_manager;;
1824084;2023-05-26T14:44:38.00;2023-05-26T14:44:38.00;26.05.2023;;;Demo merchant A;
→6655;1712;Demo Ams EUR;EUR;;BANK WIRE;85.132.52.251;transfer;approved;;;;;55.550;1,
→6851E+12;;Test full name;7812013775;4,096E+19;j***h@gmail.com;7***29876;;;;;Test
→payment;inv9157586;inv9157586;;829695;;;;;No;;;;;2471;B MC CC Ams EUR;
→AMSTERDAM;;;https://sandbox.doc2.com/paynet/view-receipt/6655/43b0c106-0f10-307a-
→99f8-b3df4823337d;;0.000;;;;;Test processor;;2023-05-26T14:44:38.00;2023-05-
→26T14:44:38.00;55.550;;2% cyprus;;;2023-05-26T14:44:38.00;582;USD;59.580;;;1.
→072435511772000;1.072435511772000;;1549073;0.000;0.000;0.000;0.000;0;;route by
→Purpose;OTHERS;Unknown;Unknown;76;test_manager;2023-05-26T14:44:12.00;
1824082;2023-05-26T15:31:23.00;2023-05-26T15:31:23.00;26.05.2023;;;Demo merchant A;

```

(continues on next page)

(continued from previous page)

```

→6655;1712;Demo Ams EUR;EUR;;BANK WIRE;85.132.52.251;transfer;declined;1015;SECURE_
→3D_TIMEOUT;;;55.550;;;Test ful name;7812013775;4,096E+19;j***h@gmail.com;7***29876;
→;;;;;Test payment;inv9157586;inv9157586;;;;;3,14622E+11;;;Yes;;;;;
→2471;B MC CC Ams EUR;AMSTERDAM;;;https://sandbox.doc2.com/paynet/view-receipt/
→6655/e5b22b51-c51a-37e3-be9c-3ea581ccf583;;0.000;;;;;Test processor;timeout;;
→2023-05-26T15:31:23.00;2023-05-26T15:31:23.00;55.550;;2% cyprus;;;2023-05-
→26T15:31:23.00;582;USD;59.580;;1.072435511772000;1.072435511772000;;1549088;0.000;0.
→000;0.000;0.000;0;;route by Purpose;OTHERS;Unknown;Unknown;76;test_manager;2023-05-
→26T14:42:49.00;
1824083;2023-05-26T15:31:23.00;2023-05-26T15:31:23.00;26.05.2023;;;Demo merchant A;
→6655;1712;Demo Ams EUR;EUR;;BANK WIRE;85.132.52.251;transfer;declined;1015;SECURE_
→3D_TIMEOUT;;;500.000;;;Test;7812013775;4,096E+19;j***h@gmail.com;7***29876;;;;;
→34T43R77N;34T43R77N;;;;;3,14692E+11;;;Yes;;;;;2471;B MC CC Ams EUR;
→AMSTERDAM;;;https://sandbox.doc2.com/paynet/view-receipt/6655/b580d3c7-8ff1-3cb3-
→8292-87486edd18c0;;0.000;;;;;Test processor;timeout;;2023-05-26T15:31:23.00;2023-
→05-26T15:31:23.00;500.000;;2% cyprus;;;2023-05-26T15:31:23.00;582;USD;536.220;;1.
→072435511772000;1.072435511772000;;1549089;0.000;0.000;0.000;0.000;0;;route by
→Purpose;OTHERS;Unknown;Unknown;76;test_manager;2023-05-26T14:44:10.00;
1824985;2023-05-29T14:45:00.00;2023-05-29T14:45:00.00;29.05.2023;;;Demo merchant A;
→6655;1712;Demo Ams EUR;EUR;;Visa;188.253.224.81;sale;approved;;;;;100.000;PNTEST-
→1824985;444455XXXXX1111;TEST HOLDER;;a***v@example.com;+3***12233;Wes***HX;***;
→Ams***en;;NLD;Firstname;Lastname;Test payment;test;1824984;;115524;407275;;3,
→1496E+11;647547344;;Yes;;;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https://
→sandbox.doc2.com/paynet/view-receipt/6655/3317093d-b5b3-3d09-aa9b-28e1c80166d8;;2.
→200;;;UNKNOWN;;US;United States;;;Test processor;;;2023-05-29T14:45:00.00;2023-05-
→29T14:45:00.00;100.000;;2% cyprus;;;2023-05-29T15:07:30.00;582;USD;107.320;;1.
→072435511772000;1.072435511772000;;1550036;0.000;2.200;0.000;1.000;202701;;route by
→Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;2023-05-29T14:40:11.00;
1824987;2023-05-29T14:56:48.00;2023-05-29T14:56:48.00;29.05.2023;;;Demo merchant A;
→9068;1712;Demo Ams EUR;EUR;;Any;188.253.224.81;sale;declined;920084;Request failed:
→Timer expired. User not paid. Request failed: Timer expired. User not paid.;100.
→000;;999999XXXXX9999;Unknown holder;;a***v@example.com;+3***12233;Wes***HX;***;
→Ams***en;;NLD;Firstname;Lastname;Test payment;test;1824984;;No;;;;;
→4441;Test demo EUR;DEM03;;;https://sandbox.doc2.com/paynet/view-receipt/9068/
→e626abe4-52eb-3a25-ae93-19de6ca44023;;0.000;;;;;Test processor;920084;;2023-05-
→29T14:56:48.00;2023-05-29T14:56:48.00;100.000;;2% cyprus;;;2023-05-29T14:56:48.00;
→582;EUR;100.000;;;;;1550041;0.000;0.000;0.000;0.000;202412;;route by Source Credit
→Card Type;ANY_CREDIT_CARD;Unknown;Unknown;76;test_manager;2023-05-29T14:41:45.00;
1824985;2023-05-29T15:07:30.00;2023-05-29T15:07:30.00;29.05.2023;2023-05-29T00:00:00.
→00;2023-05-29T00:00:00.00;Demo merchant A;6655;1712;Demo Ams EUR;EUR;;Visa;188.253.
→224.81;retrieval;approved;;;;;manual;100.000;PNTEST-1824985;444455XXXXX1111;TEST
→HOLDER;;a***v@example.com;+3***12233;Wes***HX;***;Ams***en;;NLD;Firstname;Lastname;
→Test payment;test;1824984;;115524;407275;;3,1496E+11;647547344;;Yes;;;;;
→2470;A VISA CC Ams EUR;DEM01;;;test1;https://sandbox.doc2.com/paynet/view-receipt/
→6655/3317093d-b5b3-3d09-aa9b-28e1c80166d8;;0.000;;;;;UNKNOWN;;US;United States;;;Test
→processor;;;2023-05-29T14:45:00.00;2023-05-29T14:45:00.00;100.000;;2% cyprus;;;2023-
→05-29T15:07:30.00;582;USD;107.320;;1.073134088104000;1.073134088104000;;1550043;0.
→000;0.000;0.000;0.000;202701;;route by Customer Loyalty;OTHERS;Unknown;Unknown;76;
→test_manager;2023-05-29T14:40:11.00;
1826965;2023-05-31T16:42:20.00;2023-05-31T16:42:20.00;31.05.2023;;;Demo merchant A;
→6655;1712;Demo Ams EUR;EUR;;Visa;188.253.237.101;sale;approved;;;;;100.000;PNTEST-
→1826965;444455XXXXX1111;TEST HOLDER;;a***v@example.com;+3***12233;Wes***HX;***;
→Ams***en;;NLD;Firstname;Lastname;Test payment;test;1826964;;432862;200594;;3,
→15182E+11;302446749;;Yes;;;;;2470;A VISA CC Ams EUR;DEM01;;;test1;https://
→/sandbox.doc2.com/paynet/view-receipt/6655/861f0fd1-9f2e-3acf-9a05-b7f8c927eae3;;2.
→200;;;UNKNOWN;;US;United States;;;Test processor;;;2023-05-31T16:42:20.00;2023-05-

```

(continues on next page)

(continued from previous page)

```

↪31T16:42:20.00;100.000;;2% cyprus;;;2023-05-31T16:42:20.00;582;USD;107.220;;;1.
↪072140012909000;1.072140012909000;;1551928;0.000;2.200;0.000;1.000;202601;;route by
↪Customer Loyalty;OTHERS;Unknown;Unknown;76;test_manager;2023-05-31T16:37:25.00;
1826966;2023-05-31T16:53:50.00;2023-05-31T16:53:50.00;31.05.2023;;;Demo merchant A;
↪9068;1712;Demo Ams EUR;EUR;;Any;188.253.237.101;sale;declined;920084;Request
↪failed: Timer expired. User not paid. Request failed: Timer expired. User not paid.;
↪;;;100.000;;999999XXXXXX9999;Unknown holder;;;a***v@example.com;+3***12233;Wes***HX;
↪***;Ams***en;;NLD;Firstname;Lastname;Test payment;test;1826964;;;;;;;No;;;;;;
↪;;;4441;Test demo EUR;DEM03;;;https://sandbox.doc2.com/paynet/view-receipt/9068/
↪4e6c1674-aeb1-3d5e-bbe4-0af9c37fd95d;;0.000;;;;;;Test processor;920084;;2023-05-
↪31T16:53:50.00;2023-05-31T16:53:50.00;100.000;;2% cyprus;;;2023-05-31T16:53:50.00;
↪582;EUR;100.000;;;1551934;0.000;0.000;0.000;0.000;202412;;route by Source Credit
↪Card Type;ANY_CREDIT_CARD;Unknown;Unknown;76;test_manager;2023-05-31T16:38:47.00;

```

7.54.7 Fail Response Example

```

HTTP/1.1 403
Server: server
Date: Fri, 28 Jul 2023 12:44:12 GMT
Content-Type: text/html
Content-Length: 735
Connection: close

```

```

...
<body>
  <p>Access is denied</p>
</body>

```

7.54.8 Postman Collection

7.54.9 Request Builder

7.55 /services/v2/rpt-engine

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.55.1 Introduction

Custom transactions report provide an opportunity to get customized information about transactions with their statuses for verification and further analysis.

Custom transactions report is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#) specified below. Use [HMAC-SHA1](#) for authentication. Report parameters are being sent via JSON.

7.55.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet-ui/services/v2/rpt-engine	https://gate.doc2.com/paynet-ui/services/v2/rpt-engine

7.55.3 Request Parameters

Each custom report has a different JSON body parameters. JSON templates and supported formats for required custom reports could be provided by support managers upon request.

Request body example:

```
{
  "reportIdentifier": "exampleReport",
  "reportEncode": "UTF-8",
  "reportName": "example",
  "reportCsvDelimiter": ",",
  "reportFormat": "CSV",
  "configurationReportParameters": [],
  "inputReportParameters": [
    {
      "type": "INPUT_TEXT",
      "name": "example",
      "value": "example"
    }
  ],
  "outputReportParameters": [],
  "filters": {
    "gates": [],
    "companies": [],
    "projects": [],
```

(continues on next page)

(continued from previous page)

```
        "endpoints": [],
        "merchants": [],
        "processors": [],
        "managers": [],
        "resellers": [],
        "currencies": []
    }
}
```

7.55.4 Response Parameters

Each custom report has a different response body parameters. Usually it's provided in CSV or PDF format.

Note: The returned report has wellformed CSV format. In accordance with , fields containing line breaks (CRLF, CR, LF), double quotes, and commas are enclosed in double-quotes.

7.55.5 Request Example

```
POST /paynet-ui/services/v2/rpt-engine HTTP/1.1
Host: gate.doc2.com
User-Agent: curl/7.88.1
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="login",oauth_
→timestamp="1690553049",oauth_nonce="2lrWDbf0oq",oauth_signature_method="HMAC-SHA1",
→oauth_signature="ZAI2xFUlh35DAzDYKnSbyUUGGIA%3D"
Content-Type: application/json
Content-Length: 417
Connection: close

{"reportIdentifier":"exampleReport","reportEncode":"UTF-8","reportName":"example",
→"reportCsvDelimiter":",","reportFormat":"CSV","configurationReportParameters":[],
→"inputReportParameters":[{"type":"INPUT_TEXT","name":"example","value":"example"}],
→"outputReportParameters":[],"filters":{"gates":[],"companies":[],"projects":[],
→"endpoints":[],"merchants":[],"processors":[],"managers":[],"resellers":[],
→"currencies":[]}]}
```

7.55.6 Success Response Example

```
"Merchant_Name","Transaction_id","Merchant_OID","Created_date","Amount","Currency",
↪ "Transaction_type","Order_description","Date_of_report"
"TestMerchant3","6891747","902B4FF5","2023-07-27 16:37:21","90.000","USD","Sale",
↪ "Order #293927","28.07.2023"
```

7.55.7 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Mon, 31 Jul 2023 09:49:07 GMT
Content-Type: text/plain
Content-Length: 46
Connection: close
Strict-Transport-Security: max-age=31536000

You are not authorised to perform such request
```

7.55.8 Postman Collection

7.55.9 Request Builder

7.56 /view-invoice

- [Introduction](#)
- [API URLs](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.56.1 Introduction

To start Swift transaction use [/api/v2/auth-cash/](#) and get receipt-id in [order status response](#). Swift invoice in PDF is initiated through HTTPS GET request by using [URLs](#) specified below. Use [HMAC-SHA1](#) for authentication. See [Statuses](#).

7.56.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/view-invoice/ENDPOINTID/RECEIPTID	https://gate.doc2.com/paynet/view-invoice/ENDPOINTID/RECEIPTID

7.56.3 Request Example

```
POST /paynet/view-invoice/6657/a5e8f278-8936-3286-aabb-c8960a5d04f4 HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="test_merch",
  ↳oauth_timestamp="1687518156",oauth_nonce="EozU8eCdyB0",oauth_signature_method="HMAC-
  ↳SHA1",oauth_signature="bNQNg2CLuGo0rf5eji3ffj96bC0%3D"
Content-Length: 140
Content-Type: application/x-www-form-urlencoded
Connection: close

oauth_consumer_key=test_merch
&oauth_nonce=EozU8eCdyB0
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1687518156
&oauth_version=1.0
```

7.56.4 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Fri, 23 Jun 2023 11:03:05 GMT
Content-Type: application/pdf;charset=utf-8
Connection: close
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000
Content-Length: 34929

...
*PDF content*
```

7.56.5 Fail Response Example

```
HTTP/1.1 200
Server: server
Date: Fri, 23 Jun 2023 12:23:15 GMT
Content-Type: application/pdf;charset=utf-8
Connection: close
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000
Content-Length: 34929

...
<body>
  <p>Sorry, that page doesn't exist!</p>
</body>
```

7.56.6 Postman Collection

7.56.7 Request Builder

7.57 /view-invoice-xml

- [Introduction](#)
- [API URLs](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.57.1 Introduction

To start Swift transaction use [/api/v2/auth-cash/](#) and get receipt-id in [order status response](#). Swift invoice in XML is initiated through HTTPS GET request by using [URLs](#) specified below. Use [HMAC-SHA1](#) for authentication. See [Statuses](#).

7.57.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/view-invoice-xml/ENDPOINTID/RECEIPTID	https://gate.doc2.com/paynet/view-invoice-xml/ENDPOINTID/RECEIPTID

7.57.3 Request Example

```
POST /paynet/view-invoice-xml/6657/6d0442b3-f4f9-3264-8cfd-f811195b693a HTTP/1.1
Host: sandbox.doc2.com
User-Agent: curl/7.83.0
Accept: */*
Authorization: OAuth realm="",oauth_version="1.0",oauth_consumer_key="test_merch",
  ↳oauth_timestamp="1687422393",oauth_nonce="I4FlRMNDfFe",oauth_signature_method="HMAC-
  ↳SHA1",oauth_signature="sw7XXVhkxXAwE61CLnwCpJxJpD0%3D"
Content-Length: 140
Content-Type: application/x-www-form-urlencoded
Connection: close

oauth_consumer_key=test_merch
&oauth_nonce=I4FlRMNDfFe
&oauth_signature_method=HMAC-SHA1
&oauth_timestamp=1687422393
&oauth_version=1.0
```

7.57.4 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Thu, 22 Jun 2023 08:26:51 GMT
Content-Type: text/xml; charset=utf-8
Content-Length: 833
Connection: close
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Strict-Transport-Security: max-age=31536000

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root>
<pdf_template_name>swift_local.xslt</pdf_template_name>
<additional_params/>
<date>22.06.2023</date>
<bank_name>TestBank</bank_name>
<bank_address>Lima, Peru</bank_address>
<company_name>Company</company_name>
<company_address>P. Sherman 42 Wallaby way</company_address>
```

(continues on next page)

(continued from previous page)

```

<iban>1122334455</iban>
<swift>123IAMS123</swift>
<customer>
<firstname>John</firstname>
<lastname>Smith</lastname>
<address>98102, United States, WA, Seattle, 100 Main st</address>
<phone>12063582043</phone>
<cell_phone>19023384543</cell_phone>
</customer>
<rate>1</rate>
<quantity>1</quantity>
<currency>EUR</currency>
<amount>10</amount>
<order_desc>Test Order Description</order_desc>
<invoice_no>6657-587</invoice_no>
<order_invoice_no>902B4FF5</order_invoice_no>
<reference_id>1</reference_id>
</root>

```

7.57.5 Fail Response Example

```

HTTP/1.1 404
Server: server
Date: Thu, 22 Jun 2023 08:30:18 GMT
Content-Type: text/xml;charset=utf-8
Content-Length: 750
Connection: close
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000

...
<body>
  <p>Sorry, that page doesn't exist!</p>
</body>

```

7.57.6 Postman Collection

7.57.7 Request Builder

7.58 /api/3ds/v1/upload-cres-result

- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)

- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

After receiving cRes value to notificationURL, upload the results using HTTP POST request to [URLs](#) and the [parameters](#) specified below.

7.58.1 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/3ds/v1/upload-cres-result/ENDPOINTID	https://gate.doc2.com/paynet/api/3ds/v1/upload-cres-result/ENDPOINTID
https://sandbox.doc2.com/paynet/api/3ds/v1/upload-cres-result/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/3ds/v1/upload-cres-result/ENDPOINTGROUPID

7.58.2 Request Parameters

Parameter Name	Max Length	Description
cRes	4096	The CRes message is the ACS response to the CReq message. It indicates the result of the Cardholder authentication.
orderid	128	Order id assigned to the order by Doc2.0.
login	20	Merchant login name.
client-order-id	128	Merchant order identifier.
control	40	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. login 2. client_orderid 3. orderid 4. merchant_control (Control key assigned to Merchant account in the Doc2.0 gateway system)

7.58.3 Response Parameters

Parameter Name	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Merchant order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Merchant.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.

7.58.4 Request Example

```
POST /paynet/api/3ds/v1/upload-cres-result/39529 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 495
Content-Type: application/x-www-form-urlencoded
Connection: close

cres=ewogICJtZXNzYWdlVHlwZSIgICAgOiAiQ1JlcyIsCiAgIm1lc3NhZ2VWZXJzaW9uIiA6IClYlEuMCIsCgogICJ0aHJlZUR
&orderid=6684126
&control=d0539beedb9a7266547530c8d2027f2b14503a61
&login=test1
&client-order-id=1e21f041-871d-4642-a6d9-968457df47a0
```

7.58.5 Success Response Example

```
HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 11:03:17 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=async-response
&serial-number=00000000-0000-0000-0000-000002d6202d
&merchant-order-id=1e21f041-871d-4642-a6d9-968457df47a0
&paynet-order-id=6684126
```

7.58.6 Fail Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 11:03:17 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=validation-error
&serial-number=000000000-0000-0000-0000-000002b36f64
&merchant-order-id=inv4097763
&error-message=End+point+with+id+22903+not+found
&error-code=3

```

7.58.7 Postman Collection

7.58.8 Request Builder

7.59 /api/3ds/v1/upload-method-url-result

- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

After completing the 3DS Method and gathering customer browser fingerprint, upload the results using HTTP POST request to [URLs](#) and the [parameters](#) specified below.

7.59.1 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/3ds/v1/upload-method-url-result/ENDPOINTID	https://gate.doc2.com/paynet/api/3ds/v1/upload-method-url-result/ENDPOINTID
https://sandbox.doc2.com/paynet/api/3ds/v1/upload-method-url-result/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/3ds/v1/upload-method-url-result/ENDPOINTGROUPID

7.59.2 Request Parameters

Parameter Name	Length	Description
orderid	max 128	Order id assigned to the order by Doc2.0.
login	max 20	Merchant login name.
client-order-id	max 128	Merchant order identifier.
control	max 40	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. login 2. client_orderid 3. orderid 4. merchant_control (Control key assigned to Merchant account in the Doc2.0 gateway system)
threeDSServer-TransID	max 36	3DS Server Transaction ID. Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction.
browserAccept-Header	max 2048	Browser Accept Headers. Exact content of the HTTP accept headers as sent to the 3DS Requestor from the Cardholder's browser.
browserIP	max 45	Browser IP Address.*
browserJavaEnabled	min 4, max 5	true/false Browser Java Enabled. Boolean that represents the ability of the cardholder browser to execute java. Value is returned from the navigator.javaEnabled property.
browser-JavascriptEnabled	min 4, max 5	true/false Browser JavaScript Enabled. Boolean that represents the ability of the cardholder browser to execute javascript.
browserLanguage	max 8	Value representing the browser language as defined in IETF BCP47. Returned from navigator.language property.

continues on next page

Table 211 – continued from previous page

Parameter Name	Length	Description
browserColorDepth	max 2	Browser Screen Color Depth. Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from Cardholder browser using the screen.colorDepth property
browserScreenHeight	max 6	Browser Screen Height. Total height of the cardholder's screen in pixels. Value is returned from the screen.height property.
browserScreenWidth	max 6	Browser Screen Width. Total width of the cardholder's screen in pixels. Value is returned from the screen.width property.
browserTZ	max 5	Browser Time Zone. Time difference between UTC time and the cardholder browser local time, in minutes. Value is returned from the getTimezoneOffset() method.
browserUserAgent	max 2048	Browser User-Agent. Exact content of the HTTP user-agent header. Note: If the total length of the User-Agent sent by the browser exceeds 2048 characters, the 3DS Server truncates the excess portion.
threeD-SCompInd	max 1	3DS Method Completion. Indicates whether the 3DS Method was successfully completed. Values accepted: Y = Successfully completed N = Did not successfully complete. The 3DS Method was not completed within 10 seconds. U = Unavailable — tds-method-url-frame-3ds-method-url was not present in Status response.
notificationURL	max 256	Fully qualified URL of merchant system that will receive the CRes message or Error Message. The CRes message is posted by the ACS through the Cardholder browser at the end of the challenge.

Note:

* IP address of the browser as returned by the HTTP headers to the 3DS Requestor IPv4 address is represented in the dotted decimal format of 4 sets of decimal numbers separated by dots.

The decimal number in each and every set is in the range 0 to 255.

Example IPv4 address: 1.12.123.255

IPv6 address is represented as eight groups of four hexadecimal digits, each group representing 16 bits (two octets). The groups are separated by colons (:).

Example IPv6 address: 2011:0db8:85a3:0101:0101:8a2e:0370:7334

7.59.3 Response Parameters

Parameter Name	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details.
paynet-order-id	Order id assigned to the order by Doc2.0.
merchant-order-id	Merchant order id.
serial-number	Unique number assigned by Doc2.0 server to particular request from the Merchant.
error-message	If status is error this parameter contains the reason for decline or error details.
error-code	The error code is case of error status.
end-point-id	Endpoint id used for the transaction.

7.59.4 Request Example

```

POST /paynet/api/3ds/v1/upload-method-url-result/39529 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 721
Content-Type: application/x-www-form-urlencoded
Connection: close

notificationURL=https://merchant.com/shop/notification/1e21f041-871d-4642-a6d9-
↪968457df47a0
&browserLanguage=en-us
&orderid=6684126
&browserColorDepth=24
&browserJavaEnabled=false
&control=24dc828de77bae41b9c4bf4dcbdb5c80158eb2db
&login=test_merchant
&browserIP=10.25.8.19
&browserUserAgent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6) AppleWebKit/605.1.
↪15 (KHTML, like Gecko) Version/14.0 Safari/605.1.15
&browserAcceptHeader=text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
&browserTZ=-180
&browserScreenHeight=900
&browserJavascriptEnabled=true
&threeDSCompInd=Y
&browserScreenWidth=1440
&client-order-id=1e21f041-871d-4642-a6d9-968457df47a0
&threeDSServerTransID=d00a7afe-2a4d-46b4-8f7f-f4efadf142b3

```

7.59.5 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 11:03:17 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=async-response
&serial-number=00000000-0000-0000-0000-000002d62029
&merchant-order-id=1e21f041-871d-4642-a6d9-968457df47a0
&paynet-order-id=6684126

```

7.59.6 Fail Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 11:03:17 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=validation-error
&serial-number=00000000-0000-0000-0000-000002b36f64
&merchant-order-id=inv4097763
&error-message=Endpoint+with+id+22903+not+found
&error-code=3

```

7.59.7 Postman Collection

7.59.8 Request Builder

7.60 /api/3ds/v1/upload-pares-result

- [Upload PaRes Result](#)
- [API URLs](#)
- [Request Parameters](#)
- [Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Postman Collection](#)
- [Request Builder](#)

7.60.1 Upload PaRes Result

After receiving paRes value to TermUrl, upload the results using HTTP POST request to [URLs](#) and the [parameters](#) specified below.

7.60.2 API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/api/3ds/v1/upload-pares-result/ENDPOINTID	https://gate.doc2.com/paynet/api/3ds/v1/upload-pares-result/ENDPOINTID
https://sandbox.doc2.com/paynet/api/3ds/v1/upload-pares-result/ENDPOINTGROUPID	https://gate.doc2.com/paynet/api/3ds/v1/upload-pares-result/ENDPOINTGROUPID

7.60.3 Request Parameters

Parameter Name	Max Length	Description
paRes	10240	Payer Authentication Response message is returned by the ACS with the result of cardholder payment authentication.
orderid	128	Order id assigned to the order by Doc2.0.
login	20	Merchant login name.
client-order-id	128	Merchant order identifier.

continues on next page

Table 214 – continued from previous page

Parameter Name	Max Length	Description
control	40	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. login 2. client_orderid 3. orderid 4. merchant_control (Control key assigned to Merchant account in the Doc2.0 gateway system)

7.60.4 Response Parameters

Parameter Name	Description
type	The type of response. May be async-response, validation-error, error. If type equals validation-error or error, error-message and error-code parameters contain error details
paynet-order-id	Order id assigned to the order by Doc2.0
merchant-order-id	Merchant order id
serial-number	Unique number assigned by Doc2.0 server to particular request from the Merchant
error-message	If status is error this parameter contains the reason for decline or error details
error-code	The error code is case of error status

7.60.5 Request Example

```
POST /paynet/api/3ds/v1/upload-pares-result/39529 HTTP/1.1
User-Agent: curl/7.83.0
Accept: */*
Content-Length: 2631
Content-Type: application/x-www-form-urlencoded
Connection: close
```

```
orderid=6684127
```

```
&
```

```
→md=BB4A517A42736975764E3453302B303144794D754B7A4F697A7953765930784259503870546A79416F6542646F4E4A6
```

```
&paRes=eJydlluTokgThu/nVxh0xN4YPZwV5rPdKI6CgHIQxTsaEDnIWUB//
```

```
→aL09Ngdc7HfcgFFkvnwZmVWRc3/7s7JqPHLKszS1zHyAx6P/
```

```
→NTNvDANXsdbk38hx60qdLLPSbLUfx2n2fjvxdw8lb7PGr57Kf3Ft9FoNFf8qnICfxR6r2P4/
```

```
→Xr5dUM+Rv01JY6edxw/wu6RG6D71f8fdw99173oZf9A59Cv1ycPxS/
```

```
→dk5PWT7a72XELWlQX0IJMCXI0vb9+9jn7pcguEBTDiemMp0A5NFie6NAf8PPN5W6r/
```

```
→C+4LvQWamTjKstdVZhRuPwk8vv0kKPDbt2A1z109/gc4zm1v0BhFEFgdDaC0Z8I/
```

```
→BPvE33YP7v21Sj7w10XUxybQx9vn538Lu+L2MvtGR/jzy75XT04Z5f+C3Kf9jn0bHp0/
```

```
→g+JznMnXcxQjIBRLjgRBIgi6Gw6JQmSmvU2pJ/ru8dTgLn/
```

```
→IqA0z59zxqmfBDGHHvbPrn1f1pdqYc+h99GXbN1wAfeR9+ez7t+/7H04N9/QwUYYPd219Ef9kkir1/
```

```
→Gprv0fENS27Y8W+5GVAYT28wHBFNQ7eFUYfB8vHlG+J6bHbDFnnDRLQ9dJwptT942o+PUp80YgCbIyrE/
```

(continues on next page)

(continued from previous page)

```

→nPyFN/U5FIJ1jXnrSi4vg6cVdAmMIMYYWv2X9G9pXgWXLvFQnB7mDdP/o35vCH2118XX8/d+ttjkbBn5V/
→5ef//rxQLCc50IvVLFd0tZGL5PaWkMNtmqIDYDqdQnh/
→Qp49pxDH4L78fM0f8zI4ChXgtEZhyIt0K6WdJy5vhUC2dnwKScKue1cfDNt8ZLKzFaheInrfQNTd44xWZaSiTh2q+mWZ6gTqN0
→fEex/
→3lQZhlr2sg0Iutoh2tlv0KiBsCyuaDTx1lifdz72tLNgXrcJ8xvBpouWza+G0+DJwILpyCkrN+Qll5idnIIoYvK3WmLDCzukJmx
→UAs5KVhozuIn/oB7iH224yPFBviFW7Ke911RebQR0leX4diPBVgVvKvQ2X2BEyxTu0MI8Yv6/
→DYr4d+s1JEkd2yDA0igGE05mY62yk6YW+HG1DpIC50cShQLUwDreIBSxeKVrWMZr0WpglcK9k7lpMVEAsA2XKDXAa0zI4P0s4E
→xTU5XAPmgMW0rDkRLsG4eb12cVdZxEdAGqQIw1oNK25jSKibw+Ra+9k9CYeNOMblbXwDnYYt+2wbi/
→YsiZB1zA9JAs02Qn0yn7EQWPLL70EP3eh5KBDX3lnHHS8C4xw5EGmQKgzz0sG8o0to7os+Ua9n2QZNZcHpkau2I3GM5Q6HBgwr
→t2b0tQicD8CaARoJ7l5Ms0rHHECQws5DYVJaBy5M0jzbe3GL+JjH0NaK4Fy53xoGorGKpfqYi2wDgSu8v8UJFC831Zt8pRhCQm
→GAdaCWleaXgSpFgGdN0Qbd+Q69YZbilJEWwVfjTc3EK0XrvULeio4woHojt4TpDqBPxpsrTmshXpLnd8Yl7zWaYpdRtgFsklpD
→btHbphr2inYQRgqhY0Cw/LI03wfCCIxxCFdvcWKEHJ2A7+ibdlSxtIGtn0zW5QMRa/
→vm0ACtAFhgjEIwxDeM1TiaabcA4GJfKZpIG87BcaqY6da6ZByDulx1WyHI2yTh4BsxwQz5LYowfiBGhcupZ4/
→zk9UVEgQLKzB3nZS7xGFX2rr0upaLkq00QSZr0loiql4yjeBsjpJtxlvnyGIRfyn1fYutdu8dnjNxxkqCqvAq3pmDlR+QApwpWo
→gGdswlzoEJGK0tqk21NaVfbkftHLUCkfLyIBY8oyi1K2epL/
→GqPbrttVQamLLcDWsk28Su39d1UyRkgxZ2sybgTC+9DK4PRiNL0yKZGmZ18qMqv+jQ8lozEKVHDELWuJpSvHWeuSzC9eekmaqJ
→XfXPyE+zsWLb/254/nQ/A8pzm2f
&control=987e7aa6630a862805c65eee02392d7a769cadbe
&login=test_merchant
&client-order-id=e690bf9f-0c90-4073-a30a-604d13ae9fb9

```

7.60.6 Success Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 11:03:17 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding
X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=async-response
&serial-number=00000000-0000-0000-0000-000002d62036
&merchant-order-id=e690bf9f-0c90-4073-a30a-604d13ae9fb9
&paynet-order-id=6684127

```

7.60.7 Fail Response Example

```

HTTP/1.1 200
Server: server
Date: Wed, 17 Nov 2021 11:03:17 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=60
Vary: Accept-Encoding

```

(continues on next page)

(continued from previous page)

```

X-XSS-Protection: 1
Strict-Transport-Security: max-age=31536000
Content-Language: ru-RU
P3P: CP="NOI ADM DEV COM NAV OUR STP"
Content-Encoding: gzip

type=validation-error
&serial-number=00000000-0000-0000-0000-000002b36f64
&merchant-order-id=inv4097763
&error-message=End+point+with+id+22903+not+found
&error-code=3

```

7.60.8 Postman Collection

7.60.9 Request Builder

7.61 CRes Redirect

- [CRes Redirect Parameters](#)
- [Redirect Request Example](#)

7.61.1 CRes Redirect Parameters

CRes redirect to connecting party's website is performed when a payer ends Challenge flow. Final redirect is initiated from within payers browser using HTTP POST method. HTTP payload contains CRes parameters which in turn contains base64 encoded json (see example below).

Redirect parameter name	Description
serial-number	Unique number assigned by doc2 server to particular request from the Merchant
merchant-order-id	Merchant order id
paynet-order-id	Order id assigned to the order by gate.doc2.com

7.61.2 Redirect Request Example

Connecting party receives HTTP POST request base64 encoded **CRes** parameter similar to below:

```
cres=ewogICJtZXNzYWdlVHlwZSIgICAgOiAiQ1JlcyIsCiAgIm1lc3NhZ2VWZXJzaW9uIiA6ICIyLjEuMCIsCgogICJ0aHJlZUR
```

After decoding cres one will find a json similar to below:

```
{
  "messageType"      : "CRes",
  "messageVersion"   : "2.1.0",

  "threeDSServerTransID" : "someThreeDSServerTransID",
  "acsTransID"         : "someAcTransID",

  "challengeCompletionInd" : "Y",
  "transStatus"          : "Y"
}
```

7.62 PaRes Redirect

- [PaRes Redirect Parameters](#)
- [Redirect Request Example](#)

7.62.1 PaRes Redirect Parameters

PaRes redirect to connecting party's website is performed when a payer ends 1.0.2 authentication flow. Final redirect is initiated from within payers browser using HTTP POST method. HTTP payload contains PaRes parameters which in turn contains base64 encoded json (see example below).

Redirect parameter name	Description
serial-number	Unique number assigned by doc2 server to particular request from the Merchant
merchant-order-id	Merchant order id
paynet-order-id	Order id assigned to the order by gate.doc2.com

7.62.2 Redirect Request Example

Connecting party receives HTTP POST request deflate archived base64 encoded PaRes parameter similar to below:

```
paRes=eJydlluTokgThu/nVxh0xN4YPZwV5rPdKI6CgHIQxTsaEDnIWUB//aL09Ngdc7HfcgFFkvnwZmVWRc3/
↪7s7JqPHLKszS1zHyAx6P/
↪NTNvDANXsdbk38hx60qdLLPSbLUfx2n2fjvxdw8lb7PGr57Kf3Ft9FoNFf8qnICfxR6r2P4/
↪Xr5dUM+Rv01JY6edxw/wu6RG6D71f8fdw99173oZf9A59Cv1ycPxS/
↪dk5PW7a72XELWlQX0IJMCXI0vb9+9jn7pcguEBTDiemMp0A5NFie6NAf8PPN5W6r/
↪C+4LvQWamTjKstdVZhrUPwk8vv0kKPDbt2A1zl09/gc4zm1v0BhFEFgdDaC0Z8I/
↪BPvE33YP7v2LSj7wl0XUxybQx9vn538Lu+L2MvtGR/jzy75XT04Z5f+C3Kf9jn0bHp0/
↪g+JznMnXcxQjIBRLJgRBIGi6Gw6JQmSmvU2pJ/ru8dTgLn/
↪IqA0z59zxqmfBDGHHvbPrn1f1pdqYc+h99GXbN1wAfeR9+ez7t+/7H04N9/QwUYypD2l9Ef9kkir1/
↪Gprv0fENS27Y8W+5GVAYT28wHBFNQ7eFUYfB8vHlG+J6bHbDFnnDRLQ9dJwptT942o+PUUp80YgCbIyrE/
↪nPyFN/U5FIJ1jXnrSi4vg6cvdAmMIMYYWv2X9G9pXgWxlVfQnB7mDdP/o35vCH2118XX8/d+ttjkbBn5V/
↪5ef//rxQLCc50IvVLFd0tZGL5PaWkMNTmqIDYDqdQnh/
```

(continues on next page)

(continued from previous page)

```

→ Qp49pxDH4L78fM0f8zI4ChXgtEZhyIt0K6WdJy5vhUC2dnwKScKue1cfDNt8ZlKzFaheInrfQntd44xWZaSiTh2q+mWZ6gTqN0
→ fEex/
→ 3lQZhlr2sg0Iutoh2tlv0KiBsCyuaDTx1lfdz72tLnGxrcJ8xvBpouWza+G0+DJwILpyCkrN+Qll5idnIIoYvK3WmLDCzukJmx
→ UAs5KVhozuIn/oB7iH224yPfBviFW7Ke911RebqRQleX4diPBVgvvKvQ2X2BEyxTu0MI8Yv6/
→ DYr4d+s1JEkd2yDA0igGE05mY62yk6YW+HG1DpIC50cShQLUwDreIBSxeKVrWMZr0WpglcK9k7lpMVEAsA2XKDXAa0zI4P0s4E
→ xTU5XAPmgMW0rDkRLsG4eb12cvdZxEdAGqqIwloNK25jSKibw+Ra+9k9CYeN0Mb1bXwDnYYt+2wbi/
→ YsiZB1zA9JAs02Qn0yn7EQWPLL70EP3eh5KBDX3lnHHs8C4xw5EGmQKgzz0sG8o0to7os+Ua9n2QZNZcHpkau2I3GM5Q6HBgwr
→ t2b0tQIcd8CaARoJ7l5Ms0rHHECQws5DYVJaBy5M0jzbe3GL+JjH0NaK4Fy53xoGorGKpfqYi2wDgSu8v8UJFC831Zt8pRhCQm
→ GAdaCWIEaXgSpFgGdN0Qbd+Q69YZbilJEWWvfjTc3EK0XrvULeio4woHojt4TpDqBPxpsrTmshXpLnd8Yl7zWaYpdRtgFsklpD
→ btHbphr2inYQRgqhY0Cw/LI03wfcCIxxCFdvcWKEHZJ2A7+ibdlSxtIGtn0zW5QMRA/
→ vm0ACTAFhgjEIwxDeM1TiaabcA4GJfKZpIG87BcaqY6da6ZByDulx1WyHI2yTh4BsxxwQz5LYowfiBGhcupZ4/
→ zk9UveGQLKzB3nZS7xGFX2rr0upaLkq00QSZr0loiqL4yjeBsJpJtxlvnyGIrfyn1fYutdu8dnjNxxkqCqvAq3pmDlR+QApwpWo
→ gGdswlzoEJGK0tqk21NaVFbkfthLUckfLyIBy8oyi1K2epL/
→ GqPbrttVQamLLcDWSk28Su39d1UyRkgxZ2sybgTC+9DK4PRiNL0yKZGmZ18qMqv+jQ8lozEKVHDELWuJpSvHWeuSzC9eekmaqJ
→ XfXPyE+zsWLb/254/nQ/A8pzm2f

```

After decoding paRes one will find a xml similar to below:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ThreeDSecure>
  <Message id="00000000-0000-0001-0000-00000065fddf">
    <PARes id="00000000-0000-0001-0000-00000065fddf">
      <version>1.0.2</version>
      <Merchant>
        <acqBIN>411658</acqBIN>
        <merID>1234567890</merID>
      </Merchant>
      <Purchase>
        <xid>NjY4NDEyN0CMEphIFXxSLjZW0vA=</xid>
        <date>20211027 02:10:42</date>
        <currency>643</currency>
        <exponent>2</exponent>
        <purchAmount>10000</purchAmount>
      </Purchase>
      <pan>7235022575552227668589750218</pan>
      <TX>
        <time>20211027 02:49:55</time>
        <status>Y</status>
        <eci>05</eci>
      </TX>
    </PARes>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-
→ c14n-20010315"/>
        <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1
→ "/>
        <Reference URI="#00000000-0000-0001-0000-00000065fddf">
          <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <DigestValue>NIwHBVPRLltV0/v3Kv5PA/t0r/4=</DigestValue>
        </Reference>
      </SignedInfo>
      <SignatureValue>LsGSxSZqnq2xtJR4CybqG8xY0hp5qLwxc4P6w47N57KiIuktXv/
→ UWaS+HrJT1aYwQRVdSN+/wPGD&#13;
NL0/wDuWA1NKWlQmevE/v/3oksB2ld0hRepdULGYuQs3eCSeB5wFoyqaw4Hga/Bsaq9NpF+9Tpha&#13;
I30Us03GK3mnh37q/Moa8YoLBC5cv7zAYjRi43AP+sUCdjZ4kZSAKqIRW30s+2Lyn8kuD91GWKq&#13;

```

(continues on next page)

(continued from previous page)

```

DmBeZ2JlG9MTNoLCRAKEtRNe++pR+G+ix2WD//kNq0J+5BpiJHwcanHodIGToQzn17cAM2E387gX&#13;
qLHSY2W/F6eg4d1Yb7FjeY/FME6FdxYK8pNuMg==</SignatureValue>
  <KeyInfo>
    <X509Data>
      <X509Certificate>
        MIIDUDCCAjgCCQCzTaU62+DzZzANBgkqhkiG9w0BAQsFADBQMQswCQYDVQQGEwJYWDELMAkGA1UE&#13;
        CAwCWFgXETAPBgNVBACMCFRlc3RjaXR5MRQwEgYDVQQKDAUZXN0Q29tcGFueTERMA8GA1UECwwI&#13;
        VGVzdFVuaXQxEjAQBgNVBAMMCVRlc3RBY3MwMTAeFw0yMTA5MDkxMTEzMTNaFw0yMjA5MDkxMTEz&#13;
        MTNaMGoxCzAJBgNVBAYTA1hYMQswCQYDVQQIDAjYWDERMA8GA1UEBwwIVGVzdG9pdHkxFDASBgNV&#13;
        BAoMC1Rlc3RDb21wYW55MREwDwYDVQQLEDAhUZXN0VW5pdDESMBAGA1UEAwwJVGVzdEFjczAxMIIB&#13;
        IjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAl1qYpiG+rVZEilx4oXdkwle3dCBVK5EcLlgo&#13;
        SKkjtfpIDv/Ay0XzkL/kHPsblY9C5J1kcxqT04oRIgTUQGuaV90TgsYw9GH7g/8NZIbEM06jBwLW&#13;
        l8NnAY0wqISaYsNLUjYJB4ZVZBMgkFMqErI5DCTTcAx0b2y3sScrJJjVVqe6mqhiEBXa9zqx9Sjk&#13;
        wZy719h5bNL6t5pK8TUWFLcyo73VMtwg4V83v8AVSBxlVAA1VtFcvAvYLRt+yx81SkVnH2X3NuJC&#13;
        hQ0evLY0+HUv0X7PwYrciTcAWJ35GIMV/7pZHSbGSdGF0i/sDvzDGX3NoxG4WBzQz0qBS3Kaozxp&#13;
        3wIDAQABMA0GCSqGSIb3DQEBChUA4IBAQB5nvEa449q7RV0rCaSEuyRYM58z+lE0z5+3SLbjj3F&#13;
        jqcENmdEelKyAd8w0M0Ey+rWLaDKQ0toxwEjLUJ+1+08VH1NRrCvGaPfJYTkUafD3KeHJQYD3KWz&#13;
        pCkll2NLKIuTGVPf1Z0nM3s2SkYAs+1c9Pf+oXN7liu/yINDv4N8h0LjkpH83YPXJ92VbQw9uyV&#13;
        j9ePJQPeS3fpJEe05k2tV9wnQ09Qjcx9WGOM/8LHZGZdfIJwKaHRH4swfcwyrMv09VcP/SnULYtI&#13;
        vql8v2qYv050oRrdo0tZSvLJW5l6STshejspuR/HytC/9RjC18t4Nn9FVm7cD1Expo7NQIKt</
        </X509Certificate>
      </X509Data>
    </KeyInfo>
  </Signature>
</Message>
</ThreeDSecure>

```

7.63 Final Redirect

- [Final Redirect Parameters](#)
- [Final Redirect Request Example](#)

7.63.1 Final Redirect Parameters

Final redirect to merchant (represented by Connecting Party) website is performed when Payer ends payment flow. Final redirect is initiated from within Payer's browser using HTTP POST method.

Redirect parameter name	Description
status	See Status List for details.
orderid	Order id assigned to the order by gate.doc2.com.
merchant_order	Connecting Party order id.
client_orderid	Connecting Party order id.
error_message	If status is declined or error this parameter contains the reason for decline or error details.

continues on next page

Table 218 – continued from previous page

Redirect parameter name	Description
control	Checksum used to ensure that it is gate.doc2.com (and not a fraudster) that initiates the request. This is SHA-1 checksum of the concatenation status + orderid + client_orderid + merchant-control.
descriptor	Gate descriptor.
processor-tx-id	Acquirer transaction identifier.
amount	Actual transaction amount.
bin	Bank BIN of customer credit card number.
type	The type of response.
card-type	Type of customer credit card.
phone	Customer phone.
last-four-digits	Last four digits of customer credit card number.
card-holder-name	Cardholder name.
error_code	Error Code.

Connecting Party should not use Payer redirect as a main source of information about transaction's final status. If Connecting Party has passed server_callback_url in initial transaction request, Payment Gateway will call this URL. Connecting Party may use it for custom processing of the transaction completion, e.g. to collect sales data in Connecting Party database. The parameters sent to this URL are specified in [Connecting Party Callbacks](#).

7.63.2 Final Redirect Request Example

Payment Gateway sends POST redirect to https://connectingparty.com/pay/redirect?token=some_token with params in x-www-form-urlencoded format:

```
error_message=null
&processor-tx-id=PNTEST-1788082
&amount=100.00
&merchant_order=test
&orderid=1788082
&client_orderid=test
&bin=444455
&control=854f38715ee6bf43fa18e3ca202985a06e0ae755
&gate-partial-reversal=enabled
&descriptor=DEM01
&gate-partial-capture=enabled
&type=sale
&card-type=VISA
&phone=%2B31201112233
&last-four-digits=1111
&card-holder-name=TEST+HOLDER
&status=approved
```

7.64 Mobile Device Connecting Party Server API

7.64.1 Initiate verification

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Initiate verification is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

To initiate card verification, Connecting Party's app sends {accessToken} with other device parameters to Connecting Party's server. These parameters are used to start a session with unique random {nonce} and encrypted {signature}. In response Connecting Party's server also provides information about {invoiceId} in Connecting Party's system and {endpointId} in Doc2.0.

API URLs

Note:

This is an example of URL format on Connecting Party server.

Connecting Party Server
https://proxy.connectingpartyserver.com/verification/initiate-verification

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
transaction		Necessity: Required Type: Object
transaction.currency	Upper case letters (ISO 4217 alpha code).	Necessity: Required Type: String Length: 3
session		Necessity: Required Type: Object
session.accessToken	Access token key received in Optional consumer authentication.	Necessity: Required Type: String Length: 32-128
consumer.device.imei	Consumer's device international mobile equipment identity.	Necessity: Optional Type: String Length: 14-16

continues on next page

Table 220 – continued from previous page

Parameter Name	Description	Value
location		Necessity: Optional Type: Object
location.lat	The latitude specified in the range of -90.0 through +90.0, both inclusive.	Necessity: Optional Type: Double
location.lng	The longitude specified in the range -180.0 through +180.0, both inclusive.	Necessity: Optional Type: Double
consumer.ipAddresses	Consumer's IP addresses.	Necessity: Optional Type: List
consumer.ipAddresses[]	Consumer's IP address.	Necessity: Optional Type: String Length: 7-15

Success Response Parameters

Parameter Name	Description	Value
endpointId	Entry point identifier for verification transaction.	Type: String
invoiceId	Verification transaction identifier.	Type: String
session		Type: Object
session.nonce	Unique random string generated by Connecting Party's server.	Type: String
session.signature	Checksum generated by hmac-sha1 (see Signature computation) from the concatenation of accessToken, consumer.device.serialNumber, endpointId, session.nonce, invoiceId and transaction.currency.	Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, SERVER_FAILED, SERVER_UNAVAILABLE.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String

Request Example

```
{
  "consumer": {
    "device": {
      "imei": "string",
      "serialNumber": "string"
    },
    "ipAddresses": ["string"]
  },
  "location": {
    "lat": "double",
    "lng": "double"
  },
  "session": {
    "accessToken": "string"
  },
  "transaction": {
    "currency": "string"
  }
}
```

Success Response Example

```
{
  "endpointId": "string",
  "invoiceId": "string",
  "session": {
    "nonce": "string",
    "signature": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  }
}
```

7.64.2 Check verification

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Check verification is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

Check verification is used for security purposes and allows Doc2.0 to compare the data sent by Connecting Party's app with the data stored on Connecting Party's server.

It is highly recommended to respond with the actual information stored on Connecting Party's server.

API URLs

Note:

This is an example of URL format on Connecting Party server.

Connecting Party Server
https://proxy.connectingpartyserver.com/verification/{invoiceId}/check

Request Parameters

Parameter Name	Description	Value
consumer		Type: Object
consumer.device		Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Type: String Length: 1-50
session		Type: Object
session.accessToken	Consumer's access token key sent by Connecting Party's app.	Type: String Length: 32-128
session.nonce	Value sent by Connecting Party's app.	Type: String Length: 32-128
session.signature	Value sent by Connecting Party's app.	Type: String Length: 32-128
session.token	The session token of verification transaction.	Type: String Length: 16-36

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Verification transaction identifier.	Necessity: Required Type: String
session		Necessity: Required Type: Object

continues on next page

Table 225 – continued from previous page

Parameter Name	Description	Value
session.checkSignature	Checksum generated by hmac-sha1 (see Signature computation) from the concatenation of consumer.device.serialNumber, session.nonce, endpointId and invoiceId.	Necessity: Required Type: String
session.nonce	Unique random string generated by Connecting Party's server in Initiate verification response.	Necessity: Required Type: String
session.token	The session token of verification transaction.	Necessity: Required Type: String
billing		Necessity: Optional Type: Object
billing.address		Necessity: Optional Type: Object
billing.address.city	City name.	Necessity: Optional Type: String
billing.address.country	ISO alpha3 country code.	Necessity: Optional Type: String
billing.address.postcodeZip	ZIP post code.	Necessity: Optional Type: String
billing.address.state	State is used if present. For example, Los Angeles is in CA state.	Necessity: Optional Type: String

continues on next page

Table 225 – continued from previous page

Parameter Name	Description	Value
billing.address.street	Street is used for district names and street names.	Necessity: Optional Type: String
billing.address.street2	Street 2 is used for long addresses. Usually it contains apartment and building №.	Necessity: Optional Type: String
billing.phone	Customer's contact phone number.	Necessity: Optional Type: String
consumer.birthDay	Consumer's birth day in "ddMMyyyy" format.	Necessity: Optional Type: String
consumer.email	Consumer's email address.	Necessity: Optional Type: String
consumer.ipAddresses	Consumer's IP addresses.	Necessity: Optional Type: List
consumer.ipAddresses[]	Consumer's IP address.	Necessity: Optional Type: String
transaction.currency	Upper case letters (ISO 4217 alpha code).	Necessity: Optional Type: String
transaction.randomSumCentis	Random sum amount (for random sum check), 1.00 EUR = 100 centis.	Necessity: Optional Type: Integer

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, SERVER_FAILED, SERVER_UNAVAILABLE.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Verification transaction identifier.	Type: String

Request Example

```
{
  "consumer": {
    "device": {
      "serialNumber": "string"
    }
  },
  "session": {
    "accessToken": "string",
    "nonce": "string",
    "signature": "string",
    "token": "string"
  }
}
```

Success Response Example

```
{
  "billing": {
    "address": {
      "city": "string",
      "country": "string",
      "postcodeZip": "string",
      "state": "string",
      "street": "string",
      "street2": "string"
    },
    "phone": "string"
  },
  "consumer": {
    "birthDay": "string",
```

(continues on next page)

(continued from previous page)

```
"email": "string",
"ipAddresses": ["string"]
},
"invoiceId": "string",
"session": {
  "checkSignature": "string",
  "nonce": "string",
  "token": "string"
},
"transaction": {
  "currency": "string",
  "randomSumCentis": "integer"
}
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

7.64.3 Verification card mapping notification

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Verification card mapping notification is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

Card mapping notification is used to allow Consumer credentials (phone, email, billing address, twitter name, etc) for mapping to card data. Card mappings can be used later in transfer or sale transactions instead of card numbers and other cardholder information.

Doc2.0 sends Verification card mapping notification request to Connecting Party's server/proxy with created on its side card reference - {serverCardId}.

Connecting Party's server maps this {serverCardId} to {clientCardId}, created on its side. This {clientCardId} can then be mapped in Connecting Party's app or on Connecting Party's server to any Consumer's data from the initial Perform validation request. This allows Connecting Party's app to process sale or transfer transactions by Consumer's email, phone number, etc.

{uniqueReferenceCardId} remains the same if Consumer makes another verification with the same card number but different expiry date, for example. It can be used by Connecting Party for loyalty programs or fraud control.

API URLs

Note:

This is an example of URL format on Connecting Party server.

Connecting Party Server
https://proxy.connectingpartyserver.com/verification/notification

Request Parameters

Parameter Name	Description	Value
consumer		Type: Object
consumer.device		Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Type: String Length: 1-50
reference		Type: Object

continues on next page

Table 228 – continued from previous page

Parameter Name	Description	Value
reference.serverCardId	The card reference identifier (on server side).	Type: String Length: 1-64
reference.uniqueReferenceCardId	The card identifier (on server side).	Type: Integer
session		Type: Object
session.accessToken	Consumer's access token key.	Type: String Length: 32-128
session.token	The session token of verification transaction.	Type: String Length: 16-36
sourceCard		Type: Object
sourceCard.bin	The Source card BIN.	Type: Integer

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Verification transaction identifier.	Type: String
session		Type: Object
session.checkSignature	Checksum generated by hmac-sha1 (see Signature computation) from the concatenation of the session.nonce, the endpointId and the invoiceId.	Type: String
session.nonce	Unique random string generated by Connecting Party's server.	Type: String
session.token	The session token of verification transaction.	Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, SERVER_FAILED, SERVER_UNAVAILABLE.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Verification transaction identifier.	Type: String

Request Example

```
{
  "consumer": {
    "device": {
      "serialNumber": "string"
    }
  },
  "reference": {
    "serverCardId": "string",
    "uniqueReferenceCardId": "integer"
  },
  "sourceCard": {
    "bin": "integer"
  },
  "session": {
    "accessToken": "string",
    "token": "string"
  }
}
```

Success Response Example

```
{
  "invoiceId": "string",
  "session": {
    "checkSignature": "string",
    "nonce": "string",
    "token": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

7.64.4 Initiate sale

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Initiate sale is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

To initiate sale, Connecting Party's app sends {accessToken} with transaction amount and other device parameters to Connecting Party's server, which are used to start a session with unique random {nonce} and encrypted {signature}.

In response Connecting Party's server also provides information about invoice in Connecting Party's system and endpoint in Doc2.0.

API URLs

Note:

This is an example of URL format on Connecting Party server.

Connecting Party Server

https://proxy.connectingpartyserver.com/sale/initiate-sale

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
transaction		Necessity: Required Type: Object
transaction.amountCentis	Transaction amount, 1.00 EUR = 100 centis.	Necessity: Required Type: Integer
transaction.currency	Upper case letters (ISO 4217 alpha code).	Necessity: Required Type: String Length: 3
session		Necessity: Required Type: Object
session.accessToken	Access token key received in Optional consumer authentication.	Necessity: Required Type: String Length: 32-128

continues on next page

Table 232 – continued from previous page

Parameter Name	Description	Value
consumer.device.imei	Consumer's device international mobile equipment identity.	Necessity: Optional Type: String Length: 14-16
location		Necessity: Optional Type: Object
location.lat	The latitude specified in the range of -90.0 through +90.0, both inclusive.	Necessity: Optional Type: Double
location.lng	The longitude specified in the range -180.0 through +180.0, both inclusive.	Necessity: Optional Type: Double
consumer.ipAddresses	Consumer's IP addresses.	Necessity: Optional Type: List
consumer.ipAddresses[]	Consumer's IP address.	Necessity: Optional Type: String Length: 7-15

Success Response Parameters

Parameter Name	Description	Value
endpointId	Entry point identifier for sale transaction.	Necessity: Required Type: String
invoiceId	Transfer transaction identifier.	Necessity: Required Type: String

continues on next page

Table 233 – continued from previous page

Parameter Name	Description	Value
session		Necessity: Required Type: Object
session.nonce	Unique random string generated by Connecting Party's server.	Necessity: Required Type: String
session.signature	Checksum generated by hmac-sha1 (see Signature computation) from the concatenation of the accessToken, consumer.device.serialNumber, endpointId, session.nonce, invoiceId, transaction.amountCentis and the transaction.currency.	Necessity: Required Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, SERVER_FAILED, SERVER_UNAVAILABLE.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String

Request Example

```
{
  "consumer": {
    "device": {
      "imei": "string",
      "serialNumber": "string"
    },
    "ipAddresses": ["string"]
  },
  "location": {
    "lat": "double",
    "lng": "double"
  },
}
```

(continues on next page)

(continued from previous page)

```
"session": {
  "accessToken": "string"
},
"transaction": {
  "amountCentis": "integer",
  "currency": "string"
}
}
```

Success Response Example

```
{
  "endpointId": "string",
  "invoiceId": "string",
  "session": {
    "nonce": "string",
    "signature": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  }
}
```

7.64.5 Check sale

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Check sale is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#). Check sale is used for security purposes and allows Doc2.0 to compare the data sent by Connecting Party's app with the data stored on Connecting Party's server. It is highly recommended to respond with the actual information stored on Connecting Party's server.

API URLs

Note:

This is an example of URL format on Connecting Party server.

Connecting Party Server
https://proxy.connectingpartyserver.com/sale/{invoiceId}/check

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
session		Necessity: Required Type: Object

continues on next page

Table 236 – continued from previous page

Parameter Name	Description	Value
session.accessToken	Consumer's access token key sent by Connecting Party's app.	Necessity: Required Type: String Length: 32-128
session.nonce	Value sent by Connecting Party's app.	Necessity: Required Type: String Length: 32-128
session.signature	Value sent by Connecting Party's app.	Necessity: Required Type: String Length: 32-128
session.token	The session token of sale transaction.	Necessity: Required Type: String Length: 16-36
sourceOfFunds		Necessity: Optional Type: Object
sourceOf-Funds.reference		Necessity: Optional Type: Object
sourceOf-Funds.reference.clientCardId	The source card reference identifier (on client side) if sent in Perform sale request.	Necessity: Optional Type: String Length: 1-128

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Sale transaction identifier	Necessity: Required Type: String
session		Necessity: Required Type: Object
session.checkSignature	Checksum generated by hmac-sha1 (see Signature computation) from the concatenation of the consumer.device.serialNumber, session.nonce, endpointId, invoiceId, sourceOfFunds.reference.serverCardId (if present).	Necessity: Required Type: String
session.nonce	Unique random string generated by Connecting Party's server in Initiate sale response.	Necessity: Required Type: String
session.token	The session token of sale transaction.	Necessity: Required Type: String
consumer.email	Consumer's email address, if sent in original card mapping transaction.	Necessity: Optional Type: String
sourceOfFunds		Necessity: Optional Type: Object
sourceOfFunds.reference		Necessity: Optional Type: Object

continues on next page

Table 237 – continued from previous page

Parameter Name	Description	Value
sourceOf-Funds.reference.serverCardId	The source card reference identifier (on Server side).	Necessity: Optional Type: String
transaction.currency	Upper case letters (ISO 4217 alpha code).	Necessity: Optional Type: String
transaction.randomSumCentis	Random sum amount (for random sum check), 1.00 EUR = 100 centis.	Necessity: Optional Type: Integer

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, SERVER_FAILED, SERVER_UNAVAILABLE.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Sale transaction identifier.	Type: String

Request Example

```
{
  "consumer": {
    "device": {
      "serialNumber": "string"
    }
  },
  "session": {
    "accessToken": "string",
    "nonce": "string",
    "signature": "string",
    "token": "string"
  }
}
```

(continues on next page)

(continued from previous page)

```
},
"sourceOfFunds": {
  "reference": {
    "clientCardId": "string"
  }
}
}
```

Success Response Example

```
{
  "consumer": {
    "email": "string"
  },
  "invoiceId": "string",
  "session": {
    "checkSignature": "string",
    "nonce": "string",
    "token": "string"
  },
  "sourceOfFunds": {
    "card": {
      "expiry": {
        "month": "integer",
        "year": "integer"
      },
      "holder": {
        "firstName": "string",
        "lastName": "string"
      },
      "number": "string"
    },
    "reference": {
      "serverCardId": "string",
    }
  },
  "transaction": {
    "currency": "string",
    "randomSumCentis": "integer"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

7.64.6 Sale card mapping notification

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Sale card mapping notification is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

Card mapping notification is used to allow Consumer credentials (phone, email, billing address, twitter name, etc) for mapping to card data. Card mappings can be used later in transfer or sale transactions instead of card numbers and other cardholder information.

Doc2.0 sends Sale card mapping notification request to Connecting Party's server/proxy with created on its side card reference - {serverCardId}.

Connecting Party's server maps this {serverCardId} to {clientCardId}, created on its side. This {clientCardId} can then be mapped in Connecting Party's app or on Connecting Party's server to any Consumer's data from the initial Perform sale request. This allows Connecting Party's app to process sale or transfer transactions by Consumer's email, phone number, etc. {uniqueReferenceCardId} remains the same if Consumer makes another transaction with the same card number but different expiry date, for example. It can be used by Connecting Party for loyalty programs or fraud control.

Connecting Party's app makes new sale requests using {clientCardId} instead of source card data. Doc2.0 sends this {clientCardId} to Connecting Party's server in "Check sale request" and gets mapped to it {serverCardId} in "Check sale response" from Connecting Party's server, which is used to continue the processing of this transaction.

{serverCardId} of source or verified card can be used for new sale transactions and both as source or destination for new transfer transactions.

API URLs

Note:

This is an example of URL format on Connecting Party server.

Connecting Party Server
https://proxy.connectingpartyserver.com/sale/notification

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
session		Necessity: Required Type: Object
session.accessToken	Consumer's access token key.	Necessity: Required Type: String Length: 32-128
session.token	The session token of sale transaction.	Necessity: Required Type: String Length: 16-36
sourceCard		Necessity: Optional Type: Object
sourceCard.bin	The Source card BIN.	Necessity: Optional Type: Integer

continues on next page

Table 240 – continued from previous page

Parameter Name	Description	Value
sourceCard.reference		Necessity: Optional Type: Object
source-Card.reference.serverCardId	The Source card reference identifier (on server side).	Necessity: Optional Type: String Length: 1-64
source-Card.reference.uniqueReferenceCardId	The Source card unique identifier (on server side).	Necessity: Optional Type: Integer

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Sale transaction identifier.	Type: String
session		Type: Object
session.checkSignature	Checksum generated by hmac-sha1 (see Signature computation) from the concatenation of session.nonce, endPointId and invoiceId.	Type: String
session.nonce	Unique random string generated by Connecting Party's server.	Type: String
session.token	The session token of sale transaction.	Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, SERVER_FAILED, SERVER_UNAVAILABLE.	Type: Enum

continues on next page

Table 242 – continued from previous page

Parameter Name	Description	Value
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Sale transaction identifier.	Type: String

Request Example

```
{
  "consumer": {
    "device": {
      "serialNumber": "string"
    }
  },
  "session": {
    "accessToken": "string",
    "token": "string"
  },
  "sourceCard": {
    "bin": "integer",
    "reference": {
      "serverCardId": "string",
      "uniqueReferenceCardId": "integer"
    }
  }
}
```

Success Response Example

```
{
  "invoiceId": "string",
  "session": {
    "checkSignature": "string",
    "nonce": "string",
    "token": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

7.64.7 Initiate transfer

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Initiate transfer is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

To initiate funds transfer, Connecting Party's app sends {accessToken} with transaction amount and other device parameters to Connecting Party's server, which are used to start a session with unique random {nonce} and encrypted {signature}.

In response Connecting Party's server also provides information about invoice in Connecting Party's system and endpoint in Doc2.0.

The Connecting Party's server may also respond with the transaction's commission.

API URLs

Note:

This is an example of URL format on Connecting Party server.

Connecting Party Server
https://proxy.connectingpartyserver.com/transfer/initiate-transfer

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
transaction		Necessity: Required Type: Object
transaction.amountCentis	Transaction amount, 1.00 EUR = 100 centis.	Necessity: Required Type: Integer
transaction.currency	Upper case letters (ISO 4217 alpha code).	Necessity: Required Type: String Length: 3
session		Necessity: Required Type: Object
session.accessToken	Access token key received in Optional consumer authentication.	Necessity: Required Type: String Length: 32-128

continues on next page

Table 244 – continued from previous page

Parameter Name	Description	Value
consumer.device.imei	Consumer's device international mobile equipment identity.	Necessity: Optional Type: String Length: 14-16
location		Necessity: Optional Type: Object
location.lat	The latitude specified in the range of -90.0 through +90.0, both inclusive.	Type: Double
location.lng	The longitude specified in the range -180.0 through +180.0, both inclusive.	Type: Double
consumer.ipAddresses	Consumer's IP addresses.	Type: List
consumer.ipAddresses[]	Consumer's IP address.	Necessity: Optional Type: String Length: 7-15
fromBin	Source of funds card BIN (first 6 digits of card number) may be used to display pre-calculated commission.	Necessity: Optional Type: String Length: 6
toBin	Destination of funds card BIN (first 6 digits of card number) may be used to display pre-calculated commission.	Necessity: Optional Type: String Length: 6

Success Response Parameters

Parameter Name	Description	Value
endpointId	Entry point identifier for transfer transaction.	Necessity: Required Type: String

continues on next page

Table 245 – continued from previous page

Parameter Name	Description	Value
invoiceId	Transfer transaction identifier.	Necessity: Required Type: String
session		Necessity: Required Type: Object
session.nonce	Unique random string generated by Connecting Party's server.	Necessity: Required Type: String
session.signature	Checksum generated by hmac-sha1 (see Signature computation) from the concatenation of the accessToken, consumer.device.serialNumber, endpointId, session.nonce, invoiceId, transaction.amountCentis and the transaction.currency.	Necessity: Required Type: String
rates		Necessity: Optional Type: Object
rates.min	Possible minimum commission amount.	Necessity: Optional Type: String
rates.max	Possible maximum commission amount.	Necessity: Optional Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, SERVER_FAILED, SERVER_UNAVAILABLE.	Type: Enum

continues on next page

Table 246 – continued from previous page

Parameter Name	Description	Value
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String

Request Example

```
{
  "consumer": {
    "device": {
      "imei": "string",
      "serialNumber": "string"
    },
    "ipAddresses": ["string"]
  },
  "location": {
    "lat": "double",
    "lng": "double"
  },
  "session": {
    "accessToken": "string"
  },
  "transaction": {
    "amountCentis": "integer",
    "currency": "string"
  }
}
```

Success Response Example

```
{
  "endpointId": "string",
  "invoiceId": "string",
  "rates": {
    "min": "string",
    "max": "string"
  },
  "session": {
    "nonce": "string",
    "signature": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  }
}
```

7.64.8 Check transfer

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Check transfer is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#). Check transfer is used for security purposes and allows Doc2.0 to compare the data sent by Connecting Party's app with the data stored on Connecting Party's server. It is highly recommended to respond with the actual information stored on Connecting Party's server.

API URLs

Note:

This is an example of URL format on Connecting Party server.

Connecting Party Server
https://proxy.connectingpartyserver.com/transfer/{invoiceId}/check

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
session		Necessity: Required Type: Object
session.accessToken	Consumer's access token key sent by Connecting Party's app.	Necessity: Required Type: String Length: 32-128
session.nonce	Value sent by Connecting Party's app.	Necessity: Required Type: String Length: 32-128
session.signature	Value sent by Connecting Party's app.	Necessity: Required Type: String Length: 32-128

continues on next page

Table 248 – continued from previous page

Parameter Name	Description	Value
session.token	The session token of transfer transaction.	Necessity: Required Type: String Length: 16-36
sourceOfFunds		Necessity: Optional Type: Object
sourceOfFunds.reference		Necessity: Optional Type: Object
sourceOfFunds.reference.clientCardId	The source card reference identifier (on client side) if sent in Perform Transfer request.	Necessity: Optional Type: String Length: 1-128
destinationOfFunds		Necessity: Optional Type: Object
destinationOfFunds.reference		Necessity: Optional Type: Object
destinationOfFunds.reference.clientCardId	The destination card reference identifier (on client side) if sent in Perform Transfer request.	Necessity: Optional Type: String Length: 1-128

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Transfer transaction identifier.	Necessity: Required Type: String

continues on next page

Table 249 – continued from previous page

Parameter Name	Description	Value
session		Necessity: Required Type: Object
session.checkSignature	Checksum generated by hmac-sha1 (see Signature computation) from the concatenation of the consumer.device.serialNumber, session.nonce, destinationOfFunds.reference.serverCardId (if present), endpointId, invoiceId, sourceOfFunds.reference.serverCardId (if present).	Necessity: Required Type: String
session.nonce	Unique random string generated by Connecting Party's server in Initiate transfer response.	Necessity: Required Type: String
session.token	The session token of transfer transaction.	Necessity: Required Type: String
consumer.email	Consumer's email address.	Necessity: Optional Type: String
destinationOfFunds		Necessity: Optional Type: Object
destinationOfFunds.reference		Necessity: Optional Type: Object
destinationOfFunds.reference.serverCardId	The destination card reference identifier (on server side).	Necessity: Optional Type: String
sourceOfFunds		Necessity: Optional Type: Object

continues on next page

Table 249 – continued from previous page

Parameter Name	Description	Value
sourceOf-Funds.reference		Necessity: Optional Type: Object
sourceOf-Funds.reference.serverCardId	The source card reference identifier (on server side).	Necessity: Optional Type: String
transaction.currency	Upper case letters (ISO 4217 alpha code).	Necessity: Optional Type: String
transaction.randomSumCentis	Random sum amount (for random sum check), 1.00 EUR = 100 centis.	Necessity: Optional Type: Integer

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, SERVER_FAILED, SERVER_UNAVAILABLE.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Transfer transaction identifier.	Type: String

Request Example

```
{
  "consumer": {
    "device": {
      "serialNumber": "string"
    }
  },
  "destinationOfFunds": {
    "reference": {
      "clientCardId": "string"
    }
  },
  "session": {
    "accessToken": "string",
    "nonce": "string",
    "signature": "string",
    "token": "string"
  },
  "sourceOfFunds": {
    "reference": {
      "clientCardId": "string"
    }
  }
}
```

Success Response Example

```
{
  "consumer": {
    "email": "string"
  },
  "destinationOfFunds": {
    "card": {
      "number": "string"
    },
    "reference": {
      "serverCardId": "string"
    }
  },
  "invoiceId": "string",
  "session": {
    "checkSignature": "string",
    "nonce": "string",
    "token": "string"
  },
  "sourceOfFunds": {
    "card": {
      "expiry": {
        "month": "integer",
        "year": "integer"
      },
      "holder": {
        "firstName": "string",
        "lastName": "string"
      }
    }
  }
}
```

(continues on next page)

(continued from previous page)

```
    },
    "number": "string"
  },
  "reference": {
    "serverCardId": "string",
  }
},
"transaction": {
  "currency": "string",
  "randomSumCentis": "integer"
}
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

7.64.9 Transfer card mapping notification

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Transfer card mapping notification is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

Card mapping notification is used to allow Consumer credentials (phone, email, billing address, twitter name, etc) for mapping to card data. Card mappings can be used later in transfer or sale transactions instead of card numbers and other cardholder information. Doc2.0 sends Transfer card mapping notification request to Connecting Party's server/proxy with created on its side card reference - {serverCardId}.

Connecting Party's server maps this {serverCardId} to {clientCardId}, created on its side. This {clientCardId} can then be mapped in Connecting Party's app or on Connecting Party's server to any Consumer's data from the initial Perform transfer request. This allows Connecting Party's app to process sale or transfer transactions by Consumer's email, phone number, etc.

{uniqueReferenceCardId} remains the same if Consumer makes another verification with the same card number but different expiry date, for example. It can be used by Connecting Party for loyalty programs or fraud control.

Connecting Party's app makes new transfer requests using {clientCardId} instead of source or destination card data. Doc2.0 sends this {clientCardId} to Connecting Party's server in "Check transfer request" and gets mapped to it {serverCardId} in "Check transfer response" from Connecting Party's server, which is used to continue processing the transfer transaction.

{serverCardId} of source or verified card can be used both as source or destination for new transfer and sale transactions; {serverCardId} of destination card can be used only as destination for new transfers.

API URLs

Note:

This is an example of URL format on Connecting Party server.

Connecting Party Server
https://proxy.connectingpartyserver.com/transfer/notification

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object

continues on next page

Table 252 – continued from previous page

Parameter Name	Description	Value
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
session		Necessity: Required Type: Object
session.accessToken	Consumer's access token key.	Necessity: Required Type: String Length: 32-128
session.token	The session token of transfer transaction.	Necessity: Required Type: String Length: 16-36
destinationCard		Necessity: Optional Type: Object
destinationCard.bin	The Destination card BIN.	Necessity: Optional Type: Integer
destination-Card.reference		Necessity: Optional Type: Object
destination-Card.reference.serverCardId	The Destination card reference identifier (on server side).	Necessity: Optional Type: String Length: 1-64

continues on next page

Table 252 – continued from previous page

Parameter Name	Description	Value
destination-Card.reference.uniqueReferenceId	The Destination card unique identifier (on server side).	Necessity: Optional Type: Integer
sourceCard		Necessity: Optional Type: Object
sourceCard.bin	The Source card BIN.	Necessity: Optional Type: Integer
sourceCard.reference		Necessity: Optional Type: Object
source-Card.reference.serverCardId	The Source card reference identifier (on server side).	Necessity: Optional Type: String Length: 1-64
source-Card.reference.uniqueReferenceId	The Source card unique identifier (on server side).	Necessity: Optional Type: Integer

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Transfer transaction identifier.	Type: String
session		Type: Object
session.checkSignature	Checksum generated by hmac-sha1 (see Signature computation) from the concatenation of session.nonce, endpointId and invoiceId.	Type: String

continues on next page

Table 253 – continued from previous page

Parameter Name	Description	Value
session.nonce	Unique random string generated by Connecting Party's server.	Type: String
session.token	The session token of transfer transaction.	Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, SERVER_FAILED, SERVER_UNAVAILABLE.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Transfer transaction identifier.	Type: String

Request Example

```
{
  "consumer": {
    "device": {
      "serialNumber": "string"
    }
  },
  "destinationCard": {
    "bin": "integer",
    "reference": {
      "serverCardId": "string",
      "uniqueReferenceCardId": "integer"
    }
  },
  "session": {
    "accessToken": "string",
    "token": "string"
  },
  "sourceCard": {
    "bin": "integer",
    "reference": {
      "serverCardId": "string",

```

(continues on next page)

(continued from previous page)

```
    "uniqueReferenceCardId": "integer"
  }
}
```

Success Response Example

```
{
  "invoiceId": "string",
  "session": {
    "checkSignature": "string",
    "nonce": "string",
    "token": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

7.64.10 Development tools

- [Introduction](#)
- [Signature computation](#)
- [Example code and SDK](#)

Introduction

Signature computation

The signature is being used to ensure that it is Connecting Party (and not a fraudster) who sends the information to Doc2.0.

Signatures in DAPI are computed using HMAC_SHA1 algorithm. Key is derived in the following way:

1. Take Connecting Party ControlKey

2. Remove dashes from it
3. Unhex the result to binary

The correct signature values for all types of transactions supported by DAPI in the debugging tool below:

Example code and SDK

There are several solutions available on GitHub, which were developed for integration with Doc2.0 on Android and iOS platforms via DAPI.

Connecting Party proxy server for Money transfer:

<https://github.com/payneteasypaynet-transfer-merchant-proxy>

Android SDK of Java implementation for Money transfer:

<https://github.com/payneteasypaynet-transfer-api>

iOS SDK of Objective-C implementation for Money transfer:

<https://github.com/payneteasypayneteasypaynet-transfer>

7.65 Mobile Device Doc2.0 API

7.65.1 Perform validation

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

Introduction

Perform validation is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

Connecting Party's app sends cardholder, device, session data and other parameters straight to Doc2.0 to check the card for being valid.

A lot of criteria checks and filters may be applied, such as BIN, currency, country, blacklisted or not, etc. Depending on the integration, these settings may be customizable.

API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/dapi/v1/verification/{endpoint}	https://api.doc2.com/paynet/dapi/v1/verification/{endpoint}

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
consumer.email	Consumer's email address.	Necessity: Required Type: String Length: 3-128

continues on next page

Table 256 – continued from previous page

Parameter Name	Description	Value
session		Necessity: Required Type: Object
session.accessToken	Access token key received in Optional consumer authentication.	Necessity: Required Type: String Length: 32-128
session.nonce	Value received in Initial verification response.	Necessity: Required Type: String Length: 32-128
session.signature	Value received in Initial verification response.	Necessity: Required Type: String Length: 32-128
sourceCard		Necessity: Required Type: Object
sourceCard.expiry		Necessity: Required Type: Object
sourceCard.expiry.month	Source Card expiration month.	Necessity: Required Type: Integer Length: 2
sourceCard.expiry.year	Source Card expiration year.	Necessity: Required Type: Integer Length: 4

continues on next page

Table 256 – continued from previous page

Parameter Name	Description	Value
sourceCard.holder		Necessity: Required Type: Object
source-Card.holder.firstName	Cardholder's first name.	Necessity: Required Type: String Length: 1-50
source-Card.holder.lastName	Cardholder's last name.	Necessity: Required Type: String Length: 1-50
source-Card.holder.printedName	Cardholder's printed name (first and last name together). May be used instead of firstName + lastName.	Necessity: Required Type: String Length: 1-50
sourceCard.number	Source card number (PAN).	Necessity: Required Type: String Length: 9-19
source-Card.securityCode	Source card security code (CVV).	Necessity: Required Type: String Length: 3-4
transaction		Necessity: Required Type: Object
transaction.currency	Upper case letters (ISO 4217 alpha code).	Necessity: Required Type: String Length: 3

continues on next page

Table 256 – continued from previous page

Parameter Name	Description	Value
location		Necessity: Optional Type: Object
location.lat	The latitude specified in the range of -90.0 through +90.0, both inclusive.	Necessity: Optional Type: Double
location.lng	The longitude specified in the range -180.0 through +180.0, both inclusive.	Necessity: Optional Type: Double
billing		Necessity: Optional Type: Object
billing.address		Necessity: Optional Type: Object
billing.address.city	City name.	Necessity: Optional Type: String Length: 1-100
billing.address.country	ISO alpha3 country code.	Necessity: Optional Type: String Length: 3
billing.address.postcodeZip	ZIP post code	Necessity: Optional Type: String Length: 1-10
billing.address.state	State is used if present. For example, Los Angeles is in CA state.	Necessity: Optional Type: String Length: 2-3

continues on next page

Table 256 – continued from previous page

Parameter Name	Description	Value
billing.address.street	Street is used for district names and street names.	Necessity: Optional Type: String Length: 1-50
billing.address.street2	Street 2 is used for long addresses. Usually it contains apartment and building №.	Necessity: Optional Type: String Length: 1-50
billing.phone	Customer's contact phone number.	Necessity: Optional Type: String Length: 1-20
consumer.birthDay	Consumer's birth day in "ddMMyyyy" format.	Necessity: Optional Type: String Length: 8
consumer.device.imei	Consumer's device international mobile equipment identity.	Necessity: Optional Type: String Length: 14-16
consumer.ipAddresses	Consumer's IP addresses.	Necessity: Optional Type: List
consumer.ipAddresses[]	Consumer's IP address.	Necessity: Optional Type: String Length: 7-15
order		Necessity: Optional Type: Object

continues on next page

Table 256 – continued from previous page

Parameter Name	Description	Value
order.description	Brief transaction description. May be mandatory for certain Acquirer Banks.	Necessity: Optional Type: String Length: 128
order.siteUrl	URL of website associated with the transaction.	Necessity: Optional Type: String Length: 128
order.purpose	May be used for better transaction control or additional checks.	Necessity: Optional Type: String Length: 128

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Verification transaction identifier.	Type: String
session		Type: Object
session.token	The session token of verification transaction.	Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: INVALID_REQUEST, INVALID_CONFIGURATION, INTERNAL_ERROR, PROCESSING_FAILED.	Type: Enum
error.code	The unique error code.	Type: String

continues on next page

Table 258 – continued from previous page

Parameter Name	Description	Value
error.message	Description of the error.	Type: String
invoiceId	Verification transaction identifier.	Type: String

Request Example

```
{
  "billing": {
    "address": {
      "city": "string",
      "country": "string",
      "postcodeZip": "string",
      "state": "string",
      "street": "string",
      "street2": "string"
    },
    "phone": "string"
  },
  "consumer": {
    "birthDay": "string",
    "device": {
      "imei": "string",
      "serialNumber": "string"
    },
    "email": "string",
    "ipAddresses": ["string"]
  },
  "location": {
    "lat": "double",
    "lng": "double"
  },
  "order": {
    "description": "string",
    "siteUrl": "string",
    "purpose": "string"
  },
  "session": {
    "accessToken": "string",
    "nonce": "string",
    "signature": "string"
  },
  "sourceCard": {
    "expiry": {
      "month": "integer",
      "year": "integer"
    },
    "holder": {
      "firstName": "string",
      "lastName": "string"
    }
  },
}
```

(continues on next page)

(continued from previous page)

```
"number": "string",
"securityCode": "string"
},
"transaction": {
  "currency": "string"
}
}
```

Success Response Example

```
{
  "invoiceId": "string",
  "session": {
    "token": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

Request Builder

For integration purposes use [Doc2.0 test scenarios](#) to generate random card numbers. Parameters with light red background are mandatory.

7.65.2 Verification status

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)

- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

Introduction

Verification status is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

Verification status request is made by Connecting Party's app to Doc2.0 to get the status of verification transaction.

It is recommended to send Verification status request for a period of 2-3 minutes with interval of 3-5 seconds between requests.

Connecting Party's server should rely on the callback for final status.

API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/dapi/v1/verification/{endpoint}	https://api.doc2.com/paynet/dapi/v1/verification/{endpoint}

Request Parameters

Parameter Name	Description	Value
session		Necessity: Required Type: Object
session.accessToken	The access token key received in Optional consumer authentication.	Necessity: Required Type: String Length: 32-128

continues on next page

Table 260 – continued from previous page

Parameter Name	Description	Value
session.token	The session token of verification transaction.	Necessity: Required Type: String Length: 16-36

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Verification transaction identifier.	Type: String
session		Type: Object
session.token	The session token of verification transaction.	Type: String
state	Possible values: PROCESSING, REDIRECT_REQUEST, APPROVED, DECLINED.	Type: Enum
bankOrderId	Required if state is APPROVED or DECLINED.	Necessity: Conditional Type: String Length: 128
redirectUrl	Required if state is REDIRECT_REQUEST.	Necessity: Conditional Type: String Length: 128
errorCode	Decline reason code, may present if state is DECLINED.	Necessity: Optional Type: Integer Length: 128
errorMessage	Decline reason message, may present if state is DECLINED.	Necessity: Optional Type: String Length: 128

continues on next page

Table 261 – continued from previous page

Parameter Name	Description	Value
secure3DAuthStatus	Possible values: AUTHENTICATED, NOT_AUTHENTICATED, UNSUPPORTED, UNSUPPORTED means that card does not support 3-D Secure or 3DS authentication process finished with an unknown status.	Necessity: Optional Type: Enum
randomSumAuthStatus	Possible values: AUTHENTICATED, NOT_AUTHENTICATED.	Necessity: Optional Type: Enum

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, INTERNAL_ERROR, PROCESSING_FAILED, ORDER_NOT_FOUND.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Verification transaction identifier.	Type: String

Request Example

```
{
  "session": {
    "accessToken": "string",
    "token": "string"
  }
}
```

Success Response Example

```
{
  "bankOrderId": "string",
  "invoiceId": "string",
  "redirectUrl": "string",
  "session": {
    "token": "string"
  },
  "state": "enum"
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

Request Builder

7.65.3 Perform sale

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

Introduction

Perform sale is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#). On this stage Connecting Party's app sends cardholder, device, session data and other parameters straight to Doc2.0 to perform sale transaction.

A lot of criteria checks and filters may be applied, such as BIN, currency, country, blacklisted or not, etc. Depending on the integration, these settings may be customizable.

API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/dapi/v1/sale/{endpointId}/invoice/pay	https://gateway.doc2.com/paynet/dapi/v1/sale/{endpointId}/{invoiceId}/invoice/pay

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
consumer.email	Consumer's email address, if sent in original card mapping transaction and card reference is being used instead of cardholder data.	Necessity: Required Type: String Length: 3-128

continues on next page

Table 264 – continued from previous page

Parameter Name	Description	Value
session		Necessity: Required Type: Object
session.accessToken	Access token key received in Optional consumer authentication.	Necessity: Required Type: String Length: 32-128
session.nonce	Value received in initiate sale response.	Necessity: Required Type: String Length: 32-128
session.signature	Value received in initiate sale response.	Necessity: Required Type: String Length: 32-128
sourceOfFunds		Necessity: Required Type: Object
sourceOfFunds.card	Only one of these subtypes (card, reference) should be used.	Necessity: Required Type: Object
sourceOfFunds.card.expiry		Necessity: Required Type: Object
sourceOfFunds.card.expiry.month	Source Card expiration month.	Necessity: Required Type: Integer Length: 2

continues on next page

Table 264 – continued from previous page

Parameter Name	Description	Value
sourceOf-Funds.card.expiry.year	Source Card expiration year.	Necessity: Required Type: Integer Length: 4
sourceOf-Funds.card.holder		Necessity: Required Type: Object
sourceOf-Funds.card.holder.firstName	Cardholder's first name.	Necessity: Required Type: String Length: 1-50
sourceOf-Funds.card.holder.lastName	Cardholder's last name.	Necessity: Required Type: String Length: 1-50
sourceOf-Funds.card.holder.printedName	Cardholder's printed name (first and last name together). May be used instead of firstName + lastName.	Necessity: Required Type: String Length: 1-50
sourceOf-Funds.card.number	Source card number (PAN).	Necessity: Required Type: String Length: 9-19
sourceOf-Funds.card.securityCode	Source card security code (CVV).	Necessity: Required Type: String Length: 3-4
sourceOf-Funds.reference	only one of these subtypes (card, reference) should be used.	Necessity: Required Type: Object

continues on next page

Table 264 – continued from previous page

Parameter Name	Description	Value
sourceOf-Funds.reference.clientCardId	Source card reference identifier (on client side).	Necessity: Required Type: String Length: 1-128
sourceOf-Funds.reference.securityCode	Source card security code (CVV).	Necessity: Required Type: String Length: 3-4
transaction		Necessity: Required Type: Object
transaction.amountCentis	Transaction amount, 1.00 EUR = 100 centis.	Necessity: Required Type: Integer
transaction.currency	Upper case letters (ISO 4217 alpha code).	Necessity: Required Type: String Length: 3
location		Necessity: Optional Type: Object
location.lat	The latitude specified in the range of -90.0 through +90.0, both inclusive.	Necessity: Optional Type: Double
location.lng	The longitude specified in the range -180.0 through +180.0, both inclusive.	Necessity: Optional Type: Double
billing		Necessity: Optional Type: Object

continues on next page

Table 264 – continued from previous page

Parameter Name	Description	Value
billing.address		Necessity: Optional Type: Object
billing.address.city	City name.	Necessity: Optional Type: String Length: 1-100
billing.address.country	ISO alpha3 country code.	Necessity: Optional Type: String Length: 3
billing.address.postcodeZip	ZIP post code.	Necessity: Optional Type: String Length: 1-10
billing.address.state	State is used if present. For example, Los Angeles is in CA state.	Necessity: Optional Type: String Length: 2-3
billing.address.street	Street is used for district names and street names.	Necessity: Optional Type: String Length: 1-50
billing.address.street2	Street 2 is used for long addresses. Usually it contains apartment and building №.	Necessity: Optional Type: String Length: 1-50
billing.phone	Consumer's contact phone number.	Necessity: Optional Type: String Length: 1-20

continues on next page

Table 264 – continued from previous page

Parameter Name	Description	Value
consumer.birthDay	Consumer's birth day in "ddMMyyyy" format.	Necessity: Optional Type: String Length: 8
consumer.device.imei	Consumer's device international mobile equipment identity.	Necessity: Optional Type: String Length: 14-16
consumer.ipAddresses	Consumer's IP addresses.	Necessity: Optional Type: List
consumer.ipAddresses[]	Consumer's IP address.	Necessity: Optional Type: String Length: 7-15
order		Necessity: Optional Type: Object
order.description	Brief transaction description. May be mandatory for certain Acquirer Banks.	Necessity: Optional Type: String Length: 128
order.siteUrl	URL of website associated with the transaction.	Necessity: Optional Type: String Length: 128
order.purpose	May be used for better transaction control or additional checks.	Necessity: Optional Type: String Length: 128

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Sale transaction identifier.	Type: String
session		Type: Object
session.token	The session token of sale transaction.	Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: INVALID_REQUEST, INVALID_CONFIGURATION, INTERNAL_ERROR, PROCESSING_FAILED.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Sale transaction identifier.	Type: String

Request Example

```
{
  "billing": {
    "address": {
      "city": "string",
      "country": "string",
      "postcodeZip": "string",
      "state": "string",
      "street": "string",
      "street2": "string"
    },
    "phone": "string"
  },
  "consumer": {
    "birthDay": "string",
```

(continues on next page)

(continued from previous page)

```
"device": {
  "imei": "string",
  "serialNumber": "string"
},
"email": "string",
"ipAddresses": ["string"]
},
"location": {
  "lat": "double",
  "lng": "double"
},
"order": {
  "description": "string",
  "siteUrl": "string",
  "purpose": "string"
},
"session": {
  "accessToken": "string",
  "nonce": "string",
  "signature": "string"
},
"sourceOfFunds": {
  "card": {
    "expiry": {
      "month": "integer",
      "year": "integer"
    },
    "holder": {
      "firstName": "string",
      "lastName": "string"
    },
    "number": "string",
    "securityCode": "string"
  },
  "reference": {
    "clientCardId": "string",
    "securityCode": "string"
  }
},
"transaction": {
  "amountCentis": "integer",
  "currency": "string"
}
}
```

Success Response Example

```
{
  "invoiceId": "string",
  "session": {
    "token": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

Request Builder

For integration purposes use [Doc2.0 test scenarios](#) to generate random card numbers.

7.65.4 Sale status

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

Introduction

Sale status is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#). Sale status request is made by Connecting Party's app to Doc2.0 to get the status of sale transaction.

It is recommended to send Sale status request for a period of 2-3 minutes with interval of 3-5 seconds between requests. Connecting Party's server should rely on the callback for final status.

API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/dapi/v1/sale/{endpointId}/invoice/paynet/dapi/v1/sale/{endpointId}/{invoiceId}	https://gateway.doc2.com/paynet/dapi/v1/sale/{endpointId}/{invoiceId}

Request Parameters

Parameter Name	Description	Value
session		Type: Object
session.accessToken	The access token key received in Optional consumer authentication.	Type: String Length: 32-128
session.token	The session token of sale transaction.	Type: String Length: 16-36

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Sale transaction identifier.	Type: String
session		Type: Object
session.token	The session token of sale transaction	Type: String
state	Possible values: PROCESSING, REDIRECT_REQUEST, APPROVED, DECLINED.	Type: Enum
bankOrderId	Required if state is APPROVED or DECLINED.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 269 – continued from previous page

Parameter Name	Description	Value
redirectUrl	Required if state is REDIRECT_REQUEST.	Necessity: Conditional Type: String Length: 128
orderId	Order identifier in payment processing system.	Necessity: Conditional Type: String
transaction		Necessity: Conditional Type: Object
transaction.amountCentis	Amount of money transfered in centis, 1.00 EUR = 100 centis.	Necessity: Conditional Type: Integer
transaction.commissionCentis	Amount of commission held in centis (depends on the integration).	Necessity: Conditional Type: Integer
transaction.orderCreatedDate	Order creation date in payment processing system.	Necessity: Conditional Type: String
transaction.transactionCreatedDate	Transaction creation date in payment processing system.	Necessity: Conditional Type: String
errorCode	Decline reason code, may present if state is DECLINED.	Necessity: Optional Type: Integer
errorMessage	Decline reason message, may present if state is DECLINED.	Necessity: Optional Type: String Length: 128

continues on next page

Table 269 – continued from previous page

Parameter Name	Description	Value
secure3DAuthStatus	Possible values: AUTHENTICATED, NOT_AUTHENTICATED, UNSUPPORTED, UNSUPPORTED means that card does not support 3-D Secure or 3DS authentication process finished with an unknown status.	Necessity: Optional Type: Enum
randomSumAuthStatus	Possible values: AUTHENTICATED, NOT_AUTHENTICATED.	Necessity: Optional Type: Enum

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, INTERNAL_ERROR, PROCESSING_FAILED, ORDER_NOT_FOUND.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Sale transaction identifier.	Type: String

Request Example

```
{
  "session": {
    "accessToken": "string",
    "token": "string"
  }
}
```

Success Response Example

```
{
  "bankOrderId": "string",
  "invoiceId": "string",
  "redirectUrl": "string",
  "session": {
    "token": "string"
  },
  "state": "enum"
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

Request Builder

7.65.5 Perform transfer

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)
- [Request Builder](#)

Introduction

Perform transfer is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

On this stage Connecting Party's app sends cardholder, device, session data and other parameters straight to Doc2.0 to perform funds transfer from card to card.

A lot of criteria checks and filters may be applied, such as BIN, currency, country, blacklisted or not, etc. Depending on the integration, these settings may be customizable.

API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/dapi/v1/transfer/{endpointId}	https://api.doc2.com/paynet/dapi/v1/transfer/{endpointId}

Request Parameters

Parameter Name	Description	Value
consumer		Necessity: Required Type: Object
consumer.device		Necessity: Required Type: Object
consumer.device.serialNumber	Consumer's device serial number.	Necessity: Required Type: String Length: 1-50
consumer.email	Consumer's email address, if sent in original card mapping transaction and card reference is being used instead of cardholder data.	Necessity: Required Type: String Length: 3-128

continues on next page

Table 272 – continued from previous page

Parameter Name	Description	Value
destinationOfFunds		Necessity: Required Type: Object
destinationOfFunds.card	Only one of these subtypes (card, reference) should be used.	Necessity: Required Type: Object
destinationOfFunds.card.number	Destination card number (PAN).	Necessity: Required Type: String Length: 9-19
destinationOfFunds.reference	Only one of these subtypes (card, reference) should be used.	Necessity: Required Type: Object
destinationOfFunds.reference.clientCardId	Destination card reference identifier (on client side).	Necessity: Required Type: String Length: 1-128
session		Necessity: Required Type: Object
session.accessToken	Access token key received in Optional consumer authentication.	Necessity: Required Type: String Length: 32-128
session.nonce	Value received in Initiate transfer response.	Necessity: Required Type: String Length: 32-128

continues on next page

Table 272 – continued from previous page

Parameter Name	Description	Value
session.signature	Value received in Initiate transfer response.	Necessity: Required Type: String Length: 32-128
sourceOfFunds		Necessity: Required Type: Object
sourceOfFunds.card	Only one of these subtypes (card, reference) should be used.	Necessity: Required Type: Object
sourceOf-Funds.card.expiry		Necessity: Required Type: Object
sourceOf-Funds.card.expiry.month	Source Card expiration month.	Necessity: Required Type: Integer Length: 2
sourceOf-Funds.card.expiry.year	Source Card expiration year.	Necessity: Required Type: Integer Length: 4
sourceOf-Funds.card.holder		Necessity: Required Type: Object
sourceOf-Funds.card.holder.firstName	Cardholder's first name.	Necessity: Required Type: String Length: 1-50

continues on next page

Table 272 – continued from previous page

Parameter Name	Description	Value
sourceOf-Funds.card.holder.lastName	Cardholder's last name.	Necessity: Required Type: String Length: 1-50
sourceOf-Funds.card.holder.printedName	Cardholder's printed name (first and last name together). May be used instead of firstName + lastName.	Necessity: Required Type: String Length: 1-50
sourceOf-Funds.card.number	Source card number (PAN).	Necessity: Required Type: String Length: 9-19
sourceOf-Funds.card.securityCode	Source card security code (CVV).	Necessity: Required Type: String Length: 3-4
sourceOf-Funds.reference	only one of these subtypes (card, reference) should be used.	Necessity: Required Type: Object
sourceOf-Funds.reference.clientCardId	Source card reference identifier (on client side).	Necessity: Required Type: String Length: 1-128
sourceOf-Funds.reference.securityCode	Source card security code (CVV).	Necessity: Required Type: String Length: 3-4
transaction		Necessity: Required Type: Object

continues on next page

Table 272 – continued from previous page

Parameter Name	Description	Value
transaction.amountCentis	Transaction amount, 1.00 EUR = 100 centis.	Necessity: Required Type: Integer
transaction.currency	Upper case letters (ISO 4217 alpha code).	Necessity: Required Type: String Length: 3
location		Necessity: Optional Type: Object
location.lat	The latitude specified in the range of -90.0 through +90.0, both inclusive.	Necessity: Optional Type: Double
location.lng	The longitude specified in the range -180.0 through +180.0, both inclusive.	Necessity: Optional Type: Double
billing		Necessity: Optional Type: Object
billing.address		Necessity: Optional Type: Object
billing.address.city	City name.	Necessity: Optional Type: String Length: 1-100
billing.address.country	ISO alpha3 country code.	Necessity: Optional Type: String Length: 3

continues on next page

Table 272 – continued from previous page

Parameter Name	Description	Value
billing.address.postcodeZip	ZIP post code.	Necessity: Optional Type: String Length: 1-10
billing.address.state	State is used if present. For example, Los Angeles is in CA state.	Necessity: Optional Type: String Length: 2-3
billing.address.street	Street is used for district names and street names.	Necessity: Optional Type: String Length: 1-50
billing.address.street2	Street 2 is used for long addresses. Usually it contains apartment and building №.	Necessity: Optional Type: String Length: 1-50
billing.phone	Consumer's contact phone number.	Necessity: Optional Type: String Length: 1-20
consumer.birthDay	Consumer's birth day in "ddMMyyyy" format.	Necessity: Optional Type: String Length: 8
consumer.device.imei	Consumer's device international mobile equipment identity.	Necessity: Optional Type: String Length: 14-16
consumer.ipAddresses	Consumer's IP addresses.	Necessity: Optional Type: List

continues on next page

Table 272 – continued from previous page

Parameter Name	Description	Value
consumer.ipAddresses[]	Consumer's IP address.	Necessity: Optional Type: String Length: 7-15
order		Necessity: Optional Type: Object
order.description	Brief transaction description. May be mandatory for certain Acquirer Banks.	Necessity: Optional Type: String Length: 128
order.siteUrl	URL of website associated with the transaction.	Necessity: Optional Type: String Length: 128
order.purpose	May be used for better transaction control or additional checks.	Necessity: Optional Type: String Length: 128

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Transfer transaction identifier.	Type: String
session		Type: Object
session.token	The session token of transfer transaction.	Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: INVALID_REQUEST, INVALID_CONFIGURATION, INTERNAL_ERROR, PROCESSING_FAILED.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Transfer transaction identifier.	Type: String

Request Example

```
{
  "billing": {
    "address": {
      "city": "string",
      "country": "string",
      "postcodeZip": "string",
      "state": "string",
      "street": "string",
      "street2": "string"
    },
    "phone": "string"
  },
  "consumer": {
    "birthDay": "string",
    "device": {
      "imei": "string",
      "serialNumber": "string"
    },
    "email": "string",
    "ipAddresses": ["string"]
  },
  "destinationOfFunds": {
    "card": {
      "number": "string",
    },
    "reference": {
      "clientCardId": "string"
    }
  },
  "location": {
    "lat": "double",
```

(continues on next page)

(continued from previous page)

```
    "lng": "double"
  },
  "order": {
    "description": "string",
    "siteUrl": "string",
    "purpose": "string"
  },
  "session": {
    "accessToken": "string",
    "nonce": "string",
    "signature": "string"
  },
  "sourceOfFunds": {
    "card": {
      "expiry": {
        "month": "integer",
        "year": "integer"
      },
      "holder": {
        "firstName": "string",
        "lastName": "string"
      },
      "number": "string",
      "securityCode": "string"
    },
    "reference": {
      "clientCardId": "string",
      "securityCode": "string"
    }
  },
  "transaction": {
    "amountCentis": "integer",
    "currency": "string"
  }
}
```

Success Response Example

```
{
  "invoiceId": "string",
  "session": {
    "token": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

Request Builder

For integration purposes use [Doc2.0 test scenarios](#) to generate random card numbers.

7.65.6 Transfer status

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Success Response Parameters for Consumer defined transfer rates](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Success Response Example for Consumer defined transfer rates](#)
- [Fail Response Example](#)
- [Request Builder](#)

Introduction

Transfer status is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#). Funds transfer status request is made by Connecting Party's app to Doc2.0 to get the status of transfer transaction.

It is recommended to send Transfer status request for a period of 2-3 minutes with interval of 3-5 seconds between requests. Connecting Party's server should rely on the callback for final status.

API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/dapi/v1/transfer/{endpointId}/transfer/{endpointId}	https://api.doc2.com/paynet/dapi/v1/transfer/{endpointId}/transfer/{endpointId}

Request Parameters

Parameter Name	Description	Value
session		Type: Object
session.accessToken	The access token key received in Optional consumer authentication.	Type: String Length: 32-128
session.token	The session token of transfer transaction.	Type: String Length: 16-36

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Transfer transaction identifier.	Type: String
session		Type: Object
session.token	The session token of transfer transaction.	Type: String
state	Possible values: PROCESSING, REDIRECT_REQUEST, APPROVED, DECLINED. If Consumer defined transfer rates enable get TRANSFER_FEE_REQUEST.	Type: Enum
bankOrderId	Required if state is APPROVED or DECLINED.	Necessity: Conditional Type: String Length: 128

continues on next page

Table 277 – continued from previous page

Parameter Name	Description	Value
redirectUrl	Required if state is REDIRECT_REQUEST.	Necessity: Conditional Type: String Length: 128
orderId	Order identifier in payment processing system.	Necessity: Conditional Type: String
transaction		Necessity: Conditional Type: Object
transaction.amountCentis	Amount of money transfered in centis, 1.00 EUR = 100 centis.	Necessity: Conditional Type: Integer
transaction.commissionCentis	Amount of commission held in centis.	Necessity: Conditional Type: Integer
transaction.orderCreatedDate	Order creation date in payment processing system.	Necessity: Conditional Type: String
transaction.transactionCreatedDate	Transaction creation date in payment processing system.	Necessity: Conditional Type: String
errorCode	Decline reason code, may present if state is DECLINED.	Necessity: Optional Type: Integer
errorMessage	Decline reason message, may present if state is DECLINED.	Necessity: Optional Type: String Length: 128

continues on next page

Table 277 – continued from previous page

Parameter Name	Description	Value
secure3DAuthStatus	Possible values: AUTHENTICATED, NOT_AUTHENTICATED, UNSUPPORTED means that card does not support 3-D Secure or 3DS authentication process finished with an unknown status.	Necessity: Optional Type: Enum
randomSumAuthStatus	Possible values: AUTHENTICATED, NOT_AUTHENTICATED.	Necessity: Optional Type: Enum

Success Response Parameters for Consumer defined transfer rates

Transfer status response includes the state parameter with TRANSFER_FEE_REQUEST value and the transferFeeList object with a list of commissions for possible processing gates. Each processing gate is represented by it's own aliased name in assignedId parameter. TRANSFER_FEE_REQUEST value in state parameter exists only in "Consumer defined transfer rates" flow.

When the transaction reaches TRANSFER_FEE_REQUEST state, the processing will remain paused until Complete transfer request is sent.

Parameter Name	Description	Value
transferFeeList		Type: Object
transferFeeList.assignedId	Aliased payment gateway name.	Type: String
transferFeeList.commissionCentis	Amount of commission to be held in centis, 1.00 EUR = 100 centis.	Type: Integer
transferFeeList.currency	Upper case letters (ISO 4217 alpha code).	Type: String
state	In addition to the values from Status Transfer , TRANSFER_FEE_REQUEST value only exists in this flow.	Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: FILTERED, INVALID_REQUEST, INTERNAL_ERROR, PROCESSING_FAILED, ORDER_NOT_FOUND.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Transfer transaction identifier.	Type: String

Request Example

```
{
  "session": {
    "accessToken": "string",
    "token": "string"
  }
}
```

Success Response Example

```
{
  "bankOrderId": "string",
  "invoiceId": "string",
  "redirectUrl": "string",
  "session": {
    "token": "string"
  },
  "state": "enum"
}
```

Success Response Example for Consumer defined transfer rates

The following structure extends the default Transfer status response body (e.g. 2 processing gates):

```
{
  "bankOrderId": "string",
  "invoiceId": "string",
  "redirectUrl": "string",
  "session": {
    "token": "string"
  },
  "state": "TRANSFER_FEE_REQUEST",
  "transferFeeList": [
    {
      "assignedId": "string",
      "commissionCentis": "integer",
      "currency": "string"
    },
    {
      "assignedId": "string",
      "commissionCentis": "integer",
      "currency": "string"
    }
  ]
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

Request Builder

7.65.7 Complete Transfer

- [Introduction](#)
- [API URLs](#)
- [Request Parameters](#)
- [Success Response Parameters](#)
- [Fail Response Parameters](#)
- [Request Example](#)
- [Success Response Example](#)
- [Fail Response Example](#)

Introduction

Complete transfer is initiated through HTTPS POST request by using [URLs](#) and the [parameters](#).

API URLs

Note:

The path in API URL should not be hardcoded, as it may be changed in future.

Integration	Production
https://sandbox.doc2.com/paynet/dapi/v1/transfer/{endpointId}	https://api.doc2.com/paynet/dapi/v1/transfer/{endpointId}

Request Parameters

Parameter Name	Description	Value
session		Type: Object
session.accessToken	The session access token of transfer transaction.	Type: String
session.token	The session token of transfer transaction.	Type: String

Success Response Parameters

Parameter Name	Description	Value
invoiceId	Transfer transaction identifier.	Type: String
session		Type: Object
session.token	The session token of transfer transaction.	Type: String

Fail Response Parameters

Parameter Name	Description	Value
error		Type: Object
error.cause	Cause of the error. Possible values: INVALID_REQUEST, INVALID_CONFIGURATION, INTERNAL_ERROR, PROCESSING_FAILED.	Type: Enum
error.code	The unique error code.	Type: String
error.message	Description of the error.	Type: String
invoiceId	Transfer transaction identifier.	Type: String

Request Example

```
{
  "session": {
    "accessToken": "string",
    "token": "string"
  }
}
```

Success Response Example

```
{
  "invoiceId": "string",
  "session": {
    "token": "string"
  }
}
```

Fail Response Example

```
{
  "error": {
    "cause": "enum",
    "code": "string",
    "message": "string"
  },
  "invoiceId": "string"
}
```

7.66 Connecting Party Callbacks

- [General information](#)
- [Customizable Callback URL](#)
- [Callback Parameters](#)
- [Callback Macros](#)
- [Callback Request Example](#)
- [Callback to Status Parameters Mapping](#)
- [Callback Signature Check Example on Java](#)

7.66.1 General information

If Connecting Party specified a callback URL, Payment Gateway sends an HTTP GET message to the callback URL whenever transaction reaches final status, no matter if the result is approved, declined or other [final status](#). This gives Connecting Party better control of how the transaction is processed, for example to add appropriate records to Connecting Party internal accounting system.

Warning: Connecting Party server needs to respond to this GET request with 200 OK status , or else Payment Gateway will continue to send the callback 30 times for 14 days applying the progressive timeline.

Please remember, callbacks are guaranteed to report Connecting Party about transaction

status. Connecting Party on their side must provide a means of preventing receiving the same callback twice - in case of network or other technical problems. It is recommended to check status, type, orderid and client_orderid to prevent duplication of transactions on Connecting Party side.

Callback URLs can be specified by Connecting Party in the following ways:

- By sending `server_callback_url` or `notify_url` in initial transaction request. If `server_callback_url` is sent, Connecting Party will receive a callback only for the initial transaction (ex. sale approved). If `notify_url` is sent instead, Connecting Party will receive a callback for the initial transaction (ex. sale approved) and for any new transaction types applied to the same transaction (ex. reversal, chargeback).
- By specifying the callback URL on endpoint level. Each transaction type can have it's own callback URL.

Only the following ports are allowed in callback URL:

- for HTTP 80, 8080
- for HTTPS 443, 8443

7.66.2 Customizable Callback URL

Simple callback URL contains all parameters listed in [Callback Parameters](#).

Customizable callback URL is a fully defined URL with all the parameters Connecting Party target page or script would require. Customizable URL allows Connecting Party to define own parameter names, whereas the actual parameters values are defined by use of macros with the following format `${parameter_name}`. Thus `gate.doc2.com` substitutes respective parameter values into final customized URL before calling it. Available macros are listed in [Callback Macros](#).

Example of simple callback URL:

`https://connectingparty.com/sale_completed.php`

Example of customized callback URL:

`https://connectingparty.com/sale_completed.php?cardholder_name=${name}&tx_status=${status}&order`

7.66.3 Callback Parameters

Note:

The system automatically adds the following parameters to callback URL.

* - these parameters are not defined by default. Please contact tech support to include these fields in callback.

Callback Parameter	Description
status	See Status List for details.
merchant_order	Connecting Party order identifier, same as <code>client_orderid</code> .
client_orderid	Connecting Party order identifier.

continues on next page

Table 284 – continued from previous page

Callback Parameter	Description
orderid	gate.doc2.com transaction id.
type	Transaction type, sale reversal chargeback
amount	Actual transaction amount. This value can be changed during the transaction flow.
currency	Transaction currency.
descriptor	Payment descriptor of the gate through which the transaction has been processed.
error_code	Error Code. This field won't be included in callback in case transaction status=approved.
error_message	Error Message. This field won't be included in callback in case transaction status=approved.
name	Cardholder Name.
email	Payer's email.
country *	Payer's country (two-letter country code). Please see Country and State Codes for a list of valid country codes.
state *	Payer's state . Please see Country and State Codes for a list of valid state codes. Mandatory for USA, Canada and Australia.
city *	Payer's city.
zip_code *	Payer's ZIP code.
address1 *	Payer's address line 1.
approval-code	Authorization approval code, if any.
last-four-digits	Last four digits of Payer's credit card number.
bin	Bank BIN of Payer's credit card.
card-type	Type of Payer's credit card (VISA, MASTERCARD, etc).
phone	Payer's phone number.
bank-name	Payer's bank name.
card-exp-month	Card expiration month.
card-exp-year	Card expiration year.
gate-partial-reversal	Processing gate support partial reversal (enabled or disabled).
gate-partial-capture	Processing gate support partial capture (enabled or disabled).
reason-code	Reason code for chargeback or fraud operation.
processor-rrn	Bank Retrieval Reference Number.
comment	Comment in case of Return transaction
rapida-balance	Current balance for Connecting Party registered in Rapida system (only if balance check active)
control	Checksum is used to ensure that it is gate.doc2.com (and not a fraudster) that initiates the callback to Connecting Party. This is SHA-1 checksum of the concatenation status + orderid + merchant_order + merchant_control. The callback script MUST check this parameter by comparing it to SHA-1 checksum of the above concatenation.
merchantdata	The value passed by Connecting Party in initial request.
serial-number	Serial number of the request.
processor-tx-id	Transaction id set by processor.
processor-auth-credit-code	Reserved
card-hash-id	Unique hash of the particular card, does not change.
verified-3d-status	Will return AUTHENTICATED if a transaction was approved by 3DS.
processor-credit-rrn	Retrieval Reference Number set by acquirer.

continues on next page

Table 284 – continued from previous page

Callback Parameter	Description
processor-credit-arn	Acquirer card reference number for credit card.
processor-debit-arn	Acquirer card reference number for debit card.
eci	Electronic Commerce Indicator (Visa).
ips-src-payment-product-code	Code for card set by multinational financial service. (Visa/Mastercard).
ips-src-payment-product-name	Decrypted code for card set by multinational financial service. (Visa/Mastercard).
ips-src-payment-type-code	Type of card code set by multinational financial service. (Visa/Mastercard).
ips-src-payment-type-name	Decrypted code for type of card set by multinational financial service. (Visa/Mastercard).
card-country-alpha-three-code	Three letter country code of source card issuer. See Card Country Codes for details.
destination-card-country-alpha-three-code	Three letter country code of destination card issuer. See Card Country Codes for details.
initial-amount	Amount, set in initiating transaction, without any fees or commissions. This value can't change during the transaction flow.
seller-commission *	Total commission for processed transaction.
acquirer-commission *	Acquirer commission for processed transaction.
exchange-rate *	Basic exchange rate for currency conversion.
effective-exchange-rate *	Actual exchange rate, applied during currency conversion.
motivational-message *	This is an optional message which contains extended information about the reason for the declined transaction.
orig-amount	Contains the original request amount if it was converted on auxiliary endpoint in Parallel form integration.
orig-currency	Contains the original request currency if it was converted on auxiliary endpoint in Parallel form integration.

7.66.4 Callback Macros

gate.doc2.com Callback Macros name	Description
\${status}	Transaction status, ex. approved, declined, processing, etc.
\${merchant_order}	Connecting Party order identifier, same as client_orderid.
\${orderid}	gate.doc2.com transaction id.
\${type}	Transaction type, ex. sale, return, chargeback, etc.
\${amount}	Transaction amount.
\${descriptor}	Payment descriptor of the gate through which the transaction has been processed.
\${error_message}	Error message: when \${status} is declined.
\${name}	Cardholder Name.
\${email}	Payer's email.
\${last-four-digits}	Last four digits of Payer's credit card number.
\${bin}	Bank BIN of Payer's credit card.
\${card-type}	Type of Payer's credit card (VISA, MASTERCARD, etc).
\${card-exp-month}	Card expiration month.
\${card-exp-year}	Card expiration year.
\${gate-partial-reversal}	Processing gate support partial reversal (enabled or disabled).

continues on next page

Table 285 – continued from previous page

gate.doc2.com Callback Macros name	Description
<code>\${gate-partial-capture}</code>	Processing gate support partial capture (enabled or disabled).
<code>\${reason-code}</code>	Reason code for chargeback or fraud operation.
<code>\${processor-rrn}</code>	Bank Retrieval Reference Number.
<code>\${approval-code}</code>	Bank approval code.
<code>\${comment}</code>	Comment in case of Return transaction.
<code>\${rapida-balance}</code>	Current balance for Connecting Party registered in Rapida system (only if balance check active).
<code>\${control}</code>	Checksum is used to ensure that it is gate.doc2.com (and not a fraudster) that initiates the callback to Connecting Party. This is SHA-1 checksum of the concatenation status + orderid + merchant order + merchant_control. The callback script MUST check this parameter by comparing it to SHA-1 checksum of the above concatenation.
<code>\${merchantdata}</code>	The value passed by Connecting Party in initial request.

7.66.5 Callback Request Example

```
https://connectingparty.com/api/integration/check/pay/server?token=some_token
&serial-number=b8e5b762-c116-407e-a591-82a458e1
&merchant_order=preauth_1171
&client_orderid=preauth_1171
&processor-tx-id=e0a0572f-2154-737c-8ea7-92410
&orderid=57792
&status=approved
&amount=1.50
&currency=EUR
&descriptor=%D0%90+%D0%94%D0%B5%D0%BD%D0%B3%D0%B8+--+card+registration
&original-gate-descriptor=%D0%90+%D0%94%D0%BD%D1%8C%D0%B3%D0%B8+--+card+registration&gate-
->partial-capture=enabled
&type=preauth
&name=CARDHOLDER+NAME
&card-exp-month=6
&card-exp-year=2024
&email=22701231%40example.com
&processor-rrn=21660934567
&approval-code=265470
&control=bdd11a020f6bsdkfgjh23e24def54991bfb63c5&last-four-digits=0214
&bin=220220&card-type=VISA
&phone=%2B71914454778
&bank-name=Rabobank
&card-hash-id=235479750
&card-country-alpha-three-code=RUS
&ips-src-payment-product-code=VISA
&ips-src-payment-product-name=VISA
&ips-src-payment-type-code=Unknown
&ips-src-payment-type-name=VISA+Unknown
&initial-amount=1.50
&transaction-date=2022-06-15+12%3A37%3A02+CEST
```

7.66.6 Callback to Status Parameters Mapping

Callback parameter Name	Status parameter Name
amount	amount
approval-code	approval-code
bin	bin
card-type	card-type
last-four-digits	last-four-digits
bank-name	bank-name
name	name
first-name	first-name
last-name	last-name
country	country
state	state
city	city
zip_code	zip_code
address1	address1
card-exp-month	card-exp-month
card-exp-year	card-exp-year
client_orderid	merchant-order-id
comment	comment
descriptor	descriptor
dest-bin	dest-bin
dest-card-type	dest-card-type
dest-last-four-digits	dest-last-four-digits
dest-bank-name	dest-bank-name
email	email
purpose	purpose
error_code	error-code
error_message	error-message
gate-partial-capture	gate-partial-capture
gate-partial-reversal	gate-partial-reversal
loyalty-balance	loyalty-balance
loyalty-bonus	loyalty-bonus
loyalty-message	loyalty-message
loyalty-program	loyalty-program
merchant_order	merchant-order-id
merchantdata	merchantdata
orderid	paynet-order-id
original-gate-descriptor	original-gate-descriptor
phone	phone
processor-rrn	processor-rrn
processor-tx-id	processor-tx-id
rapida-balance	rapida-balance
reason-code	reason-code
serial-number	serial-number
status	status
type	transaction-type
initial-amount	initial-amount
seller-commission	seller-commission

continues on next page

Table 286 – continued from previous page

Callback parameter Name	Status parameter Name
acquirer-commission	acquirer-commission
exchange-rate	exchange-rate
effective-exchange-rate	effective-exchange-rate
card-country-alpha-three-code	card-country-alpha-three-code
destination-card-country-alpha-three-code	destination-card-country-alpha-three-code

7.66.7 Callback Signature Check Example on Java

You may see an example how to check the callback signature using JAVA programming language below:

```
import org.junit.Test;

import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import static org.junit.Assert.assertEquals;

public class TestCallbackSignatureExampleTest {

    @Test
    public void test() {
        String digest3 = calculateCallbackSignature("approved", 123, "invoice-1",
        ↪ "AF4B5DE6-3468-424C-A922-C1DAD7CB4509");
        assertEquals("5bc8ee48f9ba37c0fd1e0b052a9bc105c6df87e1", digest3);
    }

    public String calculateCallbackSignature(String aTransactionStatus, long aOrderId,
    ↪ String aMerchantOrderId, String aMerchantControlKey) {
        String text = aTransactionStatus + aOrderId + aMerchantOrderId +
    ↪ aMerchantControlKey;
        byte[] buffer = text.getBytes(StandardCharsets.UTF_8);
        byte[] shaSum = sha(buffer);
        return toHexString(shaSum);
    }

    /**
     * Calculates the SHA-1 digest and returns the value as a <code>byte[]</code>.
     *
     * @param data
     *         Data to digest
     * @return SHA-1 digest
     */
    private static byte[] sha(byte[] data) {
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA");
            return digest.digest(data);
        } catch (NoSuchAlgorithmException e) {
            throw new IllegalStateException("Couldn't calculate SHA-1 digest", e);
        }
    }
}
```

(continues on next page)

(continued from previous page)

```
/**
 * Converts bytes to hex string
 */
private static String toHexString(byte[] data) {
    StringBuilder sb = new StringBuilder();
    for (byte b : data) {
        String hex = Integer.toHexString(0xff & b);
        if (hex.length() == 1) {
            sb.append('0');
        }
        sb.append(hex);
    }
    return sb.toString();
}
}
```

MPOS INTEGRATION

8.1 mPOS API

- [Introduction](#)
- [Mobile App Integration flow](#)
 - [Sale Flow](#)
 - [Preauth Flow](#)

8.1.1 Introduction

mPOS is a device that is able to read card data from plastic cards (like magnetic stripe tracks) or other devices and transmit this data to the Doc2.0 mPOS API in an encrypted form.

mPOS API (MAPI) supports [Sale](#) and [Preauth](#) transactions with status requests and callbacks for transaction status handling.

Depending on the exact acquiring solution, it might be required to implement regular Close Day procedure for bank payment clearing via [/api/v2/close-day/](#) API command. Please contact support manager for details.

8.1.2 Mobile App Integration flow

Doc2.0 team creates a dedicated account on sandbox environment and provides integrator with test ENDPOINTID, merchant_control_key and test mPOS devices with pre-installed encryption key for processing test transactions. mPOS devices (each has a serial number) are handed to integrator along with SDK for communication (see [mPOS SDK for iOS](#) and [mPOS SDK for Android and Java](#)). Integrator team implements Sale and/or Preauth transaction flows and MobileApp-to-mPOS communication tailoring their needs.

Sale Flow

- (2) To implement sale request see </mapi/v1/sale/>.
- (5) To implement callback with final status handling see [Connecting Party Callback](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

Preauth Flow

- (2) To implement preauth request see </mapi/v1/preauth/>.
- (5) To implement callback with final status handling see [Connecting Party Callback](#).
- (7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

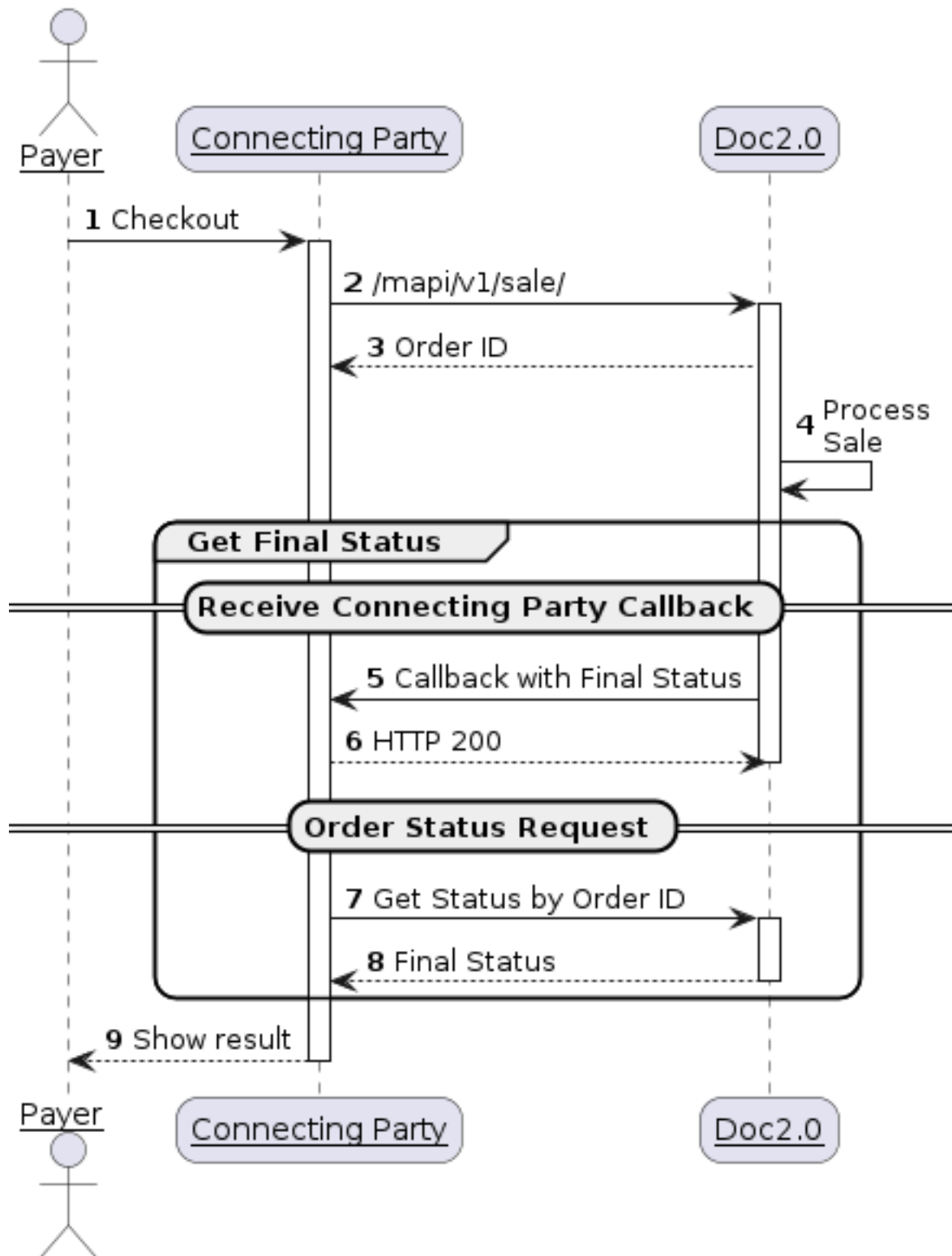
8.2 mPOS SDK for iOS

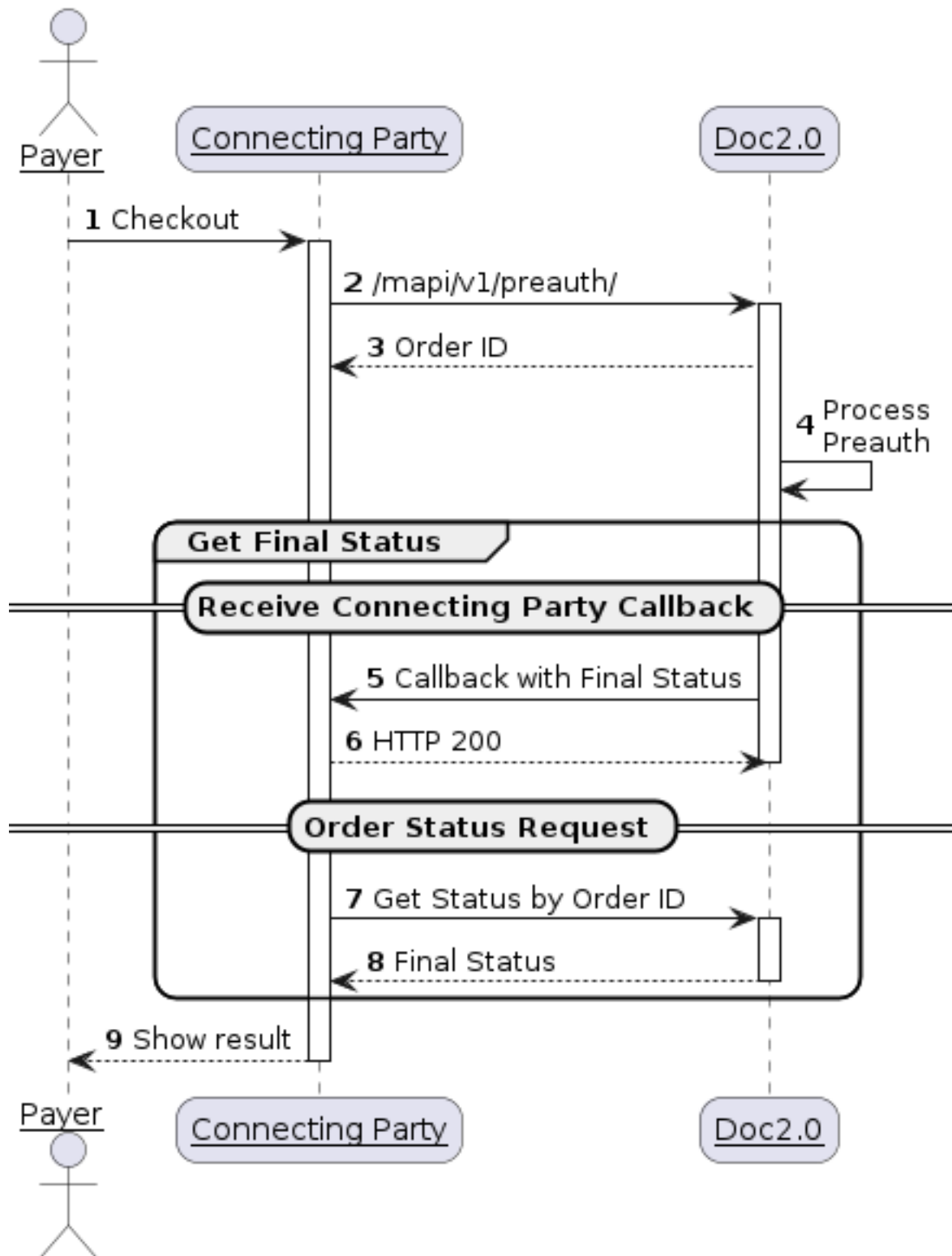
- [Introduction](#)
- [Supported Terminals](#)
- [Integration](#)
 - [Add the SDK to the project](#)
 - [For Miura, Spire and Pax](#)
 - [Sample Code](#)
 - [Useful Links](#)
 - [Test Environment](#)

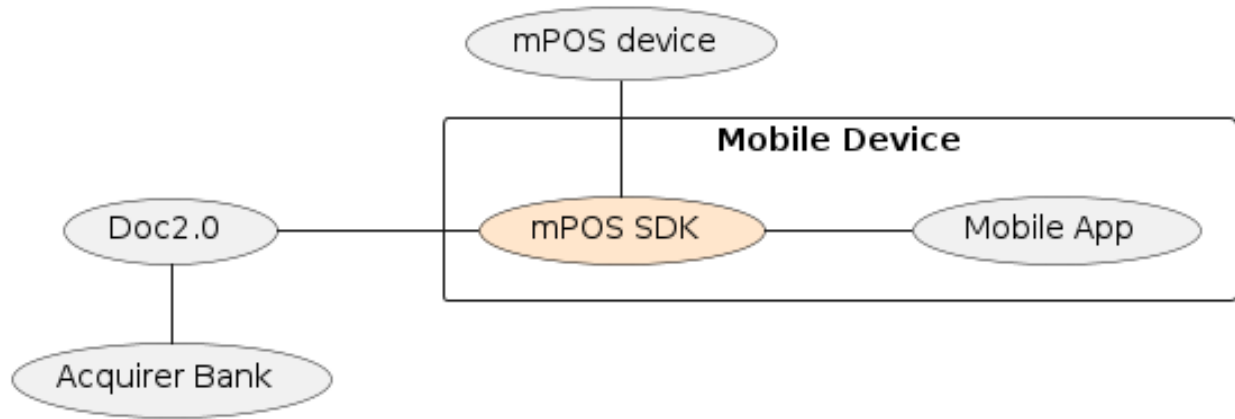
8.2.1 Introduction

Doc2.0 mPOS SDK is a mobile payment software & hardware package that allows to quickly and easily start offering a mobile payment service. Usage of an affordable mobile point of sale (mPOS) terminal allows Connecting party to extend their business with in-person non-cash payments.

Payneteasereader SDK provides fast, ADVT & M-TIP compliant integration with [mPOS](#) terminals in mobile apps. SDK includes header files and a single static library.







8.2.2 Supported Terminals

Vendor	Model	Connectivity
Miura	M006, M007, M010	Bluetooth, USB, Wi-Fi
Spire	SPm2, SPm20	Bluetooth
Verifone	Vx820	Ethernet, USB, RS232
PAX	SP30, D200	Ethernet, USB, RS232

Note:

Requirements: Payneteasereader SDK supports target deployment of iOS version 7.0+ and instruction set armv7+ (including 64-bit), x86_64, i386 (for emulator).

8.2.3 Integration

Add the SDK to the project

Add the following to Podfile:

```
pod "Payneteasereader", :git => 'git@github.com:evsinev/Payneteasereader.git', :tag =>
  ↳ '$VERSION'
```

Please change \$VERSION to the latest version from releases list for or respectively.

For Miura, Spire and Pax

Add the following to *-Info.plist:

```
<key>UISupportedExternalAccessoryProtocols</key>
<array>
  <string>com.miura.shuttle</string>
  <string>com.thyron</string>
  <string>com.paxsz.ipos</string>
</array>
```

Sample Code

Implement the PNEReaderPresenter protocol:

```
- (void)stateChanged:(PNEReaderEvent *)aEvent {
    // displays reader status
    // see an example at https://github.com/payneteasy/ReaderExample/blob/master/ReaderExample/PaymentModule/PaymentPresenter.m#L69
}

- (PNEProcessingContinuation *)onCard:(PNECard *)aCard {
    // provide payneteasy.com account info
    PNEProcessingContinuation * continuation = [PNEProcessingContinuation
        continuationWithBaseUrl:@"https://sandbox.payneteasy.com/paynet"
        merchantLogin:MERCHANT_LOGIN
        merchantKey:MERCHANT_KEY
        merchantEndPointId:END_POINT_ID
        orderInvoiceNumber:[[NSUUID UUID] UUIDString]];

    return continuation;
}

- (void)onCardError:(PNECardError *)aError {
    // deal with the error
    // see an example at https://github.com/payneteasy/ReaderExample/blob/master/ReaderExample/PaymentModule/PaymentPresenter.m#L93
}

- (void)onProcessingEvent:(PNEProcessingEvent *)aEvent {
    // wait for Result event
    // see an example at https://github.com/payneteasy/ReaderExample/blob/master/ReaderExample/PaymentModule/PaymentPresenter.m#L96
}

- (PNEConfigurationContinuation *)onConfiguration {
    return [[PNEConfigurationContinuation alloc]
        initWithBaseUrl:@"https://paynet-qa.clubber.me/paynet/rki"
        merchantLogin:_payment.merchantLogin
        merchantKey:_payment.merchantKey
        merchantEndPointId:_payment.merchantEndPointId
        merchantName:_payment.merchantName
    ];
}
```

Start Reader Manager:

```
PNERedReaderFactory *factory = [[PNERedReaderFactory alloc] init];
PNERedReaderInfo *reader = [PNERedReaderInfo infoWithType:PNERedReaderType_MIURA_OR_SPIRE];
// Note: manager must be a property or a field or a static local variable, to prevent
↳ an elimination
manager = [factory createManager:reader
               amount:[NSDecimalNumber decimalNumberWithString:@"1.00"]
               currency:@"RUB"
               presenter:self];
[manager start];
```

Useful Links

- [Objective-c with Cocoapods](#)²¹
- [Swift with Cocoapods](#)²²
- [Swift with Carthage](#)²³
- Additional samples can be provided by request.
- Other releases can be found [here](#)²⁴.

Test Environment

Amount	Status	Stage
1.00	APPROVED	Purchase, Final Advice
2.00	DECLINED	Purchase
3.00	APPROVED	Purchase
4.00	DECLINED	Purchase, Final Advice
1000.01	APPROVED	Purchase, Final Advice with PIN Online Test. Only test cards can be used in test purposes.

Note:

Any card can be used.

If card asks ARQC then host replies with Authorisation Response Code = 'Z3' (Unable to go online, offline declined).

²¹ <https://github.com/payneteasy/ReaderExample>

²² <https://github.com/payneteasy/ReaderExampleSwift>

²³ <https://github.com/payneteasy/ReaderExampleSwiftCarthage>

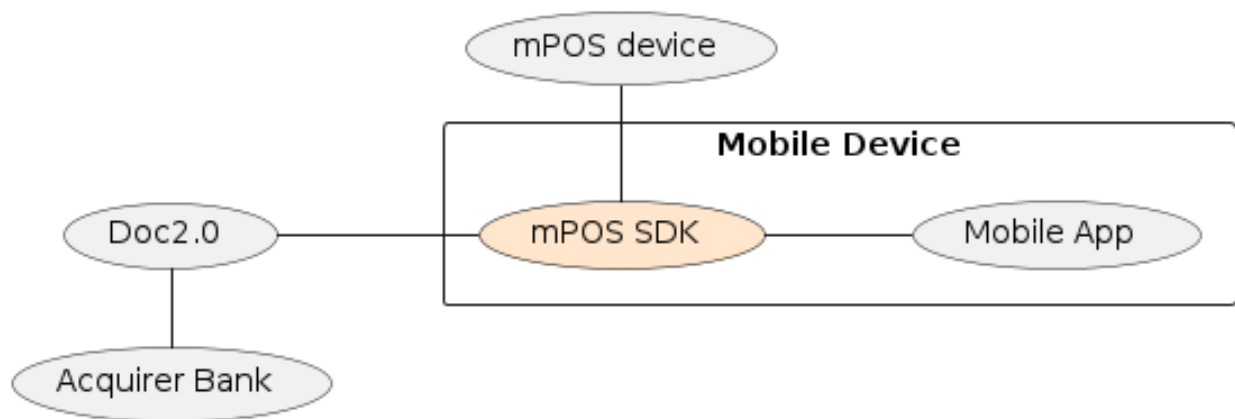
²⁴ <https://github.com/payneteasy/PayneteasyReader/releases>

8.3 mPOS SDK for Android and Java

- [Introduction](#)
- [Supported Terminals](#)
- [Integration](#)
 - [Setup maven](#)
 - [Setup gradle](#)
 - [For Bluetooth Readers \(Android\)](#)
 - [For USB terminals \(Android\)](#)
 - [For Miura \(Java\)](#)
 - [Sample Code](#)
- [Useful Links](#)

8.3.1 Introduction

Doc2.0 mPOS SDK is a mobile payment software & hardware package that allows to quickly and easily start offering a mobile payment service. Usage of an affordable mobile point of sale (mPOS) terminal allows Connecting party to extend their business with in-person non-cash payments.



Payneteasereader SDK provides fast integration with [mPOS](#) terminals in mobile apps.

Features

- Runs both on Android and Java
- MSR, EMV Contact, EMV Contactless transactions
- Signature or PIN based transactions
- Remote key injection
- Remote firmware update
- Remote mPOS configuration

- Certified with OpenWay. The first mobile POS Chip&PIN solution in Russia -
- Custom LCD messages for all stages (Miura only)
- Unified API for all readers
- Implemented [Doc2.0 Server API](#)
- Many transaction types: sale, preauthorization, capture, cancel, reversal, partial reversal and etc.

Note:

Requirements: SDK supports target deployment of Android from 2.2 and Oracle JVM (tested on 1.8)

8.3.2 Supported Terminals

Vendor	Model	Connectivity
Miura	M006, M007, M010	Supports MSR, Contact, Contactless transactions. With or without PIN. Remote keys injection, remote configuration and remote MPI and OS update
Spire	SPm2	Bluetooth
Verifone	Vx820	Ethernet, USB, RS232
PAX	SP30	Ethernet, USB, RS232

Note:

All terminals support remote configuration and firmware update.

Warning:

Deprecated from the 1st of January 2015:

GD Seed: Integrated with the official SDK.

GD Seed: Own SDK. With memory and performance optimization.

ID Tech Unimag II: Integrated with the official SDK.

ID Tech Unimag II: Own SDK. Memory and performance optimization. Extends supported phone models.

Bluebamboo P25.

Warning:

Evaluated:

Datecs DRD50, DRD10

Datecs MPED400, Bluepad 50

Wizar POS

BBPOS

8.3.3 Integration

The SDK includes a maven repository - <http://paynet-qa.clubber.me/reader/maven/> For Miura Shuttle model: Use TEST on emulators for events.

Setup maven

[Android Maven repository can be found here²⁵](#). Add the repository to pom.xml:

```
<repository>
  id>reader-repo</id>
  <name>reader repo</name>
  <url>http://paynet-qa.clubber.me/reader/maven</url>
</repository>
```

Setup gradle

[Android Gradle repository can be found here²⁶](#). Add the following to repositories section:

```
repositories {
  ...
  maven { url "https://jitpack.io" }
  maven { url "http://paynet-qa.clubber.me/reader/maven" }
  ...
}
```

- Add the following to dependencies:

```
dependencies {
  ...
  def readerVersion = // check the latest version :ref:`here<https://github.com/
  ↪payneteasys/reader-example/wiki/Changelog>`
  compile 'com.payneteasys.android.reader:api:' + readerVersion
  compile ('com.payneteasys.android.reader:lib:' + readerVersion)

  compile 'com.payneteasys.android.reader.readers:readers-common-bluetooth:' +
  ↪readerVersion
  compile ('com.payneteasys.android.reader.readers:readers-miura:' + readerVersion) {
    exclude group:'junit', module:'junit'
  }
  ...
}
```

For additional details, please refer to the complete [Gradle example available here²⁷](#).

²⁵ <https://github.com/payneteasys/reader-example>

²⁶ <https://github.com/payneteasys/reader-example-gradle>

²⁷ <https://github.com/payneteasys/reader-example-gradle>

For Bluetooth Readers (Android)

Add the following to AndroidManifest.xml:

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
```

For USB terminals (Android)

- Add new file res/xml/device_filter.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

  <!-- 0x1234 / 0x0101 PAX / S80 -->
  <usb-device vendor-id="4660" product-id="257" class="255" subclass="0" protocol="0"/
  <!-- 0x11CA / 0x0219 VeriFone Inc / Trident USB Device 1.1 / bInterfaceClass = 10
  CDC Data -->
  <usb-device vendor-id="4554" product-id="537" class="10" subclass="0" protocol="0"/>

</resources>
```

- Add any activity element to AndroidManifest.xml:

```
<intent-filter>
  <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
</intent-filter>
<meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
  android:resource="@xml/device_filter" />
```

- Add to the activity:

```
public class MainActivity extends Activity {

  private final UsbPermissionResolver usbPermissionResolver = new
  UsbPermissionResolver();

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    ...
    usbPermissionResolver.checkPermission(getIntent(), this);
  }

  @Override
  protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    usbPermissionResolver.checkPermission(intent, this);
  }

  @Override
  protected void onDestroy() {
    super.onDestroy();
  }
}
```

(continues on next page)

(continued from previous page)

```

        usbPermissionResolver.unregister(this);
    }

    ...

}

```

Note:

Please see [this commit](#)²⁸ for more information.

For Miura (Java)

Please see the example app-2 how to run and assemble the application for Miura Readers to run on JVM.

app-3-webstart shows how to run the application from a browser via Java WebStart:

- [mPOS SDK on Linux](#)²⁹
- [Mpos SDK + Java webstart on Mac OS X](#)³⁰
- [mPOS SDK on Windows](#)³¹

Sample Code

Implement the IReaderPresenter interface:

```

@Override
public ProcessingContinuation onCard(BankCard bankCard) {

    setStatus("onCard: %s", bankCard);

    return ProcessingContinuation.Builder
        .startSaleOnline()
        .processingBaseUrl ( Config.SERVER_BASE_URL)
        .merchantLogin     ( Config.MERCHANT_LOGIN )
        .merchantControlKey ( Config.MERCHANT_KEY   )
        .merchantEndPointId ( Config.END_POINT_ID   )
        .orderDescription   ( "test description"    )
        .orderInvoiceNumber ( "invoice-"+System.currentTimeMillis())
        .orderMerchantData  ( "custom merchant data for a internal use")
        .customerPhone      ( "+7 499 918-64-41"    )
        .customerEmail      ( "info@payneteasy.com" )
        .customerCountry    ( "RUS"                )
        .listener(new IProcessingStageListener() {
            @Override
            public void onStageChanged(ProcessingStageEvent aEvent) {

```

(continues on next page)

²⁸ <https://github.com/payneteasy/reader-example-gradle/commit/2aa02c47f38cef4f3a673e98fc5d5ecb98720656>

²⁹ <https://www.youtube.com/watch?v=rLSx3SrKMQA>

³⁰ <https://www.youtube.com/watch?v=OgdUfHqBvml>

³¹ <https://www.youtube.com/watch?v=I6PPO-j9KSk>

(continued from previous page)

```

        setStatus("processing: %s", aEvent);
    }
    })
    .build();

}

@Override
public void onReaderSerialNumber(String aKsn) {
    setStatus("onReaderSerialNumber: %s", aKsn);
}

@Override
public void cardReaderStateChanged(CardReaderEvent cardReaderEvent) {
    setStatus("cardReaderStateChanged: %s", cardReaderEvent);
}

@Override
public void onCardError(CardError cardError) {
    setStatus("onCardError: %s", cardError);
}

@Override
public void onReaderNotSupported(CardReaderProblem aProblem) {
    setStatus("onReaderNotSupported: %s", aProblem);
}

@Override
public void onAudioData(short[] shorts, int i) {
    // for visualization
}

@Override
public ConfigurationContinuation onConfiguration() {
    return new ConfigurationContinuation.Builder()
        .configDir          ( new File(activity.getFilesDir(), "miura-config
→"))
        .configurationBaseUrl ( Config.SERVER_CONFIG_URL    )
        .merchantLogin        ( Config.MERCHANT_LOGIN       )
        .merchantControlKey    ( Config.MERCHANT_KEY        )
        .merchantEndPointId    ( Config.END_POINT_ID        )
        .build();
}

```

Note:

Fill the Config.MERCHANT_LOGIN, Config.MERCHANT_KEY, Config.END_POINT_ID with your own values.

Starts the Reader Manager:

```

CardReaderInfo cardReader = CardReaderInfo.TEST;
BigDecimal amount = new BigDecimal(1);
String currency = "RUB";

```

(continues on next page)

(continued from previous page)

```
SimpleCardReaderPresenter presenter = new SimpleCardReaderPresenter(this, statusView);
cardReaderManager = CardReaderFactory.findManager(this, cardReader, presenter, amount,
↪ currency, null);`
```

Deal with cardReaderManager:

- ICardReaderManager.startSaleSession() starts a sale session
- ICardReaderManager.stopSession() stops a session and releases resources

8.3.4 Useful Links

- [App 4 rxtx qbang](#)³²
- [Java APIdocs](#)³³
- Additional samples can be provided by request.
- [Other releases can be found here](#)³⁴.

8.4 mPOS FAQ

- [Первоначальные действия при сбое работы терминала](#)
- [Ошибка -01](#)
- [Ошибка -02](#)
- [Ошибка 2005 - PROCESSOR_OR_GATE_NOT_CONFIGURED_PROPERLY: Invalid rates](#)
- [Ошибка Attempt using magnetic strip if chip available](#)
- [Ошибка Gate 3822 got blocked by 83677493 order, nested error is blocked.by;83677492;-03](#)
- [Ошибка Терминалы не проводят платежи](#)
- [Оповещение Locked terminals - 1517,1537,1573,1574...](#)
- [Как снимать блокировку гейта](#)
- [Платежи не проходят и даже попыток не видно](#)
- [Оповещение для UNKNOWN статусов](#)
- [Проверка вводил ли клиент пинкод или платеж прошел без ввода пинкода](#)
- [Перезагружается терминал во время оплаты](#)
- [Отображение оборотов по терминалу](#)
- [Как проверить заблокирован ли гейт](#)
- [Ошибка 05 - Do not honour](#)

³² <https://github.com/payneteas/reader-example/tree/master/app-4-rxtx-qbang>

³³ <http://paynet-qa.clubber.me/reader/apidocs/>

³⁴ <https://github.com/payneteas/reader-example/wiki/Changelog>

- Ошибка 88
- Очень долго грузится терминал. Примерно секунд 30-50. Выполняли 3+энтер и 2+энтер. Ситуация не изменилась
- Идет подключение к терминалу и так остается в этом состоянии
- Что если с утра провести принудительную сверку итогов, то в 10 часов её уже не будет?
- Написано, что автоотмена, с клиента списались деньги, касса выдала ошибку
- Терминал не работает - деньги не списаны попробуйте другую карту. Неизвестная ошибка
- Подскажите, есть ли какие блокировки по терминалу? Не отражаются попытки оплаты и идут автоотмены
- Что означают обозначения EMV_CONTACT...?
- Подскажите, есть ли какие блокировки по терминалу? Не отражаются попытки оплаты и идут автоотмены
- <PNEReaderEvent: state=Unknown (22), self.message=192.168.2.255-192.168.2.156:27015>...
- Терминал не работает - деньги не списаны попробуйте другую карту. Неизвестная ошибка
- При попытке провести оплату пишет - "нет соединения с терминалом"
- При попытке проведения - ошибка, что банк отклонил платеж, но в списке заказов нет этого платежа
- Не удалось отменить транзакцию
- В Интеграционной панели в ЛК ошибка - Signature verification failed.

8.4.1 Первоначальные действия при сбое работы терминала

В случае отсутствия описания ошибки в документации, при сбое работы терминала выполните следующие действия:

- Перезагрузить терминал, при необходимости, остальное оборудование.
- В случае ошибки сразу выгрузить логи с терминала.
- Проверить соединения: кабели, интернет и т.д.
- Переподключить терминал.
- Проверить настройки терминала в зависимости от типа подключения: USB/Ethernet.

Если проблема осталась, обратитесь в техническую поддержку.

8.4.2 Ошибка -01

No EMV final advice, autoreversed, Decline: 94029.

Не пришел EMV Final Advice на сервер. Причины:

- нет соединения с интернетом
- вытащили карту из терминала
- ошибка в ПО на терминале, в SDK или в приложении

Определить точную причину можно по логам. Найти в логах строчку с Invoice (например, у заказа 84822609 invoice - 461106A7-2E37-4D97-B594-8A1639691957).

Если в логах: CHIP_CARD_REMOVED, то вытащили карту.

Пример из логов:

```
Web Apr 10 10:41:40 GMT +03:00 2019 DEBUG sdk.CardReaderPresenter PRESENTER:
PayneteasyReaderPresenter.cardReaderStateChanged(CardReaderEvent {state=SPIRE_48_
→ TERMINATE_TRANSACTION_EVENT,message=<SpireTerminateTransaction48Event
  messageIdIdentifier = 48
  sequenceNumber      = 1
  reason               = 0x0E_14_CHIP_REMOVED
```

При этом деньги должны были вернуться автоматом.

8.4.3 Ошибка -02

Please try again later, Decline: 94030.

Причины:

- гейт заблокирован. Это видно по мониторингу на странице . Строчка “Locked terminals - “. В значении указываются номера гейтов.
- подключены несколько терминалов (Spire или Miura) к одному гейту

Если заблокирован гейт, нужно зайти в личный кабинет системы, перейти в настройки гейта, выключить его, а затем включить.

Закрытие гейтов

См. отчет Close Days Report в меню “Отчёты” - “Все отчёты”

8.4.4 Ошибка 2005 - PROCESSOR_OR_GATE_NOT_CONFIGURED_PROPERLY: Invalid rates

Проверить корректность настройки тарифов в системе. Тариф менеджера не может быть больше тарифа реселлера, банковский тариф не может быть больше тарифа реселлера или менеджера. Транзакция при этом не успешна, можно исправить статус на “declined”.

8.4.5 Ошибка Attempt using magnetic strip if chip available

Проводят карту по магнитной полосе, а у нее есть чип.

8.4.6 Ошибка Gate 3822 got blocked by 83677493 order, nested error is blocked.by;83677492;-03

Попытка провести еще одну транзакцию, во время прохождения другой.

8.4.7 Ошибка Терминалы не проводят платежи

Проверить, не заблокирован ли гейт.

8.4.8 Оповещение Locked terminals - 1517,1537,1573,1574...

- заходим на страницу гейта
- проверяем, если ошибка закрытия дня
- если есть, значит банковский день закрылся неуспешно
- нажимаем на кнопку “Считать успешно закрытым”
- нажать на кнопку “Закрыть день”
- если день опять не закроется, нужно звонить в банк

8.4.9 Как снимать блокировку гейта

- перейти на страницу “Настройки” - “Поиск настроек”
- ввести номер гейта
- нажать на кнопку Disable, потом Enable

8.4.10 Платежи не проходят и даже попыток не видно

Проверить в меню “Инструменты” - “Интеграционная панель”

8.4.11 Оповещение для UNKNOWN статусов

Включить Failed session initiators в Tasks Будет приходить по 1-2 письма будет на каждый UNKNOWN.

8.4.12 Проверка вводил ли клиент пинкод или платеж прошел без ввода пинкода

В поле Order Description пишем как прошла транзакция например, с вводом PIN - будет так Order description:EMV_CONTACT_PIN_OFFLINE.

8.4.13 Перезагружается терминал во время оплаты

Проблема с питанием терминала по USB.

Если терминал подключен к блоку питания, то проблема в БП, надо заменить его.

8.4.14 Отображение оборотов по терминалу

Для правильной суммы лучше пользоваться Performance отчетом из меню “Отчёты” - “Все отчёты

Однако, нужно учитывать, что у каждой торговой точки мерчанта должен быть свой endpoint в разрезе ридеров мы не считаем, т.к. в течение дня ридер может сломаться и его заменят.

НЕЛЬЗЯ заводить один терминал в банке и подключать к нему больше одного ридера.

8.4.15 Как проверить заблокирован ли гейт

Экран “Инструменты” - “Monitoring” - “Tasks”. Строчка: Locked mobile terminals.

8.4.16 Ошибка 05 - Do not honour

Банк эмитент отклонил транзакцию. Причину отклонения нужно узнавать у банка, который выпустил карту.

8.4.17 Ошибка 88

Данная ошибка связана с некорректной работой ключа для бесконтактной оплаты. Для устранения данной ошибки необходимо связаться с нашей службой технической поддержки.

8.4.18 Очень долго грузится терминал. Примерно секунд 30-50. Выполнили 3+энтер и 2+энтер. Ситуация не изменилась

Терминал подбирает ip. Проверьте какой сейчас ip и зафиксируйте его в админке.

8.4.19 Идет подключение к терминалу и так остается в этом состоянии

Ошибка There was a problem while creating a connection to the remote service. У терминала настройки под Ethernet, а должны быть под usb, необходимо поменять настройки на самом терминале. Также возможно, что терминалу не хватает мощности: попробуйте подключить терминал по usb к компьютеру или еще куда-то, возможно не хватает мощности запитать терминал.

8.4.20 Что если с утра провести принудительную сверку итогов, то в 10 часов её уже не будет?

Сверка будет утром в любом случае.

8.4.21 Написано, что автоотмена, с клиента списались деньги, касса выдала ошибку

Терминал не дожидается статуса платежа (либо интернет либо еще что-то) в итоге отваливается по таймауту, хотя операция проходит. В таких случаях вероятнее всего проблема связана с тем, что используется версия SDK "sdk_version" = "0.2.27-3" и старше, в более поздних версиях, например (0.2.28-3) есть повторные запросы и таких ситуаций не происходит, плюс логируются все запросы статуса операции. SDK предоставляем мы, релиз её в вашем приложении не на нашей стороне. Информация о версии SDK доступна в reader-logs в первой строчке, например: "Creating manager version 0.2.27-3".

Последние версии SDK можно посмотреть на сайте:

- iOS - <https://github.com/payneteasy/PayneteasyReader/releases>
- Android - <https://github.com/payneteasy/reader-example/wiki/Changelog>

8.4.22 Терминал не работает - деньги не списаны попробуйте другую карту. Неизвестная ошибка

Неправильно настроен терминал. Забыли указать Параметры / Редактирование / Smart-Sale / SSL без KKM = Да.

8.4.23 Подскажите, есть ли какие блокировки по терминалу? Не отражаются попытки оплаты и идут автоотмены

Почему терминалы PAX и Verifone иногда делали возврат: Все дело было в периодической (раз в час) проверке соединения с сервером. На всех терминала поменяли настройки - отключили проверку соединения, сделали ее только один раз, при включении. У нас сейчас практически все терминалы обновляются раз в день, какие не обновятся, тем нужно будет сделать 3+ENTER. Это касается только тех терминалов, которые работают по Ethernet, с USB подключением проблем нет.

8.4.24 Что означают обозначения EMV_CONTACT...?

8.4.27 Терминал не работает - деньги не списаны попробуйте другую карту. Неизвестная ошибка

Это проявляется выводом сертификата ssl в логах:

- Sat Oct 20 22:33:36 GMT+03:00 2018 DEBUG sdk.Sa63Connection16Eve Connection event is Sa63Connection16Event{

```
operation    = OPEN    host    = 185.15.175.23    port    = 20002    _91_pul-  
sar_tags    = 'CA:---BEGIN    CERTIFICATE---    MIIDTTCCAjWgAwIBAgIJALYxPm-  
mQmP+xMA0GCSqGSIb3DQEBCwUAMB4xHDAaBgNV
```

8.4.28 При попытке провести оплату пишет - “нет соединения с терминалом”

Проверить, включен ли Bluetooth.

8.4.29 При попытке проведения - ошибка, что банк отклонил платеж, но в списке заказов нет этого платежа

Зайти в меню “Инструменты” - “Интеграционная панель” и првоерить запросы по терминалу (Endpoint)

В старых версиях программы Cassby может высылаться нулевая сумма.

8.4.30 Не удалось отменить транзакцию

Проверить, не заблокирован ли гейт.

Если банк возвращает двухбуквенный код ответа, то следует обратиться в банк с номером RRN и кодом ответа банка.

8.4.31 В Интеграционной панели в ЛК ошибка - Signature verification failed.

Ошибка связана с подписью запроса. Проверьте в настройках терминала endpoint, login, control key.

8.5 mPOS Test

- [Doc2.0 QA Server](#)

8.5.1 Doc2.0 QA Server

Doc2.0 QA Environment - it is the [Sandbox server](#)³⁵ for connectivity issues testing. Server behaviour depends on the amount:

1.00 - A regular payment without any network error - result is APPROVED.
Other amounts - result is DECLINE.

Please see error types below.

Amount	Result	Description
1.00	APPROVED	A regular payment without any network error.
1.01	APPROVED	OrderID was not received on sale stage.
5.00	APPROVED	Timeout (70 seconds) for the first status poll.
5.01	APPROVED	Timeout (70 seconds) for sale.
5.02	APPROVED	Timeout (70 seconds) for advice.
5.03	APPROVED	Timeout (70 seconds) for the status poll after advice.
6.00	APPROVED	HTTP Status 503 for sale.
6.01	APPROVED	HTTP Status 503 for the first status poll.
6.02	APPROVED	HTTP Status 503 for advice.
6.03	APPROVED	HTTP Status 503 for the status poll after advice.

8.6 mPOS API Ru

- [Закрытие дня](#)
 - [Принудительное закрытие дня](#)
 - [Автоматическое закрытие дня](#)
 - [Автоматическое закрытие дня по заранее установленному времени](#)
 - [Удаленное закрытие дня](#)
 - [Инструмент снятия блокировки для шлюза](#)
 - [Инструмент отладки для удаленного закрытия дня](#)

³⁵ <https://paynet-qa.clubber.me/paynet>

8.6.1 Заккрытие дня

Описание закрытия дня

Деньги переводятся на расчетный счет торговца после закрытия дня. При неуспешном завершении процедуры закрытия дня или если процедура закрытия вовсе не инициируется торговцем, операция клиринга не производится, таким образом деньги не зачисляются на расчетный счет торговца.

Если по какой-либо причине не удастся провести закрытие дня, используя «Удаленное закрытие дня», необходимо провести операцию закрытия дня вручную через личный кабинет или обратиться за помощью в службу поддержки.

Данная операция доступна не для всех эквайеров, за информацией обращайтесь в службу поддержки.

Для закрытия дня доступны следующие способы:

[Принудительное закрытие дня](#)

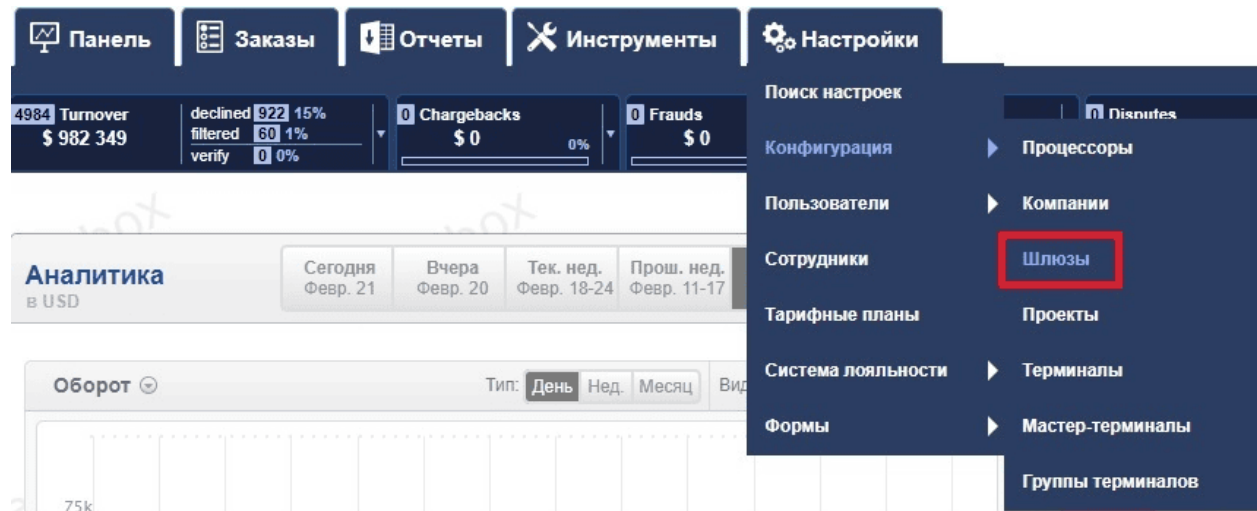
[Автоматическое закрытие дня](#)

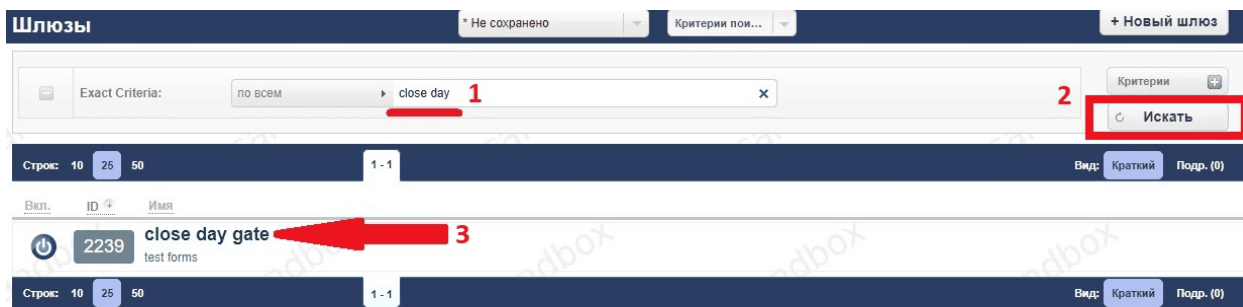
[Автоматическое закрытие дня по заранее установленному времени](#)

[Удаленное закрытие дня](#)

Принудительное закрытие дня

Для принудительного закрытия дня менеджеру или его сотруднику в личном кабинете системы надо перейти в настройки шлюза, для которого требуется закрыть день.





На странице шлюза нажмите кнопку “Закрыть день” и подтвердите закрытие дня на всплывающем окне. На странице начнет отображаться окно “Закрытие дня для ‘имя шлюза’”, дождитесь завершения операции.

После успешного закрытия дня на странице шлюза будет отображена информация о дате закрытия и сумме транзакций с момента последнего закрытия дня для данного шлюза до текущего закрытия дня.

Warning: Во время закрытия дня шлюз блокируется и транзакции по нему проходить не будут.

Автоматическое закрытие дня

Для автоматического закрытия дня менеджеру или его сотруднику в личном кабинете системы надо перейти в настройки шлюза, для которого требуется закрывать день автоматически.

На странице шлюза нажмите кнопку “Редактировать”, поставьте галочку напротив пункта “Автоматическое закрытия дня” и нажмите “Сохранить”.

При автоматическом закрытии дня оптимальное время закрытия выбирается системой автоматически (ближе к концу рабочего дня по МСК).

Шлюз

Общие

2239

close day gate

Статус:

3D:

Описание:

MID:

Теги:

Процессор:

Менеджер:

Тарифный план банка:

Дилер:

Тарифный план дилера:

Валюта:

Дескриптор:

Проект:

Сервис лояльности:

Время закрытия дня:

Внешняя форма:

Мин. сумма транзакции:

Макс. сумма транзакции:

Метод:

Последняя дата закрытия дня:

Максимальная задержка закрытия дня (дн.):

Автоматическое закрытия дня:

Удаленное закрытие дня:

Метод 2:

test processor

test manager

Zero rate

-

Нет

-

-

-

-

-

Да

Нет

-

Подтвердите действие на сайте

Вы уверены, что хотите закрыть день для шлюза?

2

Да

Нет

3

Заккрытие дня для close day gate

Имя: Запускается...

Создано: 2019.02.21 12:59:49

Общее/прошло времени: 0 из 0

Шагов пройдено: 0 из 0

1

Закреть день

Шлюз

Общие

2239

close day gate

Клонировать

Редактировать

Статус:

Включен

3D:

Нет

Описание:

test forms

MID:

-

Теги:

-

Процессор:

test processor

Менеджер:

test manager

Тарифный план банка:

Zero rate

Дилер:

-

Тарифный план дилера:

-

Валюта:

RUB

Дескриптор:

test

Проект:

-

Сервис лояльности:

-

Время закрытия дня:

-

Внешняя форма:

Нет

Мин. сумма транзакции:

-

Макс. сумма транзакции:

-

Метод:

-

Последняя дата закрытия дня:

2019-02-21 12:59:55.0

Максимальная задержка закрытия дня (дн.):

-

Автоматическое закрытия дня:

Да

Удаленное закрытие дня:

Нет

Метод 2:

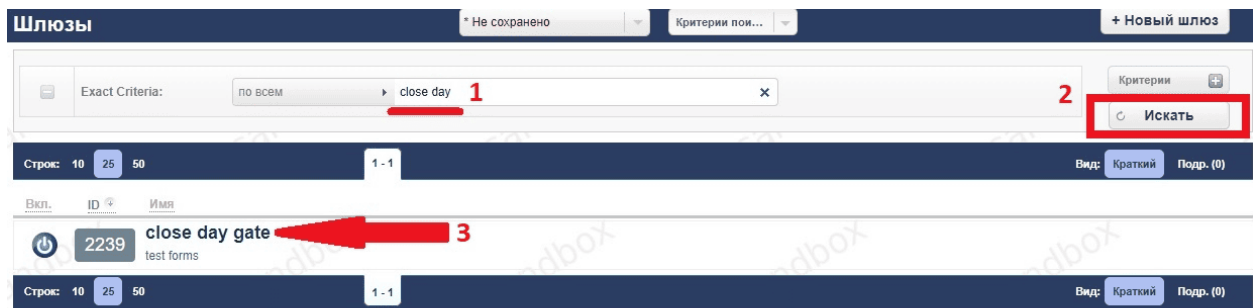
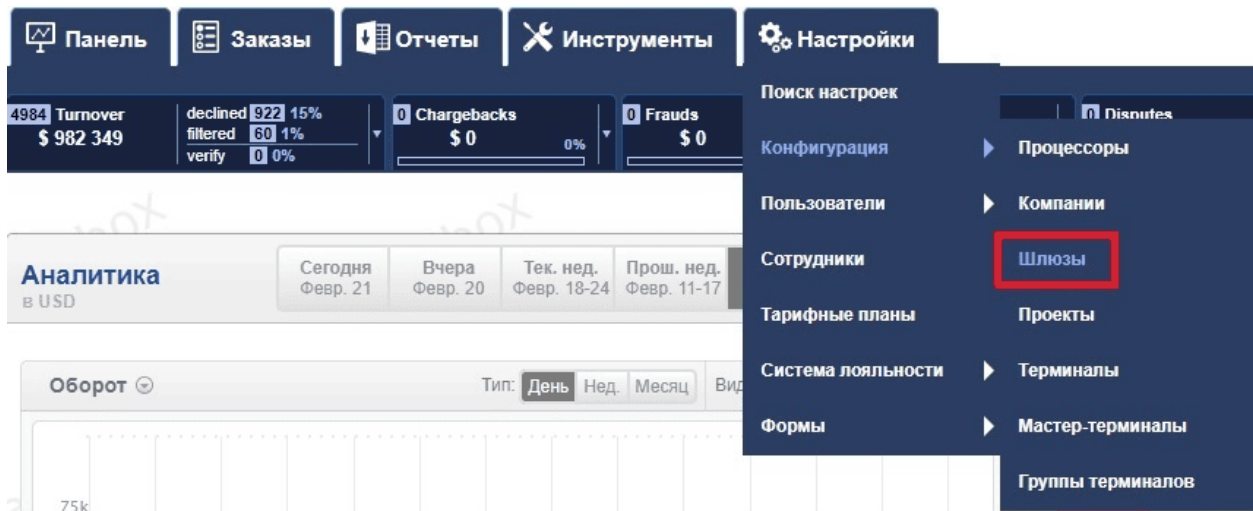
-

Закрыть день

Строк: 10 25 50

1 - 1

Дата	Ответ банка	Сумма транзакций
21.02.2019 12:59	OK	0.00



Шлюз
Общие

2239

close day gate

Использовать 3D Secure

☐

Статус

Включен

Имя

*

close day gate

Описание

*

test forms

MID

Теги

Процессор

*

test processor

Тарифный план банка

*

Zero rate

Дилер

-

Дескриптор

*

test

Компания

Искать

Программа лояльности

Искать

Закрывать день автоматически

☐

Внешняя форма

☐

Мин. сумма транзакции

Макс. сумма транзакции

Метод

Максимальная задержка закрытия

Автоматическое закрытия дня

☒

1

Удаленное закрытие дня

☐

Метод 2

2

Сохранить

Отмена

Warning: Во время автоматического закрытия дня шлюз блокируется и транзакции по нему проходить не будут.

После успешного автоматического закрытия дня на странице шлюза будет отображена информация о дате закрытия и сумме транзакций с момента последнего закрытия дня для данного шлюза до текущего закрытия дня.

Автоматическое закрытие дня по заранее установленному времени

Для автоматического закрытия дня по заданному времени менеджеру или его сотруднику в личном кабинете системы надо перейти в настройки шлюза, для которого требуется закрывать день автоматически в установленное время.

На странице шлюза нажмите кнопку “Редактировать”, поставьте галочку напротив пункта “Закрывать день автоматически”, появится пункт “Время закрытия дня”, нужно выставить время закрытия дня, затем нажать “Сохранить”.

Обратите внимание, что пункт “Автоматическое закрытия дня” должен быть выключен, иначе время закрытия дня будет выбрано автоматически системой.

Закрытие дня будет выполняться в заданное время, после успешного закрытия дня на странице шлюза будет отображена информация о дате закрытия и сумме транзакций с момента последнего закрытия дня для данного шлюза до текущего закрытия дня.

Warning: Во время автоматического закрытия дня по заданному времени шлюз блокируется и транзакции по нему проходить не будут.

Шлюз

Общие

2239

close day gate

Клонировать

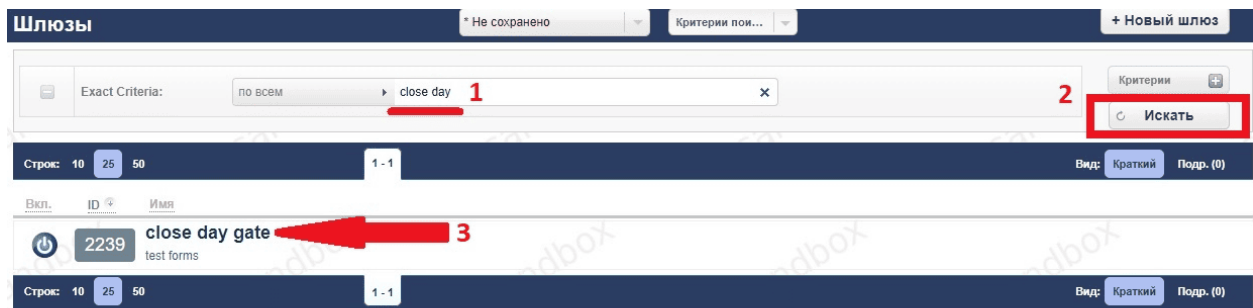
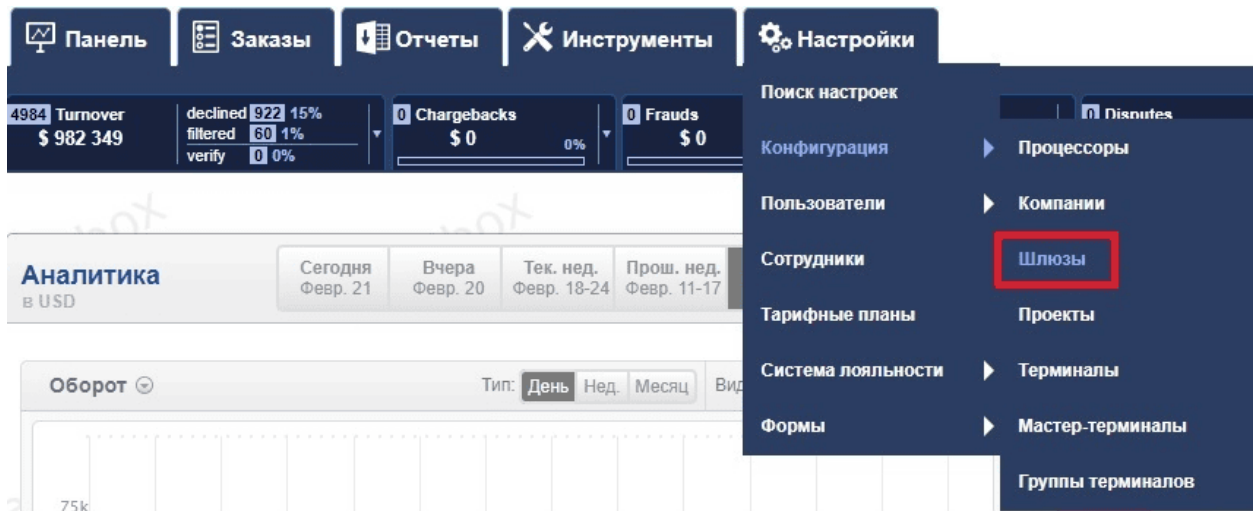
Редактировать

Статус:	Включен
3D:	Нет
Описание:	test forms
MID:	-
Теги:	-
Процессор:	test processor
Менеджер:	test manager
Тарифный план банка:	Zero rate
Дилер:	-
Тарифный план дилера:	-
Валюта:	RUB
Дескриптор:	test
Проект:	-
Сервис лояльности:	-
Время закрытия дня:	-
Внешняя форма:	Нет
Мин. сумма транзакции:	-
Макс. сумма транзакции:	-
Метод:	-
Последняя дата закрытия дня:	2019-02-21 12:59:55.0
Максимальная задержка закрытия дня (дн.):	-
Автоматическое закрытия дня:	Да
Удаленное закрытие дня:	Нет
Метод 2:	-

Закрыть день

Строк: 10 25 50 1 - 1

Дата	Ответ банка	Сумма транзакций
21.02.2019 12:59	OK	0.00



Шлюз
Общие

2239

close day gate

Использовать 3D Secure

☐

Статус

Включен

Имя

*

close day gate

Описание

*

test forms

MID

Теги

Процессор

*

test processor

Q

Тарифный план банка

*

Zero rate

Дилер

-

Дескриптор

*

test

Компания

Искать

Q

Программа лояльности

Искать

Q

Закрывать день автоматически

☒

1

Время закрытия дня

22

:

00

2

Внешняя форма

☐

Мин. сумма транзакции

Макс. сумма транзакции

Метод

Максимальная задержка закрытия

Автоматическое закрытие дня

☐

3

Удаленное закрытие дня

☐

Метод 2

4

Сохранить

Отмена

Шлюз

Общие

2239

close day gate

Статус:

Включен

3D:

Нет

Описание:

test forms

MID:

-

Теги:

-

Процессор:

test processor

Менеджер:

test manager

Тарифный план банка:

Zero rate

Дилер:

-

Тарифный план дилера:

-

Валюта:

RUB

Дескриптор:

test

Проект:

-

Сервис лояльности:

-

Время закрытия дня:

14:02

Внешняя форма:

Нет

Мин. сумма транзакции:

-

Макс. сумма транзакции:

-

Метод:

-

Последняя дата закрытия дня:

2019-02-21 14:02:06.0

Максимальная задержка закрытия дня (дн.):

-

Автоматическое закрытие дня:

Нет

Удаленное закрытие дня:

Нет

Метод 2:

-

Закрыть день

Строк: 10 25 50

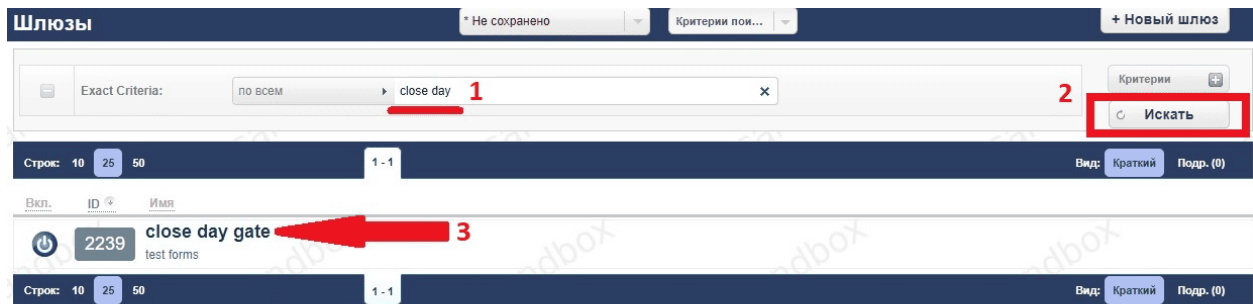
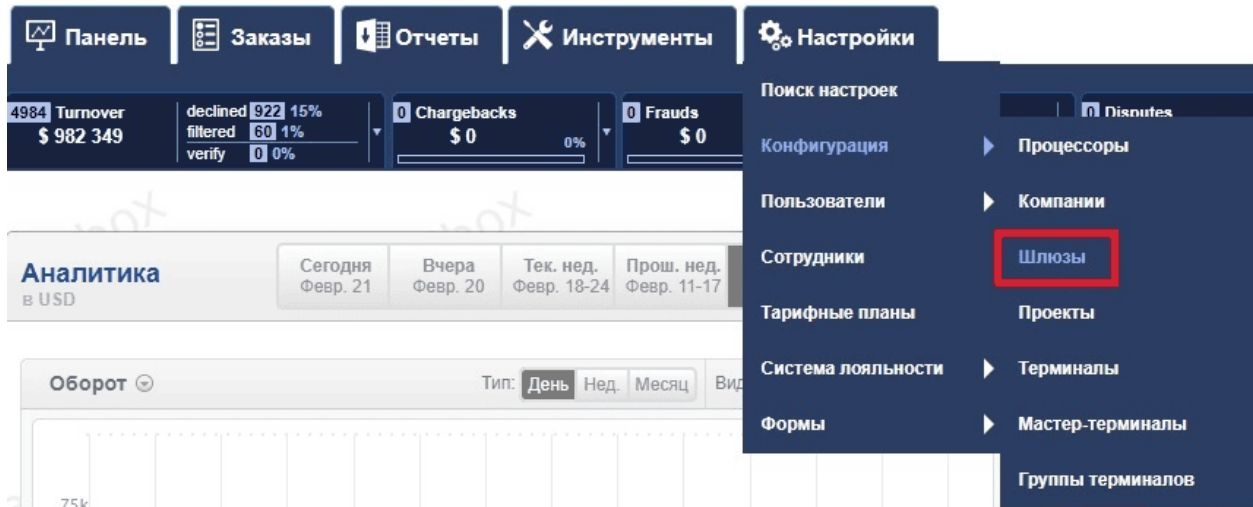
1 - 2

Дата	Ответ банка	Сумма транзакций
21.02.2019 14:02	OK	0.00

Удаленное закрытие дня

Удаленное закрытие дня позволяет закрыть день сразу на всех шлюзах определенного терминала. Для этого необходимо для ВСЕХ шлюзов ассоциированных с терминалом установить в настройках “Удаленное закрытие дня”.

Для удаленного закрытия дня менеджеру или его сотруднику в личном кабинете системы надо перейти в настройки шлюза, для которого требуется включить удаленное закрытие дня.



На странице шлюза нажмите кнопку “Редактировать”, поставьте галочку напротив пункта “Удаленное закрытие дня”, при необходимости, задайте число дней через которое будет срабатывать автоматическое закрытие дня, начиная с последнего успешного удаленного закрытия дня - пункт “Максимальная задержка закрытия дня (дн.)”, автоматически устанавливается 3 дня, затем нажмите “Сохранить”.

Шлюз
Общие

2239

close day gate

Использовать 3D Secure

☐

Статус

Включен

Имя

*

close day gate

Описание

*

test forms

MID

Теги

Процессор

*

test processor

Q

Тарифный план банка

*

Zero rate

▼

Дилер

-

Дескриптор

*

test

Компания

Искать

Q

Программа лояльности

Искать

Q

Закрывать день автоматически

☐

Внешняя форма

☐

Мин. сумма транзакции

Макс. сумма транзакции

Метод

Максимальная задержка закрытия

2

Автоматическое закрытия дня

☐

Удаленное закрытие дня

☒

Метод 2

Сохранить

Отмена

Warning: Во время удаленного закрытия дня шлюзы блокируются и транзакции по ним проходить не будут.

Для инициации Удаленного закрытия дня необходимо отправить HTTP запрос следующего вида:

```
POST https://${hostname}/paynet/api/v2/close-day/{endpointId}
```

Аутентификация пользователя происходит по OAuth 1.0, описание: [oauth](#)

Тело запроса

В теле запроса укажите данные со следующей структурой:

```
{
  "day_close_request_id": {string},
  "sale_approved_count": {string},
  "sale_approved_sum": {string},
  "reversal_approved_count": {string},
  "reversal_approved_sum": {string}
}
```

Имя свойства	Тип(длина)	Описание
Обязательные параметры		
day_close_request_id	string(8-32)	Уникальный идентификатор запроса. Может содержать буквы и цифры
Необязательные параметры		
sale_approved_count	string	Число подтвержденных sale транзакций
sale_approved_sum	string	Сумма по подтвержденным sale транзакциям
reversal_approved_count	string	Число подтвержденных reversal транзакций
reversal_approved_sum	string	Сумма по подтвержденным reversal транзакциям

Тело ответа

Пользователь генерирует идентификатор запроса и посылает запрос на закрытие дня. Если этот идентификатор используется впервые - начинается инициализация закрытия дня для гейта(ов), ассоциированного(ых) с переданным идентификатором терминала. Пользователю возвращается асинхронный ответ со статусом 'starting'.

Пример структуры ответа следующий:

```
{
  "response": {
    "status": "starting",
    "gates": [ {
      "gateId": 193,
      "isDayClosing": true,
      "epntId": 1
    }, {
      "gateId": 82,
      "isDayClosing": true,
      "epntId": 1
    }, {
      "gateId": 317,
      "isDayClosing": true,
      "epntId": 1
    }
  ],
  "theRequestSerialNumber": "00000000-0000-0000-0000-00000000001c"
}
```

Имя свойства	Тип	Описание
status	string	Статус операции закрытия дня
gateId	long	Идентификатор шлюза по которому происходит закрытие дня
isDayClosing	boolean	Флаг закрытия дня, при значении true - шлюз заблокирован
epntId	long	Свойство, присваиваемое банком терминалу при его настройке в системе банка
theRequestSerial-Number	string	Серийный номер запроса закрытия дня

Запрос с идентификатором `day_close_request_id`, уже использованным ранее, выполняет функцию статус-запроса. После первичного (инициирующего запроса) пользователь может продолжать посылать запросы по тому же адресу с тем же самым идентификатором запроса. В данном случае, этот идентификатор всегда будет ассоциирован с процедурой закрытия дня, которую он инициировал. Пока операция по закрытию дня в процессе выполнения - пользователю будет возвращаться асинхронный ответ со статусом 'processing'.

Пример структуры ответа следующий:

```
{
  "response": {
    "status": "processing",
    "gates": [ {
      "gateId": 193,
      "isDayClosing": true,
      "epntId": 1
    }, {
      "gateId": 82,
      "isDayClosing": true,
```

(continues on next page)

(continued from previous page)

```

    "epntId": 1
  }, {
    "gateId": 317,
    "isDayClosing": true,
    "epntId": 1
  }
],
"theRequestSerialNumber": "00000000-0000-0000-0000-00000000001e"
}

```

Имя свойства	Тип	Описание
status	string	Статус операции закрытия дня
gateId	long	Идентификатор шлюза по которому происходит закрытие дня
isDayClosing	boolean	Флаг закрытия дня, при значении true - шлюз заблокирован
epntId	long	Свойство, присваиваемое банком терминалу при его настройке в системе банка
theRequestSerial-Number	string	Серийный номер запроса закрытия дня

Информация о закрытом дне появится в ответе от сервера по мере ее поступления от процессоров (в случае одновременного закрытия нескольких гейтов). После закрытия дня гейта(ов) статус ответа примет значение 'finished', вне зависимости от того, завершилось ли закрытие дня на всех гейтах успешно, или часть гейтов (или все) закрылись с ошибкой.

Warning: Если во время удаленного закрытия дня шлюз(ы) останутся в заблокированном состоянии (в теле ответа при "status": "finished" для шлюза - "isDayClosing": true), воспользуйтесь инструментом снятия блокировки в личном кабинете системы, который описан далее в этом разделе.

Пример структуры ответа следующий:

```

{
  "response": {
    "status": "finished",
    "gates": [ {
      "gateId": 193,
      "isDayClosing": false,
      "gateCloseDate": "Mon Dec 10 14:44:21 MSK 2018",
      "epntId": 1,
      "saleApprovedCount": "2334",
      "saleAmount": "87328428482.00",
      "reversalApprovedCount": "123",

```

(continues on next page)

(continued from previous page)

```

        "reversalAmount": "12393423.00",
        "code39": "000",
        "RRN": "2034829434"
      }, {
        "gateId": 82,
        "isDayClosing": false,
        "gateCloseDate": "Mon Dec 10 14:44:21 MSK 2018",
        "epntId": 1,
        "saleApprovedCount": "2334",
        "saleAmount": "87328428482.00",
        "reversalApprovedCount": "123",
        "reversalAmount": "12393423.00",
        "code39": "000",
        "RRN": "2034829434"
      }, {
        "gateId": 317,
        "isDayClosing": false,
        "gateCloseDate": "Mon Dec 10 14:44:21 MSK 2018",
        "epntId": 1,
        "saleApprovedCount": "2334",
        "saleAmount": "87328428482.00",
        "reversalApprovedCount": "123",
        "reversalAmount": "12393423.00",
        "code39": "000",
        "RRN": "2034829434"
      }
    ],
    "theRequestSerialNumber": "00000000-0000-0000-0000-00000000002c"
  }
}

```

Имя свойства	Тип	Описание
status	string	Статус операции закрытия дня
gateId	long	Идентификатор шлюза по которому происходит закрытие дня
isDayClosing	boolean	Флаг закрытия дня, при значении true - шлюз заблокирован
gateCloseDate	string	Дата закрытия дня шлюза
epntId	long	Свойство, присваиваемое банком терминалу при его настройке в системе банка
saleApproved-Count	string	Число подтвержденных sale транзакций
saleAmount	string	Сумма по подтвержденным sale транзакциям
reversalApproved-Count	string	Число подтвержденных reversal транзакций
reversalAmount	string	Сумма по подтвержденным reversal транзакциям
code39	string	Результат закрытия, ответ банка на вызов процедуры закрытия гейта
RRN	string	Номер закрытия дня сгруппированный по ID терминала
theRequestSerial-Number	string	Серийный номер запроса закрытия дня

Если во время выполнения процедуры закрытия дня пользователь посылает повторный

запрос на закрытие дня с новым идентификатором запроса - сервер вернет ответ со статусом 'error' и с описанием ошибки. Использовать повторно идентификатор этого запроса для дальнейшего инициирования закрытия дня бессмысленно, т.к. статус ошибки - это финальный статус и идентификатор запроса всегда будет ассоциирован с этим запросом.

Пример структуры ответа следующий:

```
{
  "response": {
    "status": "error",
    "errorCode": "99",
    "errorMessage": "gate is closing already"
  },
  "theRequestSerialNumber": "00000000-0000-0000-0000-000000000020"
}
```

Имя свойства	Тип	Описание
status	string	Статус операции закрытия дня
errorCode	string	Код ошибки
errorMessage	string	Описание ошибки
theRequestSerialNumber	string	Серийный номер запроса закрытия дня

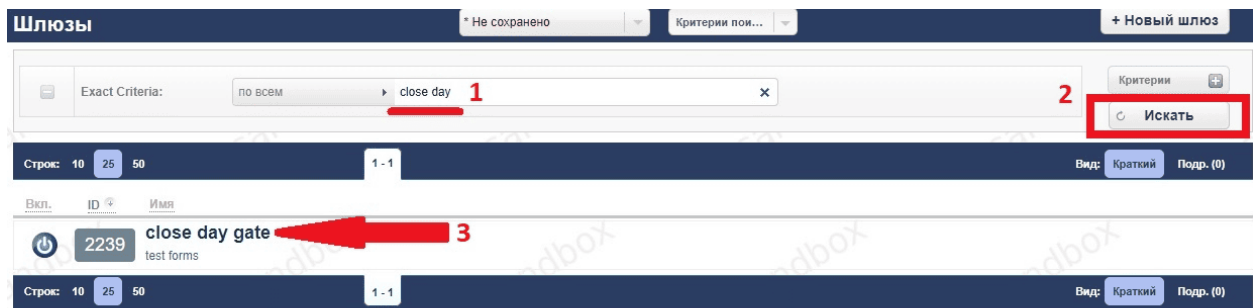
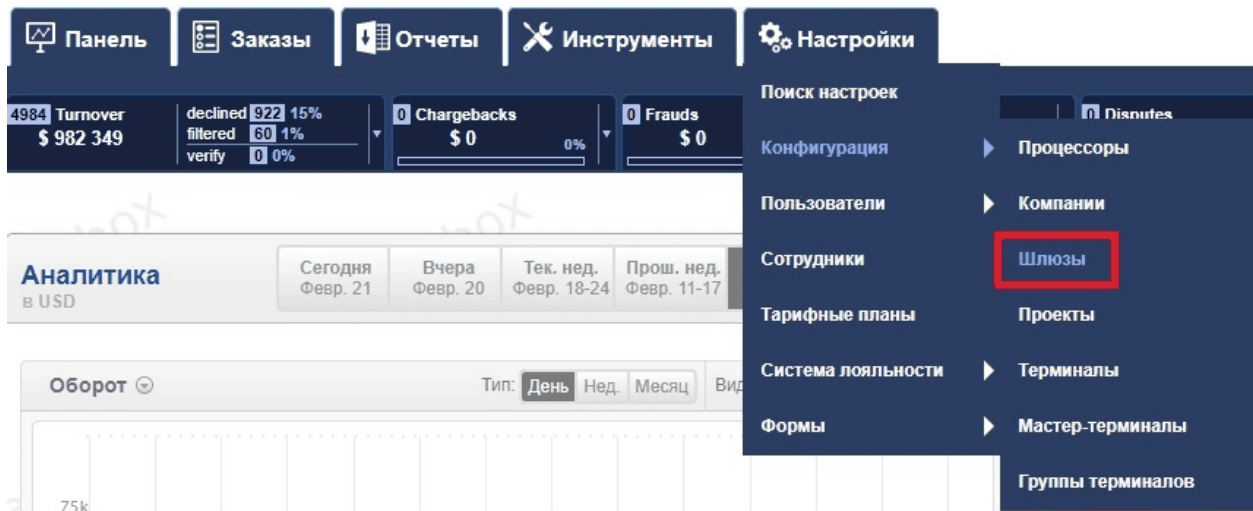
Если пользователь инициирует запрос с новым идентификатором запроса на терминал, по которому уже было закрытие дня, и часть ассоциированных с терминалом гейтов закрылись с ошибкой (и находятся в закрытом состоянии) - сервер пришлет ответ со статусом 'error' и соответствующим описанием ошибки.

Warning: Для разрешения этой проблемы проведите операцию закрытия дня вручную в личном кабинете системы.

Инструмент снятия блокировки для шлюза

Если во время проведения удаленного закрытия дня, один или несколько шлюзов остались в заблокированном состоянии (в теле ответа при "status": "finished" для шлюза - "isDayClosing": true), Вы можете снять блокировку в личном кабинете системы и, при необходимости, закрыть день вручную.

Для снятия блокировки на шлюзе менеджеру или его сотруднику в личном кабинете системы надо перейти в настройки шлюза, для которого требуется снять блокировку.



На странице шлюза нажмите кнопку “Locks”, для вызова инструмента снятия блокировки.

В меню “Current locks” отображена текущая информация о блокировке шлюза:

- Lock session ID - идентификатор сессии блокировки.
- Start day closing date - дата начала закрытия дня.
- Bank terminal lock status - блокировка терминала для проведения транзакций (Blocked/Free).
- Initial gate job status - статус задачи по инициализации шлюза.
- Close day job status - статус задачи по закрытию шлюза.
- Is day closing - шлюз в стадии закрытия, т.е. находится ли в заблокированном состоянии (Y/N). Y - шлюз заблокирован, N - блокировки нет.

Доступны следующие команды снятия блокировки:

- CLEAR_TRANSACTION_LOCK - сбросить сессию из-за которой произошла блокировка.
- CLEAR_INIT_ATTEMPT - сбросить статус задачи по инициализации шлюза.
- CLEAR_CLOSING_DAY_ATTEMPTS - сбросить статус задачи по закрытию шлюза.
- CLEAR_CLOSING_DAY_FLAG - сбросить задачу по закрытию шлюза, снять блокировку.

Отметьте галочкой нужный пункт меню и нажмите кнопку “Update”.

После выполнения команды сброса, повторно откройте инструмент снятия блокировки и проверьте, что в меню “Current locks” изменился статус сбрасываемого параметра. Например, если шлюз был заблокирован, то после снятия блокировки параметр “Is day closing” = N.

Инструмент отладки для удаленного закрытия дня

Чтобы воспроизвести вызов API, введите все данные из исходного запроса, включая маркеры проверки подлинности. Не забудьте установить nonce и timestamp значения, которые Вы использовали. Запрос, подписанный с помощью OAuth, должен совпадать со значением в инструменте отладки ниже, вне зависимости от используемой библиотеки и языка программирования. Если подписи отличаются, значит в Вашем коде подписи OAuth есть ошибка.

Шлюз

Общие

2239

close day gate

Клонировать

Редактировать

Статус:	Включен
3D:	Нет
Описание:	test forms
MID:	-
Теги:	-
Процессор:	test processor
Менеджер:	test manager
Тарифный план банка:	Zero rate
Дилер:	-
Тарифный план дилера:	-
Валюта:	RUB
Дескриптор:	test
Проект:	-
Сервис лояльности:	-
Время закрытия дня:	-
Внешняя форма:	Нет
Мин. сумма транзакции:	-
Макс. сумма транзакции:	-
Метод:	-
Последняя дата закрытия дня:	2019-02-22 08:30:50.0
Максимальная задержка закрытия дня (дн.):	2
Автоматическое закрытия дня:	Нет
Удаленное закрытие дня:	Да
Метод 2:	-

Закрыть день

Locks

Close Days Locks

Current locks

Lock	Value
Lock session ID	
Start day closing date	
Bank terminal lock status	
Initial gate job status	
Close day job status	Y
Is day closing	N

Clear locks

☐ CLEAR_TRANSACTION_LOCK

☐ CLEAR_INIT_ATTEMPT

☐ CLEAR_CLOSING_DAY_ATTEMPTS

☒ CLEAR_CLOSING_DAY_FLAG

Update

8.7 mPOS Terminals

8.7.1 Pax Manual

- Руководство администратора
 - Перевод терминала Pax в режим работы через USB-порт
 - Переключение терминала в режим работы с USB порта на Ethernet
 - Настройка терминала на работу без кассы
 - Создание нового кассира
 - Вход меню Администратора
 - Вход в системное меню
- Руководство пользователя
 - Подключение терминала к сети Ethernet
 - Нахождение IP-адреса терминала в сети
 - Автоматический поиск терминала (только для IOS)
 - Обновление конфигурации
 - * Способ №1
 - * Способ №2
 - Отправка логов на сервер
 - * Способ №1
 - * Способ №2
- FAQ
 - Каждое утро Pax при первой оплате очень долго чем-то занимается, после чего выдает ошибку “Оплата не прошла, используйте другую карту”.
 - При попытке оплаты или загрузке возникает ошибка “Нет соединения с хостом”. Терминал подключен по USB.
 - Подключённый к стороне кассе Pax не принял карту, но на кассу отдал команду печати фискального документа.
 - Как осуществить сверку итогов принудительно на терминале?
 - Эвотор теряет связь с терминалом по USB (на Эвоторе отображается ошибка “Прерывание связи”).

Руководство администратора

Перевод терминала Рах в режим работы через USB-порт

- 1) Войти в меню администратора, см. ниже.
- 2) Установить следующие параметры:
 - Параметры / Редактирование / SmartSale / Связь с банком / Через кассу
 - Параметры / Редактирование / SmartSale / Порт терминала / USB
 - Параметры / Редактирование / SmartSale / SSL без ККМ / Да
- 3) Отключить терминал от USB и подключить его обратно.

Переключение терминала в режим работы с USB порта на Ethernet

- 1) Войти в меню администратора, см. ниже.
- 2) Установить следующие параметры:
 - Параметры / Редактирование / SmartSale / Связь с банком / Через терминал
 - Параметры / Редактирование / SmartSale / Порт терминала / Ethernet
- 3) Отключить терминал от USB и подключить его обратно.

Настройка терминала на работу без кассы

- 1) Войти в меню администратора, см. ниже.
- 2) Установить следующие параметры:
 - Параметры / Редактирование / SmartSale / Режим ККМ / Касса или Терминал (по умолчанию - Касса)
- 3) После установки параметра терминал перезагрузится и в меню появится иконка для инициации платежа.

Note:

В этом режиме терминала Рах SP30 печать слипов (чеков платежной системы) невозможна, т.к. принтер отсутствует.

Создание нового кассира

- 1) Войти в меню администратора, см. ниже.
- 2) Установить следующие параметры:
 - Параметры / Управление доступом / Управление правами / Новый оператор - создать нового оператора в группу “Кассиры” и установить пароль (на этапе “Введите карту доступа” нажать Enter).

- Параметры / Управление доступом / Авто логин - выбрать нужного оператора (в случае необходимости).

Вход меню Администратора

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем нажать ENTER.
- 2) Ввести пароль 1234 и нажать ENTER.
- 3) Выбрать пункт меню “Блокировка терм.” и нажать ENTER.
- 4) Выбрать пункт “Администратор” и ввести пароль Администратора.
- 5) Дождаться логотипа или надписи Ожидание команды от ККМ, затем нажать ENTER.
- 6) Ввести пароль 1234 и нажать ENTER.

Вход в системное меню

- 1) При включении зажать кнопку “Func”.
- 2) При появлении надписи SELF-TEST SUCCESS зажать кнопку “Cancel”.

Руководство пользователя

Подключение терминала к сети Ethernet

- 1) Подключить кабель к разъему с надписью ETH.
- 2) На роутере найти IP-адрес терминала по его MAC-адресу и закрепить его на DHCP-сервере роутера за терминалом. MAC-адрес терминала написан на желтой этикетке на нижней панели терминала.
- 3) При включении терминал должен успешно установить соединение с сервером.
- 4) Выгрузить логи на сервер (2+ENTER) и запросить у службы поддержки IP-адрес для терминала в системе управления.

Нахождение IP-адреса терминала в сети

Поиск IP-адреса Pax/Verifone в сети производится самостоятельно. Указание диапазона не обязательно.

Ограничения:

- Маска сети не должна выдавать больше 500 IP адресов.
- При нахождении больше одного терминала отобразится ошибка.

Note:

В случае присутствия больше одного терминала в сети, может быть указан его идентификатор (например, `terminal_id=12345678`) после чего терминал может быть проверен.

Правила для заполнения поля `address`:

- В случае присутствия в сети одного терминала и количество IP-адресов меньше 500, нет необходимости вводить данные.
- В случае присутствия в сети больше одного терминала и количество IP-адресов меньше 500, необходимо ввести идентификатор терминала (например, `terminal_id=12345678`).
- В случае присутствия в сети одного терминала и количество IP-адресов больше 500 IP-адресов, необходимо указать диапазон поиска IP-range (например, `range=192.168.0.1-192.168.0.254`).
- В случае присутствия в сети больше одного терминала и количество IP-адресов больше 500, необходимо ввести идентификатор терминала или диапазон поиска IP-range на выбор.

Note:

Установка программы для поиска терминала на [iOS³⁶](#) и [Android³⁷](#).

Автоматический поиск терминала (только для IOS)

- В настройках терминала прописать диапазон, в котором будет находится терминал (например, `range=10.45.69.1-10.45.69.254:27015`).

Обновление конфигурации

Способ №1

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем зажать клавиши 3 и ENTER одновременно.
 - 2) Ввести пароль 1234 и нажать ENTER.
 - 3) После отображения сообщения Журнал содержит операции, возможны финансовые потери. Продолжить? нажать ENTER.
- В случае подключения терминала по USB, после отображения сообщения Загрузка параметров будет отложена нажать ENTER.
 - В случае подключения терминала по Ethernet, подключение к сети и загрузка программ пройдет автоматически.

³⁶ <https://dengisend.ru/demo-ios/SocketScanner/install.html>

³⁷ <https://mpos.pne.io/verifone/scanner/scanner.apk>

Способ №2

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем нажать ENTER.
- 2) Ввести пароль 1234 и нажать ENTER.
- 3) Выбрать пункт Параметры, нажав кнопку 6, затем нажать ENTER.
- 4) Нажать на ENTER (курсор на пункте ЗАГРУЗКА ПАРАМ.).
- 5) После отображения сообщения Журнал содержит операции, возможны финансовые потери. Продолжить? нажать ENTER.
 - В случае подключения терминала по USB, после отображения сообщения Загрузка параметров будет отложена нажать ENTER. Затем необходимо произвести оплату, во время которой будет произведено обновление конфигурации.
 - В случае подключения терминала по Ethernet, подключение к сети и загрузка программ пройдет автоматически.

Отправка логов на сервер

Способ №1

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем зажать клавиши 2 и ENTER одновременно.
- 2) В случае подключения терминала по USB, после отображения сообщения Выгрузка логов будет отложена нажать ENTER.
 - В случае подключения терминала по Ethernet, подключение к сети и загрузка программ пройдет автоматически.

Способ №2

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем нажать ENTER.
- 2) Ввести пароль 1234 и нажать ENTER.
- 3) Выбрать пункт Параметры, нажав кнопку 6, затем нажать ENTER.
- 4) Выбрать пункт меню Логирование.
- 5) Выбрать пункт Выгрузка лога.
- 6) В случае подключения терминала по USB, после отображения сообщения Выгрузка логов будет отложена нажать ENTER. Затем необходимо произвести оплату, во время которой будет произведено обновление конфигурации.
 - В случае подключения терминала по Ethernet, подключение к сети и загрузка программ пройдет автоматически.

FAQ

Каждое утро Рах при первой оплате очень долго чем-то занимается, после чего выдает ошибку “Оплата не прошла, используйте другую карту”.

В случае подключения терминала по USB, происходит сверка итогов. Процесс сверки может происходить только во время проведения транзакции.

Note:

Процесс сверки логов происходит каждый день в 10:00 GMT+0.

Рекомендация: Проведение пустой транзакции утром следующего дня.

При попытке оплаты или загрузке возникает ошибка “Нет соединения с хостом”. Терминал подключен по USB.

Рекомендация: Отключить терминал, включить и проверить настройки для работы по USB.

Note:

Необходимо использование USB-кабеля старого типа, т.к. работа USB-кабеля нового типа может быть некорректной.

Возможен недостаток питания терминала от кассы АТОЛ-150. Можно поставить USB-hub с отдельным питанием или поставить Verifone Vx820.

В случае наличия провода с разветвлением на LAN, возможна организация работы через роутер. Также возможно дополнительное питание терминала через разъемы микро-usb/мини-usb, в случае наличия их в кабеле.

Подключённый к стороне кассы Рах не принял карту, но на кассу отдал команду печати фискального документа.

Логи по сторонним кассам есть только у вас, у нас в этом случае не пишутся.

Как осуществить сверку итогов принудительно на терминале?

Рекомендация: Зажать на терминале клавиши 4 и Enter одновременно.

Эвотор теряет связь с терминалом по USB (на Эвоторе отображается ошибка “Прерывание связи”).

- Полностью удалить терминал с Эвотора, перезагрузить Эвотор и терминал.
- Переподключить терминал к другому порту.
- Привязать терминал, проверить настройки Эвотора - что указан верный номер терминала и стоит галка по умолчанию.
- Проверить настройки терминала:

Параметры / Редактирование / SmartSale / Связь с банком. Параметры / Редактирование / SmartSale / Порт терминала. Параметры / Редактирование / SmartSale / SSL без ККМ. Если есть возможность переподключить терминал по Ethernet кабелю.

8.7.2 Spire Manual

- [Руководство пользователя](#)
 - [Подключение терминала по Bluetooth к IOS](#)
 - [Настройка терминала](#)
 - [Сброс настроек Bluetooth](#)
 - [Настройка Bluetooth](#)
- [FAQ](#)
 - [Бесконтактные платежи картами МИР на Spire SPm2 не проходят](#)
 - [Бесконтактные платежи на Spire SPm2 работают через раз](#)

Руководство пользователя

Подключение терминала по Bluetooth к IOS

Для создания пары mPOS терминала и IOS-устройства необходимо выполнить следующие действия:

1. Нажать желтую кнопку на терминале.
2. В появившемся меню выбрать пункт №1.
 - В случае запроса терминалом кода супервайзера, необходимо ввести 8497 и подтвердить код, нажав зеленую кнопку на терминале.
 - В случае отображения сообщения Идет спаривание, необходимо ожидание обнаружения iOS-устройством терминала по Bluetooth.

3. Проверить отображение iOS-устройства в разделе Bluetooth (устройство “PostMate” или “SPxxx”, где x - любая цифра).
4. В случае отображения устройства в доступных для подключения устройств Bluetooth, выбрать его и подключиться.
5. На дисплее IOS-устройства отобразится шестизначный код, который необходимо набрать на терминале и подтвердить зеленой кнопкой.
6. После этого будет создана пара mPOS-терминала с iOS-устройством.

Настройка терминала

Терминалы Spire настраиваются удаленно. В них заливаются следующие файлы:

- cakeys.cfg - CA ключи
- contactless.cfg - настройки бесконтакта
- currency_rub.cfg - настройки валют (Рубли)

В 99% случаев эти файлы загружаются бесппроблемно, однако иногда:

1. Терминал сбрасывает настройки (производитель).
2. Проблемы с Bluetooth.

Также в терминалы загружаются два файла firmware: boot и masterfile. Они весят 300KB и 150KB соответственно. При заливке по Bluetooth через iPad возникают ошибки при передаче файлов. В этом случае рекомендуется проверить версию firmware после заливки. Если не получилось залить, то нужно найти место, где нет других устройств с Bluetooth и повторить попытку, заново зайдя на страницу “Spire Terminals” и нажав на кнопку “Upload Firmware”.

Сброс настроек Bluetooth

Настройка Bluetooth

FAQ

Бесконтактные платежи картами МИР на Spire SPm2 не проходят

Бесконтактные платежи картами МИР на Spire SPm2 НЕ работают.

Бесконтактные платежи на Spire SPm2 работают через раз

Подключить терминал по USB, чтобы он питался от сети перманентно.

В случае, если проблемы не устранились, обновить конфигурацию терминала.

В случае, если проблемы не устранились, отправить терминал производителю по гарантии.

8.7.3 Verifone Manual

- [Руководство администратора](#)
 - [Перевод терминала Verifone в режим работы через USB-порт](#)
 - [Переключение терминала в режим работы с USB порта на Ethernet](#)
 - [Настройка терминала на работу без кассы](#)
 - [Создание нового кассира](#)
 - [Вход меню Администратора](#)
 - [Загрузка модуля CTLS на терминал Verifone](#)
 - [Изменить идентификатор терминала](#)
 - [Настройка Wi-Fi](#)
- [Руководство пользователя](#)
 - [Подключение терминала к сети Ethernet](#)
 - [Нахождение IP-адреса терминала в сети](#)
 - [Автоматический поиск терминала \(только для IOS\)](#)
 - [Обновление конфигурации](#)
 - * [Способ №1](#)
 - * [Способ №2](#)
 - [Отправка логов на сервер](#)
 - * [Способ №1](#)
 - * [Способ №2](#)
- [Руководство пользователя Verifone Vx680](#)
 - [Включение](#)
 - [Выключение](#)
 - [Вход в меню](#)
 - [Оплата](#)
 - [Отмена платежа](#)
 - [Сверка итогов](#)
 - [Обновление конфигурации](#)
 - * [Способ №1](#)
 - * [Способ №2](#)
 - [Отправка логов на сервер](#)
 - * [Способ №1](#)
 - * [Способ №2](#)
- [FAQ](#)

- При включении терминала постоянно просит ввести сумму.
- Не получается провести сверку итогов: 3 + Enter завершается с ошибкой, 4+Enter и через программу mpos-mtips-advт не получается выполнить
- Поиск IP-адреса Verifone в сети
- Восстановление терминала после ошибки -3158
- Первичная установка терминала

Руководство администратора

Перевод терминала Verifone в режим работы через USB-порт

- 1) Войти в меню администратора, см. ниже.
- 2) Установить следующие параметры:
 - Параметры / Редактирование / SmartSale / Связь с банком / Через кассу
 - Параметры / Редактирование / SmartSale / Порт терминала / USB
 - Параметры / Редактирование / SmartSale / SSL без ККМ / Да

Переключение терминала в режим работы с USB порта на Ethernet

- 1) Войти в меню администратора, см. ниже.
- 2) Установить следующие параметры:
 - Параметры / Редактирование / SmartSale / Связь с банком / Через терминал
 - Параметры / Редактирование / SmartSale / Порт терминала / Ethernet

Настройка терминала на работу без кассы

- 1) Войти в меню администратора, см. ниже.
- 2) Установить следующие параметры:
 - Параметры / Редактирование / SmartSale / Режим ККМ / Касса или Терминал (по умолчанию - Касса)
- 3) После установки параметра терминал перезагрузится и в меню появится иконка для инициации платежа.

Note:

В этом режиме терминала Verifone печать слипов (чеков платежной системы) невозможна, т.к. принтер отсутствует.

Создание нового кассира

- 1) Войти в меню администратора, см. ниже.
- 2) Установить следующие параметры:
 - Параметры / Управление доступом / Управление правами / Новый оператор - создать нового оператора в группу “Кассиры” и установить пароль (на этапе “Введите карту доступа” нажать Enter).
 - Параметры / Управление доступом / Авто логин - выбрать нужного оператора (в случае необходимости).

Вход меню Администратора

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем нажать ENTER.
- 2) Ввести пароль 1234 и нажать ENTER.
- 3) Выбрать пункт меню “Блокировка терм.” и нажать ENTER.
- 4) Выбрать пункт “Администратор”, ввести пароль Администратора, затем нажать ENTER.

Загрузка модуля CTLS на терминал Verifone

- 1) Скачать USB Driver, POS Loader и OS_RemoteUpdatePkg.zip .
- 2) Установить USB Driver и POS Loader.
- 3) Запустить POS Loader, нажать Терминал, выбрать Verifone.
- 4) Подключить терминал к USB и к Ethernet-сети.
- 5) В POS Loader нажать “Файл” > “COM порт”, затем выбрать порт.
- 6) На терминале зажать клавиши 7 и ENTER одновременно.
- 7) Ввести пароль для входа в инженерное меню.
- 8) Выбрать Download (3) – Enter – 166831 – Single-app (1) – Partial dnld (2) – USB Dev (7).

Warning:

Пункт Full download не должен быть выбран.

- 9) Выбрать Загрузить CTLS, выбрать Нет. Дождаться завершения загрузки на терминале.

Warning:

Обязательно выбирать в диалоге HET.

- 10) Провести тестовую оплату с помощью бесконтактной карты (не Apple/Google Pay).

Изменить идентификатор терминала

1. Выбрать пункт меню Параметры.
2. Выбрать Редактирование.
3. Выбрать Настройка агента.
4. Для входа в меню Параметры приложения, нажать 1.
5. Стереть неправильный номер терминала и ввести новый. Правильный номер терминала вводится исходя из его серийного номера. Серийный номер находится на нижней части терминала. Идентификатор терминала должен состоять из восьми цифр.

Note:

Для Verifone не вводить первую цифру и знак тире. Например, если серийный номер терминала 123-456-789, то нужно ввести 23456789.

Настройка Wi-Fi

Руководство пользователя

Подключение терминала к сети Ethernet

- 1) Подключить кабель к разъему с надписью ЕТН.
- 2) Дождаться смены цвета иконки ЕТН с красной на зеленую в статусной строке экрана.
- 3) На роутере найти IP-адрес терминала по его MAC-адресу и закрепить его на DHCP-сервере роутера за терминалом. MAC-адрес терминала написан на желтой этикетке на нижней панели терминала.
- 4) При включении терминал должен успешно установить соединение с сервером.

Нахождение IP-адреса терминала в сети

Поиск IP-адреса Pax/Verifone в сети производится самостоятельно. Указание диапазона не обязательно.

Ограничения:

- Маска сети не должна выдавать больше 500 IP адресов.
- При нахождении больше одного терминала отобразится ошибка.

Note:

В случае присутствия больше одного терминала в сети, может быть указан его идентификатор (например, terminal_id=12345678) после чего терминал может быть проверен.

Правила для заполнения поля address:

- В случае присутствия в сети одного терминала и количество IP-адресов меньше 500, нет необходимости вводить данные.
- В случае присутствия в сети больше одного терминала и количество IP-адресов меньше 500, необходимо ввести идентификатор терминала (например, terminal_id=12345678).
- В случае присутствия в сети одного терминала и количество IP-адресов больше 500 IP-адресов, необходимо указать диапазон поиска IP-range (например, range=192.168.0.1-192.168.0.254).
- В случае присутствия в сети больше одного терминала и количество IP-адресов больше 500, необходимо ввести идентификатор терминала или диапазон поиска IP-range на выбор.

Note:

Установка программы для поиска терминала на [iOS³⁸](#) и [Android³⁹](#).

Автоматический поиск терминала (только для IOS)

- В настройках терминала прописать диапазон, в котором будет находится терминал (например, range=10.45.69.1-10.45.69.254:27015).

Обновление конфигурации

Способ №1

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем зажать клавиши 3 и ENTER одновременно.
 - 2) Ввести пароль 1234 и нажать ENTER.
 - 3) После отображения сообщения Журнал содержит операции, возможны финансовые потери. Продолжить? нажать ENTER.
- В случае подключения терминала по USB, после отображения сообщения Загрузка параметров будет отложена нажать ENTER. Затем необходимо провести оплату, во время которой будет произведено обновление конфигурации.
 - В случае подключения терминала по Ethernet, подключение к сети и загрузка программ пройдет автоматически.

³⁸ <https://mpos.pne.io/scanner/ios/install.html>

³⁹ <https://mpos.pne.io/verifone/scanner/scanner.apk>

Способ №2

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем нажать ENTER.
- 2) Ввести пароль 1234 и нажать ENTER.
- 3) Выбрать пункт Параметры, нажав кнопку 6, затем нажать ENTER.
- 4) Нажимать на ENTER (курсор на пункте ЗАГРУЗКА ПАРАМ.).
- 5) После отображения сообщения Журнал содержит операции, возможны финансовые потери. Продолжить? нажать ENTER.
 - В случае подключения терминала по USB, после отображения сообщения Загрузка параметров будет отложена нажать ENTER.
 - В случае подключения терминала по Ethernet, подключение к сети и загрузка программ пройдет автоматически.

Отправка логов на сервер

Способ №1

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем зажать клавиши 2 и ENTER одновременно.
- 2) Ожидать отправки логов на сервер.

Способ №2

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем нажать ENTER.
- 2) Ввести пароль 1234 и нажать ENTER.
- 3) Выбрать пункт Параметры, нажав кнопку 6, затем нажать ENTER.
- 4) Выбрать пункт меню Логирование.
- 5) Выбрать пункт Выгрузка лога.
- 6) В случае подключения терминала по USB, после отображения сообщения Выгрузка логов будет отложена нажать ENTER. Затем необходимо произвести оплату, во время которой будет произведено обновление конфигурации.
 - В случае подключения терминала по Ethernet, подключение к сети и загрузка программ пройдет автоматически.

Руководство пользователя Verifone Vx680

Руководство по установке может быть найдено по [ссылке](#)⁴⁰.

Включение

Нажать ENTER (зеленая кнопка). Если не включается, то зарядить.

Выключение

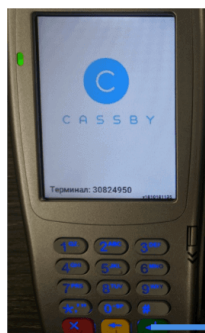
Нажать на CANCEL (красная кнопка) и удерживать в течение 10 секунд.

Вход в меню

Нажать клавиши 5 и ENTER (сначала 5, потом ENTER).

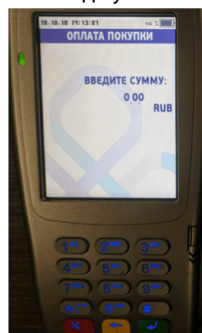
Оплата

Терминал после включения



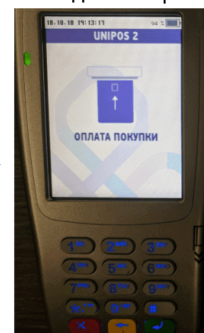
Нажать ENTER

Ввод суммы



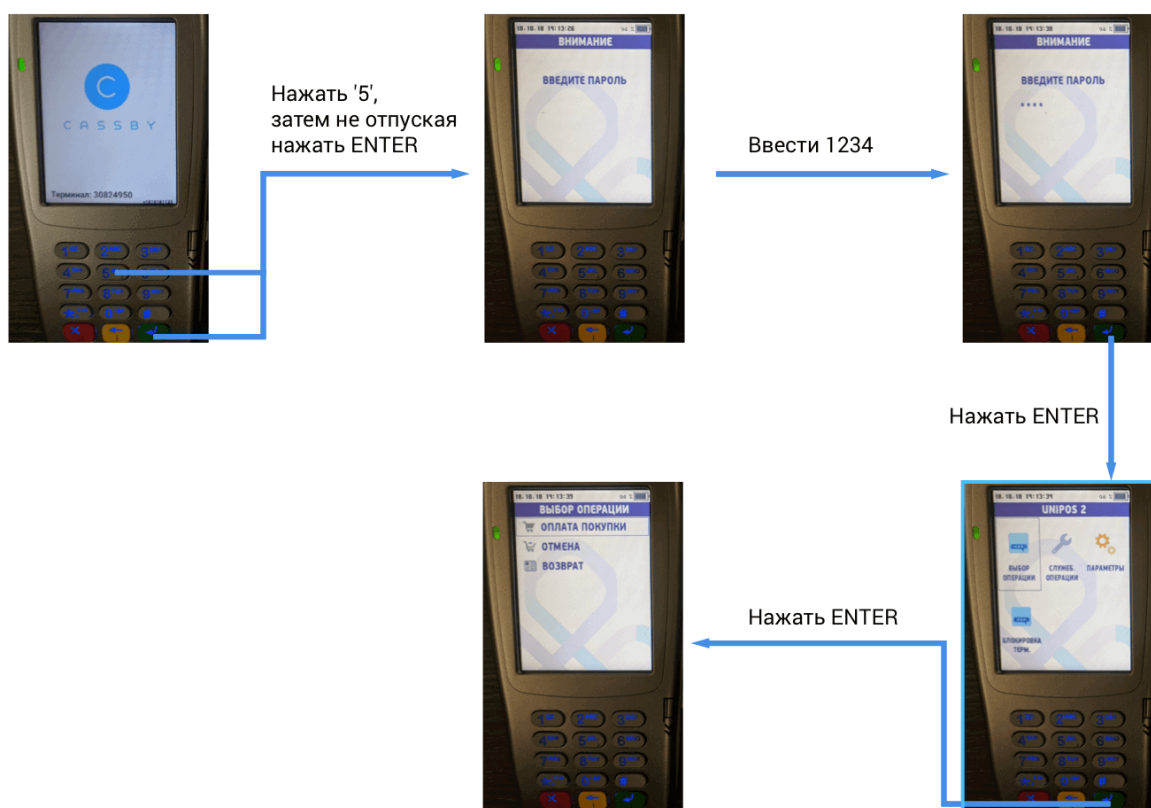
Ввести сумму
в копейках и нажать
ENTER

Ожидание карты

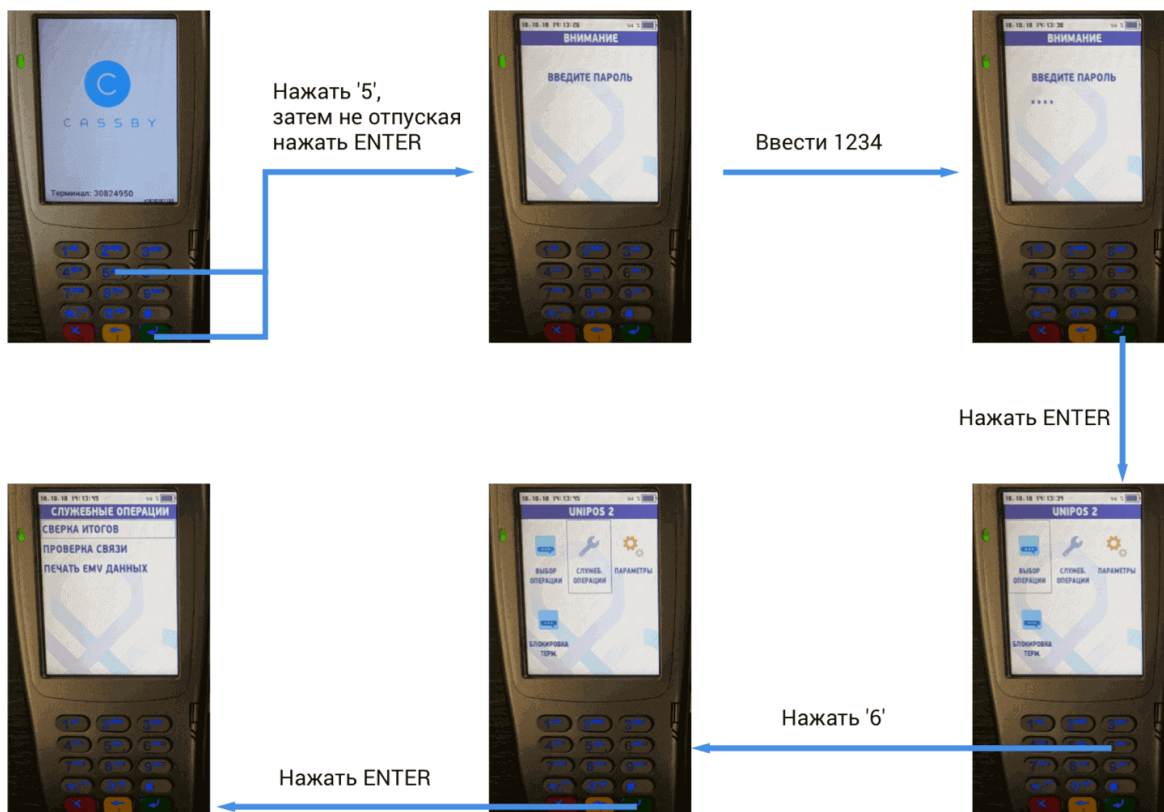


Отмена платежа

⁴⁰ <https://www.pos-shop.ru/upload/iblock/fac/verifone-vx680.pdf>



Сверка итогов



Обновление конфигурации

Способ №1

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем зажать клавиши 3 и ENTER одновременно.
- 2) Ввести пароль 1234 и нажать ENTER.
- 5) После отображения сообщения Журнал содержит операции, возможны финансовые потери. Продолжить? нажать ENTER.

Способ №2

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем нажать ENTER.
- 2) Ввести пароль 1234 и нажать ENTER.
- 3) Выбрать пункт Параметры, нажав кнопку 6, затем нажать ENTER.
- 4) Нажать на ENTER (курсор на пункте ЗАГРУЗКА ПАРАМ.).
- 5) После отображения сообщения Журнал содержит операции, возможны финансовые потери. Продолжить? нажать ENTER.

Отправка логов на сервер

Способ №1

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем зажать клавиши 2 и ENTER одновременно.
- 2) Ожидать отправки логов на сервер.

Способ №2

- 1) Дождаться логотипа или надписи Ожидание команды от ККМ, затем нажать ENTER.
- 2) Ввести пароль 1234 и нажать ENTER.
- 3) Выбрать пункт Параметры, нажав кнопку 6, затем нажать ENTER.
- 4) Выбрать пункт меню Логирование.
- 5) Выбрать пункт Выгрузка лога.

FAQ

При включении терминала постоянно просит ввести сумму.

На терминале кто-то активировал режим работы с терминалов для ввода суммы. Зайти в меню настройки SmartSale и включите режим работы терминала только с кассой.

Note:

Нажать 5 + ENTER (сначала 5, потом ENTER).

Не получается провести сверку итогов: 3 + Enter завершается с ошибкой, 4+Enter и через программу mpos-mtips-advt не получается выполнить

Необходимо проверить на терминале в настройках, правильно ли введен 8-ми значный номер терминала (первая 9-ка и тире отбрасываются: s/n на корпусе - 902-692-123, то номер в настройках - 02692123).

Смена номера терминала описана в разделе “Руководство администратора” -> “Изменить идентификатор терминала”.

Поиск IP-адреса Verifone в сети

Установить [программу поиска IP-адреса](#)⁴¹.

Восстановление терминала после ошибки -3158

Ошибка связана с повреждением базы данных, которая находится в файле UNIPOS.DBI. Для ее исправления необходимо обнулить файл UNIPOS.DBI.

Шаги для восстановления:

- 1) Включите терминал. Дождитесь, пока появится сообщение об ошибке.
- 2) Подключите mini USB кабель к компьютеру и терминалу.
- 3) Подключаем mini USB к терминалу в блок расширения, где подключен кабель ethernet к терминалу. Компьютер должен определить терминал как COM-порт. Если этого не произошло, то установите .
- 4) Затем необходимо выяснить, к какому COM-порту подключен терминал. Для этого на иконке Компьютер нажав ПКМ и выбрать пункт Управление. Затем перейдите в раздел Диспетчер устройств. Раскройте в дереве устройств узел Порты (COM и LPT). Выяснить порт, к которому подключен Verifone.

Перевод терминала в инженерный режим.

На терминале:

- 1) Нажать на терминале клавиши 7 и ENTER одновременно.
- 2) Ввести пароль 166831. Дождаться перехода терминала в специальный режим VERIX TERMINAL MGR.
- 3) Выбрать пункт Download, нажав клавишу 3, затем нажать ENTER. При этом будет выбрана Group_ID: 1.
- 4) Ввести еще раз пароль 166831, затем нажать ENTER.
- 5) Выбрать пункт Single-app, нажав клавишу 1.

⁴¹ <https://dengisend.ru/demo-ios/SocketScanner/install.html>

- 6) Выбрать пункт Partial dnld, нажав клавишу 2.
- 7) Выбрать пункт USB Dev, нажав клавишу 7.
- 8) Терминал отобразит надпись WAITING FOR DOWNLOAD.

Запуск программы на компьютере

На компьютере:

- 1) Скачать файл .
- 2) Открыть командную строку с помощью cmd.exe.
- 3) Запустить ddl.exe в окне.

```
c:\Users\Download\ddl.exe -p <port_number> -r UNIP0S.DBI
```

Где <port_number> - это номер COM-порта Verifone.

c:\Users\Download- куда была скачана программа ddl.exe

- 4) Перезапустить терминал.
- 5) Нажать ENTER. (Пункт меню 1> Restart).

После этого терминал перегрузится и начнет работать в штатном режиме.

Первичная установка терминала

Если при включении Verifone показывает следующую надпись:

```
DOWNLOAD NEEDED GO FILE NOT FOUND
```

Видео процесса установки

С компьютера:

С терминала:

Шаг 1: USB кабель

- 1) Подключить mini USB кабель к компьютеру.
- 2) Подключить mini USB к терминалу в блок расширения, где подключен кабель ethernet к терминалу.
- 3) Компьютер должен определить терминал как COM-порт. Если этого не произошло, то установить [драйвер⁴²](http://mpos.pne.io/verifone/VerifoneUnifiedDriverInstaller-5.0.4.0-B10.zip).

⁴² <http://mpos.pne.io/verifone/VerifoneUnifiedDriverInstaller-5.0.4.0-B10.zip>

- 4) Затем необходимо выяснить, к какому COM-порту подключен терминал. Для этого на иконке Компьютер нажав ПКМ и выбрать пункт Управление. Затем перейдите в раздел Диспетчер устройств. Раскройте в дереве устройств узел Порты (COM и LPT). Выяснить порт, к которому подключен Verifone.

Шаг 2: Запуск программы на компьютере

Установка программы UNIPOS на терминал с помощью компьютера.

- 1) Создать каталог c:\verifone.
- 2) Скачать файлы [ddl.exe](#)⁴³, [OW_BASE.zip](#)⁴⁴ и [Parameters.zip](#)⁴⁵, сохранить их в каталог c:\verifone.
- 3) Открыть командную строку с помощью cmd.exe и перейти в каталог c:\verifone

```
cd c:\verifone
```

- 4) Запустить следующую программу с параметрами:

```
ddl.exe -p 3 -b115200 *UNZIP=OW_BASE_2.0.13.014_RU.zip -iOW_BASE_2.0.13.014_RU.zip  
→CRYPTOCTYPE=V
```

- 5) После окончания загрузки терминал перезагрузится сам.

Шаг 3: отмена загрузки ключей

- 1) После первой загрузки терминал покажет меню загрузки ключей.
- 2) Нажать CANCEL.
- 3) Терминал перезагрузится еще раз.

Шаг 4: Загрузка параметров

- 1) После загрузки отобразится сообщение Параметры не загружены. Перевести терминал в инженерный режим. Нажать на терминале клавиши 7 и ENTER одновременно.
- 2) Ввести пароль 166831. Дождаться перехода терминала в специальный режим VERIX TERMINAL MGR.
- 3) Выбрать пункт Download, нажав клавишу 3, затем нажать ENTER. При этом будет выбрана Group_ID: 1.
- 4) Ввести еще раз пароль 166831, затем нажать ENTER.
- 5) Выбрать пункт Single-app, нажав клавишу 1.
- 6) Выбрать пункт Partial dnld, нажав клавишу 2.

⁴³ <http://mpos.pne.io/verifone/ddl.exe>

⁴⁴ https://mpos.pne.io/verifone/OW_BASE_2.0.13.014_RU.zip

⁴⁵ https://mpos.pne.io/verifone/02692552_Parameters.zip

- 7) Выбрать пункт USB Dev, нажав клавишу 7.
- 8) Терминал отобразит надпись WAITING FOR DOWNLOAD.

Шаг 5: Загрузка параметров

- В командной строке в каталоге c:verifone выполнить следующую команду:

```
ddl.exe -p3 -b115200 *UNZIP2=02692552_Parameters.zip -i02692552_Parameters.zip
```

Шаг 6: Создание администратора

- 1) После перезагрузки терминал может отобразить сообщение Создание администратора при первой загрузке.
- 2) Для этого ввести имя пользователя ROOT.
- 3) На предложение ввести смарт-карту нажать CANCEL.
- 4) Пароль 1478.

Шаг 7: Настройка номера терминала

- 1) После загрузки нажать клавиши 3 и ENTER на терминале.
- 2) Ввести номер терминала 02692552.

8.8 mPOS Test

- [Сервер Sandbox](#)
- [Сервер Qa](#)

8.8.1 Сервер Sandbox

В зависимости от суммы он ведет себя по-разному:

1.00 - обычный платеж без ошибок в сети. Результат APPROVED
другие суммы - Результат: DECLINED

8.8.2 Сервер Qa

Специальный сервер для тестирования различных ошибок связи:

<https://paynet-qa.clubber.me/paynet>

Сервер принимает любые значения для endpoint, merchant login и merchant control key.

В зависимости от суммы он ведет себя по-разному:

- 1.00 - обычный платеж без ошибок в сети. Результат APPROVED
- 1.01 - не был получен идентификатор платежа в ответе на sale. Результат: APPROVED
- 5.00 - timeout (70 секунд) на первый status. Результат: APPROVED
- 5.01 - timeout (70 секунд) на sale. Результат: APPROVED
- 5.02 - timeout (70 секунд) на advice. Результат: APPROVED
- 5.03 - timeout (70 секунд) на status после advice. Результат: APPROVED
- 6.00 - HTTP Status 503 на sale. Результат: APPROVED
- 6.01 - HTTP Status 503 на первый status. Результат: APPROVED
- 6.02 - HTTP Status 503 на advice. Результат: APPROVED
- 6.03 - HTTP Status 503 на status после advice. Результат: APPROVED

Все платежи должны в конце выдать статус APPROVED.

COMMON UTILITIES

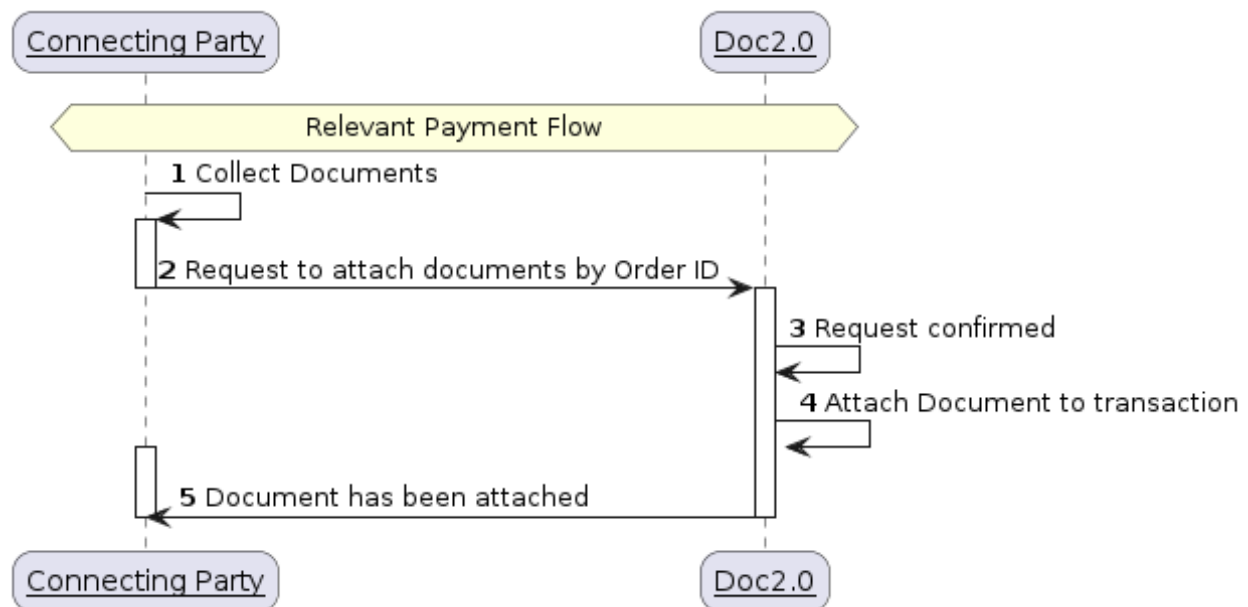
9.1 Add Document

- [Introduction](#)
- [Add Document Flow](#)

9.1.1 Introduction

Add Document allow to send request with document to attach it to the specific transaction. It is possible to attach any additional document to the existing transaction in the Payment Gateway. The attached document can be later downloaded from order details screen. This page describes how to attach the document to transaction by API.

9.1.2 Add Document Flow



(2) Each transaction in the System has assigned Order ID. Connecting Party makes upload document request to attach it to the specified transaction by Order ID, using </api/v2/documents/add-document>.

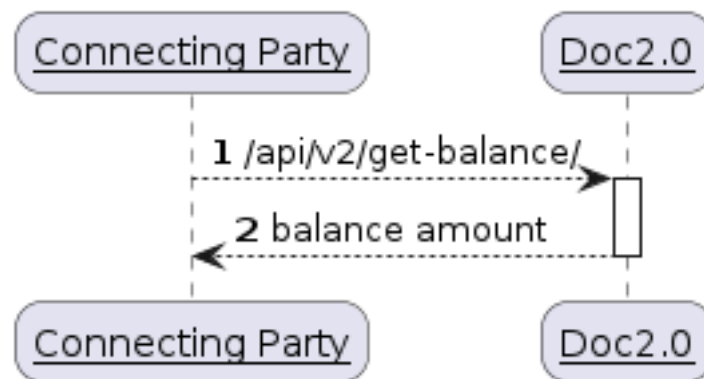
9.2 Account balance

- [Introduction](#)
- [Get Balance Flow](#)
- [Get Balance Merchant Flow](#)
- [Get Balance Manager Flow](#)
- [Get Balance Adjustments Flow](#)

9.2.1 Introduction

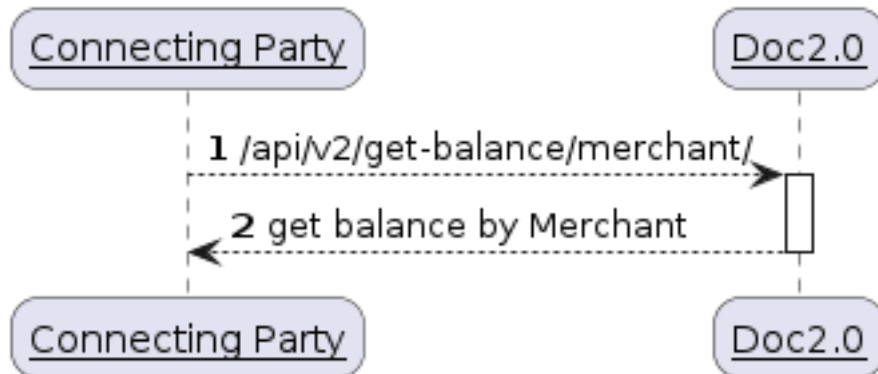
Get balance APIs help Connecting Party to check current balances configured in Payment Gateway. Balances can be requested by Endpoint (for specific provider), by Merchant (specific balance or all balances) and by Manager (specific balance or all balances). Additionally, users with Manager account are able to upload balance adjustments by API.

9.2.2 Get Balance Flow



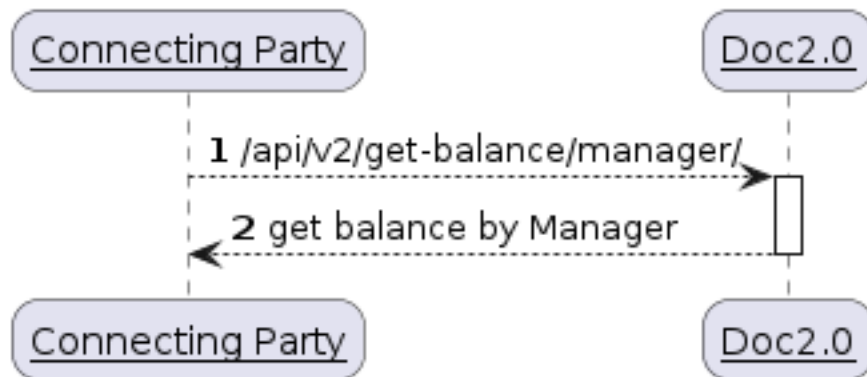
(1) To implement get balance by Endpoint request see </api/v2/get-balance/>.

9.2.3 Get Balance Merchant Flow



(1) To implement get balance by Merchant request see </api/v2/get-balance/merchant/>.

9.2.4 Get Balance Manager Flow



(1) To implement get balance by Manager request see </api/v2/get-balance/manager/>.

9.2.5 Get Balance Adjustments Flow

(1) To implement Get Balance Adjustments request see [Get Balance Adjustments](#).



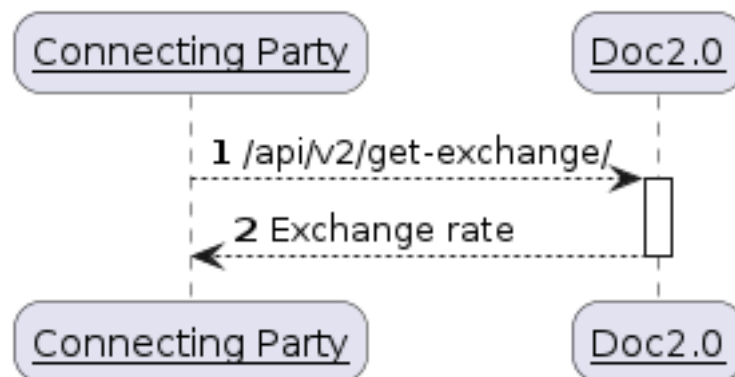
9.3 Get Exchange Rate

- [Introduction](#)
- [Get Exchange Rate Flow](#)

9.3.1 Introduction

Get exchange rate helps Connecting Party to get current conversion rate between two currencies.

9.3.2 Get Exchange Rate Flow



(1) To implement get exchange rate request see [/api/v2/get-exchange/](#).

9.4 PAN Eligibility

- [Introduction](#)
- [PAN Eligibility Direct Flow](#)
- [PAN Eligibility Form Flow](#)
- [PAN Eligibility Callback](#)
 - [Callback Parameters](#)
 - [Callback Macros](#)

9.4.1 Introduction

PAN Eligibility services allow Connecting Party to detect whether card is eligible for Money Transfer (either as sending or as receiving card) along with providing extended information about card issuer and card product before making a financial transaction request.

9.4.2 PAN Eligibility Direct Flow

(2,3,4) To implement PAN eligibility direct flow request see </api/v2/pan-eligibility/>. PAN eligibility information can be requested for Payer card (also referred as Sender), Recipient card (also referred as Receiver) or both cards at once.

(5) To implement callback with final status handling see [Connecting Party Callback](#).

(7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

9.4.3 PAN Eligibility Form Flow

(2,3,4) To implement PAN eligibility form flow request see </api/v2/pan-eligibility-form/>. PAN eligibility information can be requested for Payer card (also referred as Sender), Recipient card (also referred as Receiver) or both cards at once.

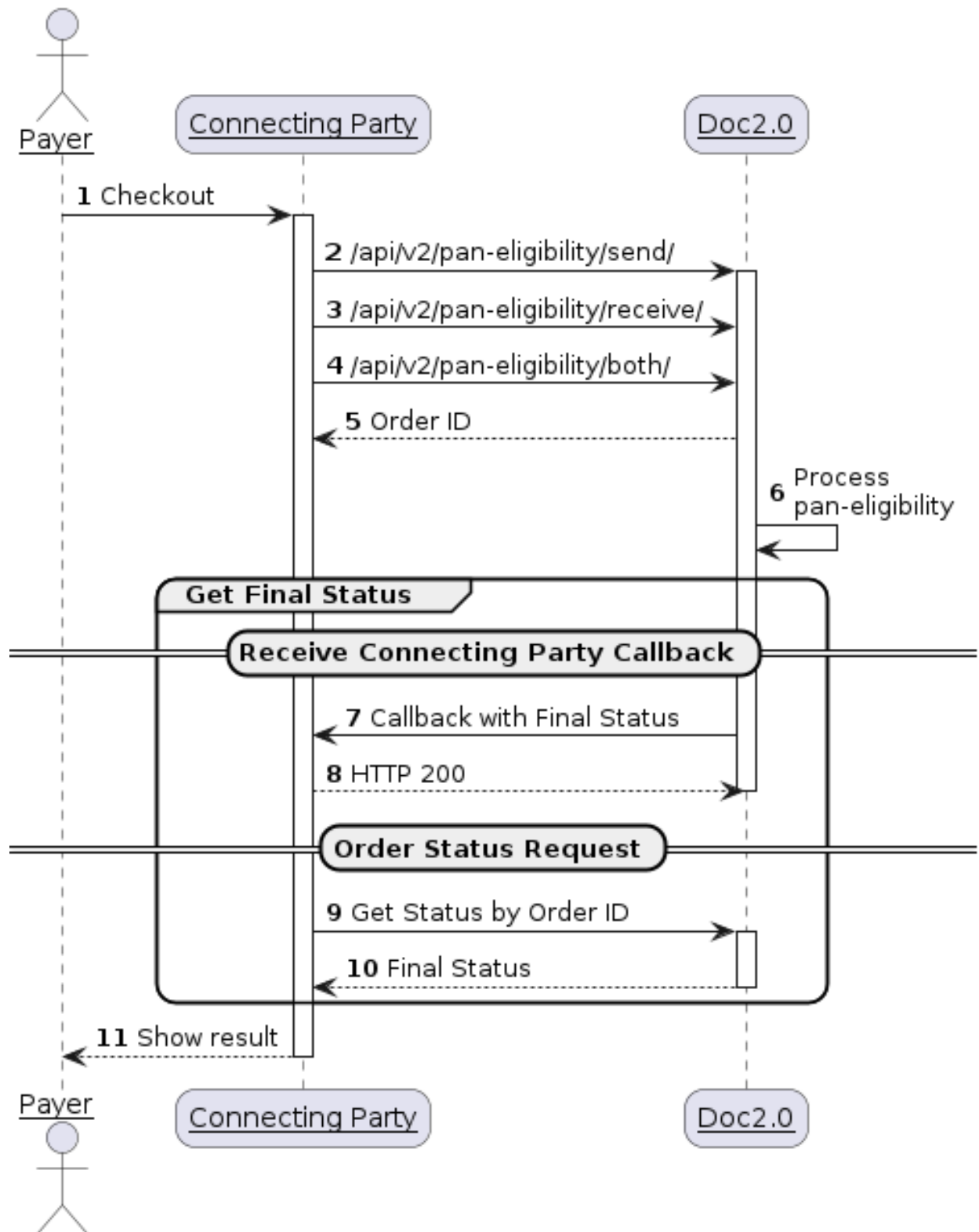
(5) To implement callback with final status handling see [Connecting Party Callback](#).

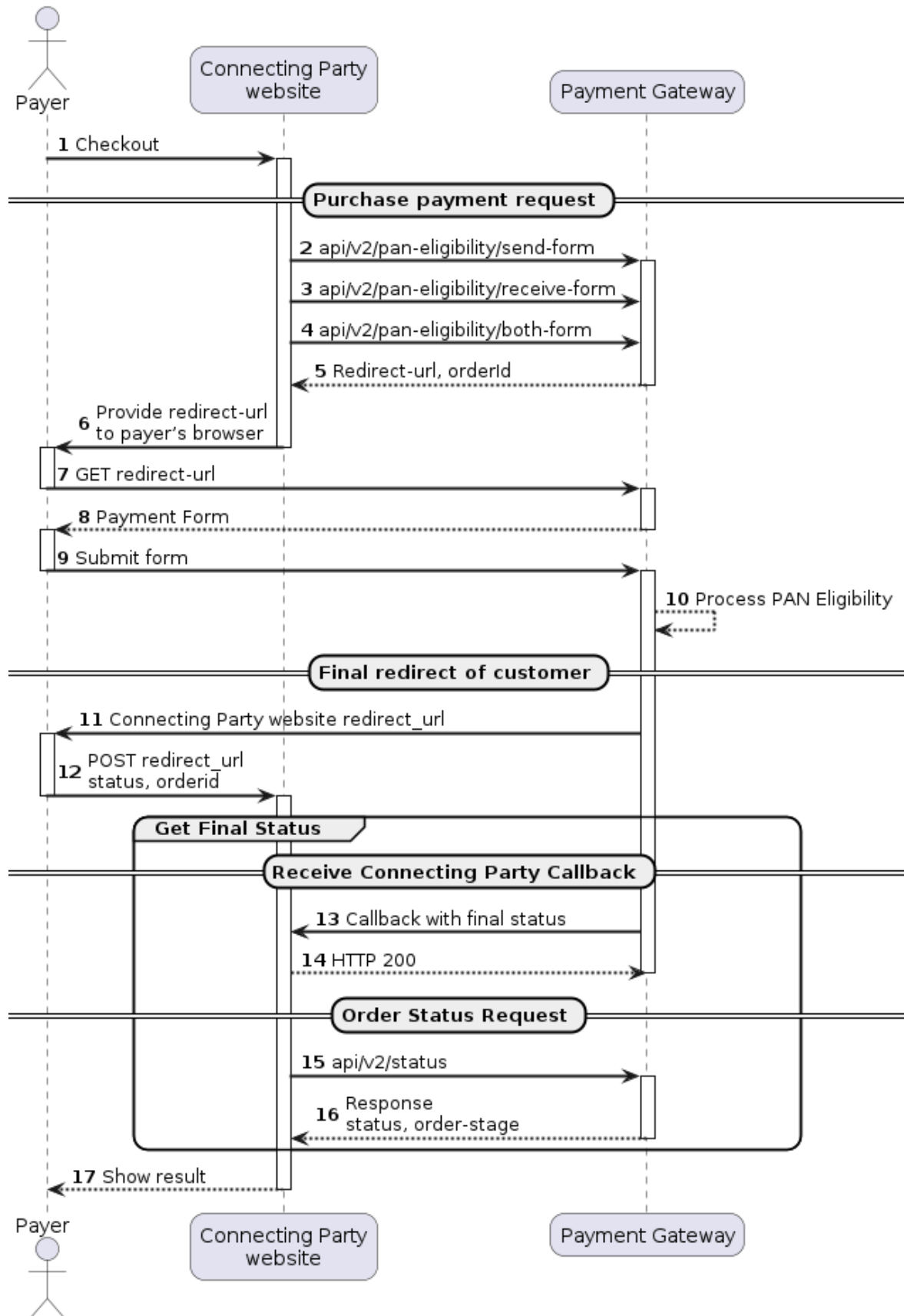
(7) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

9.4.4 PAN Eligibility Callback

Callback Parameters

Additional parameters are sent in callbacks for PAN Eligibility transactions. For more information see [Connecting Party Callback Parameters](#).





Callback Parameter	Description
status	See Status List for details.
serial-number	Serial number of the request.
client_orderid	Connecting Party order identifier.
paynet-order-id	Order id set by Doc2.0.
processor-tx-id	Transaction id set by processor.
type	Transaction type, sale reversal chargeback
sending-card-type	The sending card type (VISA, MASTERCARD).
sending-cardholder	Cardholder of the sending card.
sending-bin	The sending card Bank identification number.
sending-last-four-digits	The sending card last 4 digits.
sending-eligible	Whether the sending card is eligible for sending funds. Possible values: true/false/unknown.
sending-ica	The sending card ICA.
sending-bank-name	The sending card Bank name.
sending-currency-code	The sending card currency code.
sending-country-code	The sending card country code.
sending-acceptance-brand-code	The sending card Acceptance brand code.
sending-acceptance-brand-name	The sending card Acceptance brand name.
sending-product-brand-code	The sending card Product brand code.
sending-product-brand-name	The sending card Product brand name.
receiving-card-type	The receiving card type (VISA, MASTERCARD).
receiving-bin	The receiving card Bank identification number.
receiving-last-four-digits	The receiving card last 4 digits.
receiving-eligible	Whether the receiving card is eligible for sending funds. Possible values: true/false/unknown.
receiving-ica	The receiving card ICA.
receiving-bank-name	The receiving card Bank name.
receiving-currency-code	The receiving card currency code.
receiving-country-code	The receiving card country code.
receiving-acceptance-brand-code	The receiving card Acceptance brand code.
receiving-acceptance-brand-name	The receiving card Acceptance brand name.
receiving-product-brand-code	The receiving card Product brand code.
receiving-product-brand-name	The receiving card Product brand name.
error-code	Error Code.
error-message	Error Message.
control	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Callback parameter: status. 2. Callback parameter: orderid. 3. Callback parameter: merchant_order. 4. merchant_control (Control key assigned to Merchant account in the Doc2.0 Payment Gateway).

Callback Macros

Simple callback URL contains all parameters listed in [Callback Parameters](#). Customizable URL allows Connecting Party to define own parameter names, whereas the actual parameters values are defined by use of macros with the following format `${parameter_name}`. Available macros are listed in [Callback Macros](#).

Field Name Macro	Field Value Macro	Description
<code>\${status}</code>	n/a	Transaction status, approved declined processing.
<code>\${serial-number}</code>	n/a	Unique number assigned by Doc2.0 server to particular request from the Connecting Party.
<code>\${client-order-id}</code>	n/a	Unique order identifier assigned by Connecting Party.
<code>\${paynet-order-id}</code>	n/a	Doc2.0-generated order id.
<code>\${processor-tx-id}</code>	n/a	Processor-generated transaction id.
<code>\${type}</code>	n/a	Transaction type, ex. sale, return, chargeback, etc.
<code>\${sending-card-type}</code>	n/a	The sending card type (VISA, MASTERCARD).
<code>\${sending-cardholder}</code>	n/a	Cardholder of the sending card.
<code>\${sending-bin}</code>	n/a	The sending card Bank identification number.
<code>\${sending-last-four-digits}</code>	n/a	The sending card last 4 digits.
<code>\${sending-eligible}</code>	n/a	Whether the sending card is eligible for sending funds. Possible values: true/false/unknown.
<code>\${sending-ica}</code>	n/a	The sending card ICA.
<code>\${sending-bank-name}</code>	n/a	The sending card Bank name.
<code>\${sending-currency-code}</code>	n/a	The sending card currency code.
<code>\${sending-country-code}</code>	n/a	The sending card country code.
<code>\${sending-acceptance-brand-code}</code>	n/a	The sending card Acceptance brand code.
<code>\${sending-acceptance-brand-name}</code>	n/a	The sending card Acceptance brand name.
<code>\${sending-product-brand-code}</code>	n/a	The sending card Product brand code.
<code>\${sending-product-brand-name}</code>	n/a	The sending card Product brand name.
<code>\${receiving-card-type}</code>	n/a	The receiving card type (VISA, MASTERCARD).
<code>\${receiving-bin}</code>	n/a	The receiving card Bank identification number.
<code>\${receiving-last-four-digits}</code>	n/a	The receiving card last 4 digits.
<code>\${receiving-eligible}</code>	n/a	Whether the receiving card is eligible for sending funds. Possible values: true/false/unknown.
<code>\${receiving-ica}</code>	n/a	The receiving card ICA.
<code>\${receiving-bank-name}</code>	n/a	The receiving card Bank name.

continues on next page

Table 2 – continued from previous page

Field Name Macro	Field Value Macro	Description
<code>\${receiving-currency-code}</code>	n/a	The receiving card currency code.
<code>\${receiving-country-code}</code>	n/a	The receiving card country code.
<code>\${receiving-acceptance-brand-code}</code>	n/a	The receiving card Acceptance brand code.
<code>\${receiving-acceptance-brand-name}</code>	n/a	The receiving card Acceptance brand name.
<code>\${receiving-product-brand-code}</code>	n/a	The receiving card Product brand code.
<code>\${receiving-product-brand-name}</code>	n/a	The receiving card Product brand name.
<code>\${error-code}</code>	n/a	Error Code.
<code>\${error-message}</code>	n/a	Error Message.
<code>\${control}</code>	n/a	Checksum generated by SHA-1 . Control string is represented as concatenation of the following parameters: 1. Callback parameter: status. 2. Callback parameter: orderid. 3. Callback parameter: merchant_order. 4. merchant_control (Control key assigned to Merchant account in the Doc2.0 Payment Gateway).

9.5 Receipt and message templates

- [General Introduction](#)
- [Receipt Template](#)
 - [Receipt template macros](#)
 - [Receipt template example](#)
- [Message template](#)
 - [Message template macros](#)
 - [Message template examples](#)

9.5.1 General Introduction

Both receipt and message templates must be sent to Doc2.0 support for installation. Doc2.0 supports receipt and message delivery to customer after each successful transaction. Receipts are accessed by unique URL described below. Messages can be sent by SMS or E-mail using Connecting Party message server.

9.5.2 Receipt Template

Receipts can be sent only in HTML format.
Receipt ID can be found in status response.

Integration	Production
https://sandbox.doc2.com/paynet/view-receipt/ENDPOINT-ID/RECEIPT-ID	https://gate.doc2.com/paynet/view-receipt/ENDPOINT-ID/RECEIPT-ID

Receipt template macros

Field Name Macro	Description
<code>#{PAYMENT_FORM_HEADER}</code>	Name on payment form in End Point
<code>#{DATE}</code>	Order date.
<code>#{DESCRIPTION}</code>	Brief order description.
<code>#{AMOUNT}</code>	Amount.
<code>#{CURRENCY}</code>	Currency the transaction is charged in (three-letter currency code).
<code>#{PAYNET_ORDER_ID}</code>	Doc2.0 order id.
<code>#{MERCHANT_ORDER_ID}</code>	Order ID assigned to this order by the Connecting Party.
<code>#{RRN}</code>	Bank Receiver Registration Number.
<code>#{AUTH_CODE}</code>	Authorization Code.
<code>#{AUTH_RESPONSE_CODE}</code>	A two- to six-digit alphanumeric code provided by the issuer that indicates the authorization response (approval or denial).
<code>#{STATUS}</code>	Order status.
<code>#{SKIN_VERSION}</code>	CSS skin version.
<code>#{BIN}</code>	Bank BIN of customer credit card number.
<code>#{LAST_FOUR_DIGITS}</code>	Last four digits of a card.
<code>#{CARD_TYPE}</code>	Card type.
<code>#{CARDHOLDER_NAME}</code>	Card holder name.
<code>#{DESTINATION_BIN}</code>	Bank BIN of destination credit card number.
<code>#{DESTINATION_LAST_FOUR_DIGITS}</code>	Last four digits of a destination card.
<code>#{DESTINATION_CARD_TYPE}</code>	Destination card type.
<code>#{PAYNET_PROCESSING_DATE}</code>	Doc2.0 processing date.
<code>#{ACQUIRER_PROCESSING_DATE}</code>	Acquirer bank processing date.
<code>#{READER_PROCESSING_DATE}</code>	Reader terminal processing date.
<code>#{TERMINAL_ID}</code>	Terminal ID.

continues on next page

Table 4 – continued from previous page

Field Name Macro	Description
\${EMV_ISSUER_RESPONSE}	A response code to an authorization request sent by the issuer.
\${EMV_TERMINAL_AID_9F06}	Application Identifier terminal. Identifies the application as described in ISO/IEC 7816-5.
\${EMV_TVR_95}	Terminal Verification Results. Status of the different functions as seen from the terminal.
\${EVM_CID_9F27}	Cryptogram Information Data. Indicates the type of cryptogram and the actions to be performed by the terminal.
\${EMV_APP_LABEL_50}	Application Label. Mnemonic associated with the AID according to ISO/IEC 7816-5.
\${EMV_CVR_9F34}	Cardholder Verification Method (CVM) Results. Indicates the results of the last CVM performed.
\${EMV_APP_CRYPTOGAM_9f26}	Application Cryptogram. Cryptogram returned by the ICC in response of the GENERATE AC command.
\${EMV_PAN_SEQUENCE_5f34}	Application Primary Account Number (PAN) Sequence Number. Identifies and differentiates cards with the same PAN.
\${PIN_OR_SIGNATURE}	Ask for PIN or just have a signature.
\${READER_ENTRY_MODE}	Client authentication configuration in terminal.
\${HAS_SIGNATURE}	Presence of a signature (TRUE or FALSE).
\${RECEIPT_ID}	Receipt ID.
\${RECEIPT_URL}	URI of the current receipt.
\${FUNCTIONS}	Library of functions for working with strings.

Receipt template example

```

<!DOCTYPE html>
<html>
  <body style="font-family:'tahoma';color:#1A407D;">
    <div style="font-family:'tahoma';max-width:640px;margin:auto;margin-top:20px;
    ↪text-align: center;background-color:#F5F6F6;padding:20px;border-top:6px solid
    ↪#1A407D;border-radius:42px;border-right:1px solid #E7E7E7;border-left:1px solid
    ↪#E7E7E7;border-bottom:1px solid #E7E7E7;direction:ltr;">
      <h1 style="font-weight:900;color:#7D1A40;">
        ${AMOUNT} ${CURRENCY} Paid
      </h1>
      <p style="font-family:'tahoma';color:#1A407D;font-weight:700;font-size:17px;
    ↪line-height:20px;">
        Hi ${CARDHOLDER_NAME},
        <br>
        Thank you for your payment through Doc2.0!
        <br>
        Here are your transaction details:
      </p>
      <p style="font-family:'tahoma';color:#1A407D;text-align: left;margin-top:70px;
    ↪margin-bottom:70px;">
        ${CARDHOLDER_NAME
  
```

(continues on next page)

(continued from previous page)

```

    <br>
    Transaction # : ${PAYNET_ORDER_ID}
    <br>
    Paid with ${CARD_TYPE} ending with ${LAST_FOUR_DIGITS}
    <br>
    ${DATE}
  </p>
  <p style="font-family:'tahoma';color:#1A407D;text-align: left;">
    The transaction will appear on your credit card statement as nymeria.
  </p>
  <div style="height: 2px;background-color:#E7E7E7;margin-top:32px;margin-
  ↪bottom:32px;"></div>
    <br>
    <a style="font-family:'tahoma';color:#1A407D;" href="https://site.company_
  ↪server/"><span>gate.doc2.com</span></a>
    <span style="font-family:'tahoma';color:#1A407D;margin-right:16px;margin-
  ↪left:16px;">Support@gate.doc2.com</span>
  </div>
</body>
</html>

```

9.5.3 Message template

Messages can be sent to customer in plain text, SMS or HTML format.

Message server must be specified on Project or Endpoint level.

Message template macros

Field Name Macro	Description
\${CARDHOLDER_VALUE}	Card printed name.
\${CARDHOLDER_EMAIL}	Card holder email sent in incoming API request.
\${CARDHOLDER_PHONE}	Card holder phone number sent in incoming API request.
\${CUSTOMER_FULL_NAME}	Customer full name sent by connecting party via input parameters.
\${CUSTOMER_FIRST_NAME}	Customer first name sent by connecting party via input parameters.
\${CUSTOMER_LAST_NAME}	Customer last name sent by connecting party via input parameters.
\${CUSTOMER_COUNTRY_NAME}	Customer country name sent in incoming API request.
\${CUSTOMER_COUNTRY_CODE}	Billing country of customer sent in incoming API request. For example, IL, RU, etc.
\${CUSTOMER_STATE_NAME}	Customer state name sent by connecting party via input parameters, if there are any in customer's country.

continues on next page

Table 5 – continued from previous page

Field Name Macro	Description
\${CUSTOMER_STATE_CODE}	Customer state code sent by connecting party via input parameters, if there are any in customer's country.
\${CUSTOMER_CITY}	Customer city name sent in incoming API request.
\${CUSTOMER_ZIP_CODE}	Customer's ZIP code.
\${CUSTOMER_ADDRESS1}	Customer address sent in incoming API request.
\${CUSTOMER_ADDRESS2}	Customer address sent in incoming API request, if the additional address was required.
\${AMOUNT}	Amount.
\${GATEDESCRIPTOR}	Gate descriptor, specified in payment gate.
\${ORDERDESCRIPTION}	Order description.
\${CURRENCY}	Currency the transaction is charged in (three-letter currency code).
\${CARD_NUMBER}	6 + 4 masked card number
\${BIN}	Bank BIN of customer credit card number.
\${LAST_FOUR_DIGITS}	Last four digits of a card.
\${CARD_TYPE}	Card type.
\${PAYNET_ORDER_ID}	Doc2.0 order id.
\${MERCHANT_ORDER_ID}	Order ID assigned to this order by the Connecting Party.
\${CREATED_DATE}	Transaction creation date in Doc2.0.
\${CREATED_YEAR}	Transaction creation year in Doc2.0
\${CREATED_MONTH}	Transaction creation month in Doc2.0
\${CREATED_DAY}	Transaction creation day in Doc2.0
\${MERCHANT}	End point display name.
\${MERCHANT_NAME}	Connecting Party display name.
\${ENDPOINT_ID}	End point ID.
\${PAYMENT_FORM_NAME}	Name on payment form in End Point.
\${RRN}	Bank Receiver Registration Number.
\${APPROVAL_CODE}	Verification code needed to authorize a payment
\${AUTH_RESPONSE_CODE}	A two- to six-digit alphanumeric code provided by the issuer that indicates the authorization response (approval or denial).
\${STATUS}	Status of the order.
\${PAYNET_PROCESSING_DATE}	Doc2.0 processing date (year, month, day and time in MSK timezone).
\${ACQUIRER_PROCESSING_DATE}	Acquirer bank processing date (year, month, day and time in MSK timezone).
\${READER_PROCESSING_DATE}	Reader terminal processing date (year, month, day and time in MSK timezone).
\${TERMINAL_ID}	Terminal ID.
\${EMV_ISSUER_RESPONSE}	A response code to an authorization request sent by the issuer.

continues on next page

Table 5 – continued from previous page

Field Name Macro	Description
\${EMV_TVR_95}	Terminal Verification Results. Status of the different functions as seen from the terminal.
\${EVM_CID_9F27}	Cryptogram Information Data. Indicates the type of cryptogram and the actions to be performed by the terminal.
\${EMV_APP_LABEL_50}	Application Label. Mnemonic associated with the AID according to ISO/IEC 7816-5.
\${EMV_CVR_9F34}	Cardholder Verification Method (CVM) Results. Indicates the results of the last CVM performed.
\${EMV_APP_CRYPTOGAM_9f26}	Application Cryptogram. Cryptogram returned by the ICC in response of the GENERATE AC command.
\${EMV_PAN_SEQUENCE_5f34}	Application Primary Account Number (PAN) Sequence Number. Identifies and differentiates cards with the same PAN.
\${PIN_OR_SIGNATURE}	Ask for PIN or just have a signature.
\${READER_ENTRY_MODE}	Client authentication configuration in terminal.
\${HAS_SIGNATURE}	Presence of a signature (TRUE or FALSE).
\${RECEIPT_ID}	Receipt ID.
\${RECEIPT_URI}	URN of the current receipt (/view-receipt/ENDPOINT-ID/RECEIPT-ID).
\${PURPOSE}	Any additional information sent by Connecting Party.
\${MERCHANT_DATA}	Any additional information for this transaction which may be useful in Connecting Party's external systems.
\${ENDPOINT_SEQUENCE_VALUE}	Used when there is a sequence of endpoints, increasing by one in numerical format.

Message template examples

Plain text and SMS template example:

```

${AMOUNT} ${CURRENCY} Paid
Hi ${CARDHOLDER_VALUE},
Thank you for your payment through Doc2.0!
Here are your transaction details:
${CARDHOLDER_VALUE}
Transaction # : ${PAYNET_ORDER_ID}
Paid with ${CARD_TYPE} ending with ${LAST_FOUR_DIGITS}
${CREATED_DATE}
The transaction will appear on your credit card statement as nymeria.

```

HTML template example:

```
<!DOCTYPE html>
<html>
  <body style="font-family:'tahoma';color:#1A407D;">
    <div style="font-family:'tahoma';max-width:640px;margin:auto;margin-top:20px;
    ↪text-align: center;background-color:#F5F6F6;padding:20px;border-top:6px solid
    ↪#1A407D;border-radius:42px;border-right:1px solid #E7E7E7;border-left:1px solid
    ↪#E7E7E7;border-bottom:1px solid #E7E7E7;direction:ltr;">
      <h1 style="font-weight:900;color:#7D1A40;">
        $!AMOUNT $!CURRENCY Paid
      </h1>
      <p style="font-family:'tahoma';color:#1A407D;font-weight:700;font-size:17px;
    ↪line-height:20px;">
        Hi $!CARDHOLDER_VALUE,
        <br>
        Thank you for your payment through Doc2.0!
        <br>
        Here are your transaction details:
      </p>
      <p style="font-family:'tahoma';color:#1A407D;text-align: left;margin-top:70px;
    ↪margin-bottom:70px;">
        $!CARDHOLDER_VALUE
        <br>
        Transaction # : $!PAYNET_ORDER_ID
        <br>
        Paid with $!CARD_TYPE ending with $!LAST_FOUR_DIGITS
        <br>
        $!CREATED_DATE
      </p>
      <p style="font-family:'tahoma';color:#1A407D;text-align: left;">
        The transaction will appear on your credit card statement as nymeria.
      </p>
      <div style="height: 2px;background-color:#E7E7E7;margin-top:32px;margin-
    ↪bottom:32px;"></div>
      <br>
      <a style="font-family:'tahoma';color:#1A407D;" href="https://site.company_
    ↪server/"><span>gate.doc2.com</span></a>
      <span style="font-family:'tahoma';color:#1A407D;margin-right:16px;margin-
    ↪left:16px;">Support@gate.doc2.com</span>
    </div>
    </body>
  </html>
```

9.6 Reports for Integration

- Remote transactions report
 - Overview
 - Remote Transactions Report Flow
- Custom reports

- [Overview](#)
- [Custom Report Flow](#)

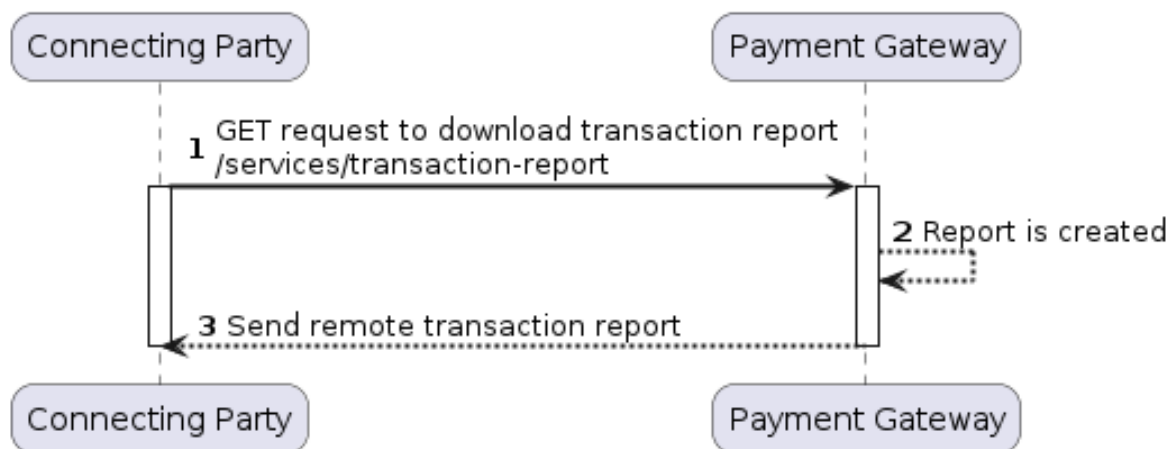
9.6.1 Remote transactions report

Overview

Remote transactions report provide an option to get complete information about transactions with their statuses for reconciliation, data integrity checks and further analysis.

Connecting Party is able to get a transaction report via Payment Gateway API.

Remote Transactions Report Flow



(1) To send GET request to download remote transaction report see [/services/transaction-report](#).

(2) Remote transaction report is generated based on applied filters.

(3) Prepared report is sent to Connecting Party.

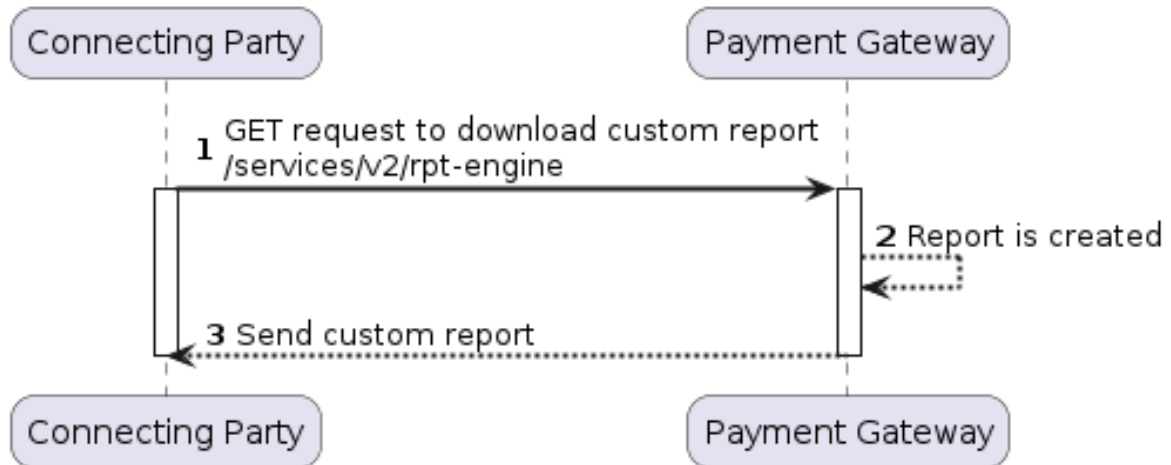
9.6.2 Custom reports

Overview

Custom reports provide an option to get customized information from Doc2.0 system to meet the specific business needs. Custom report content and format must be requested from Payment Gateway representatives.

Connecting Party is able to get a custom report via Payment Gateway API.

Custom Report Flow



(1) To send GET request to download custom report see </services/v2/rpt-engine>.

(2) Custom report is prepared based on applied filters.

(3) Prepared report is sent to Connecting Party.

9.7 Fiscalization service

- [Overview](#)
- [Fiscalization service parameter \(cheque\)](#)
- [Json string structure for cheque parameter](#)
- [Example of transaction request with fiscalization](#)

9.7.1 Overview

Fiscalization service is used for registration of financial cheques and invoices in Doc2.0 payment gateway and their further processing. Fiscalization is supported for both initial and recurring Sale and Preauth transactions, via Server-To-Server or Form integration. See the suitable integration scenario in respective Use-Cases or ask support manager for details. Transaction with cheque/invoice is processed by Doc2.0 Payment Gateway with additional requests for tax accounting.

9.7.2 Fiscalization service parameter (cheque)

In order to initiate a transaction with the use of fiscalization service, Connecting Party sends an HTTPS POST request for sale or preauth transaction with additional parameter.

Request Parameters	Type	Comment
cheque	Json string	Financial cheque or invoice. Mandatory for transaction request with Fiscalization

9.7.3 Json string structure for cheque parameter

A simple json string example for cheque parameter:

```
{
  "cheque_type" : 1,
  "tax_system" : 1,
  "positions" : [
    {
      "quantity" : 2,
      "price" : 322.94,
      "tax" : 4,
      "description" : "Product/Service 1"
    },
    {
      "quantity" : 1,
      "price" : 500,
      "tax" : 4,
      "description" : "Product/Service 2"
    }
  ]
}
```

Json string parameters:

Parameter	Type	Comment	Possible values
cheque_type	Decimal	Calculation type/sign	0 - income, 1 - expense, 2 - return of income, 3 - return of expense
tax_system	Decimal	Tax system	0 - General taxation system, 1 - Simplified income taxation system, 2 - Simplified income minus expense taxation system, 3 - Unified tax on imputed income, 4 - Unified agricultural tax, 5 - Patent system of taxation
positions	Array	Array of products/services	Array of products/services

continues on next page

Table 7 - continued from previous page

Parameter	Type	Comment	Possible values
quantity	Decimal	Quantity of product/service	
price	Decimal	Price per product/service considering discounts and mark-ups	
tax	Decimal	value-added tax	1 - 20% value-added tax, 2 - 10% value-added tax, 3 - 0% value-added tax, 4 - without value-added tax, 5 - 20/120 value-added tax, 6 - 10/110 value-added tax
description	String(128)	Description of product/service	

Note: Please take into consideration that the total price of all products/services which are provided in positions array should be equal to amount parameter value of sale request.

9.7.4 Example of transaction request with fiscalization

This example of Server-To-Server Sale request body with fiscalization displays how to correctly include the cheque parameter. Response example is provided in relevant API command documentation.

```

client_orderid=902B4FF5
&order_desc=Test Order Description
&first_name=John
&last_name=Smith
&ssn=1267
&birthday=19820115
&address1=100 Main st
&city=Seattle
&state=WA
&zip_code=98102
&country=US
&phone=%2B12063582043
&cell_phone=%2B19023384543
&amount=700.00
&email=john.smith@gmail.com
&currency=USD
&ipaddress=65.153.12.232
&site_url=www.google.com
&credit_card_number=4538977399606732
&card_printed_name=CARD HOLDER
&expire_month=12
&expire_year=2099
&cvv2=123
&purpose=user_account1
&redirect_url=https://doc2.codetime.net/doc/dummy.htm

```

(continues on next page)

(continued from previous page)

```

&server_callback_url=https://httpstat.us/200
&merchant_data=VIP customer
&cheque={ "cheque_type" : 1, "tax_system" : 1, "positions" : [ { "quantity" : 2,
↪ "price" : 100.00, "tax" : 4, "description" : "Product/Service 1" }, { "quantity" : 1,
↪ "price" : 500, "tax" : 4, "description" : "Product/Service 2" } ] }
&control=768eb8162fc361a3e14150ec46e9a6dd8fbfa483

```

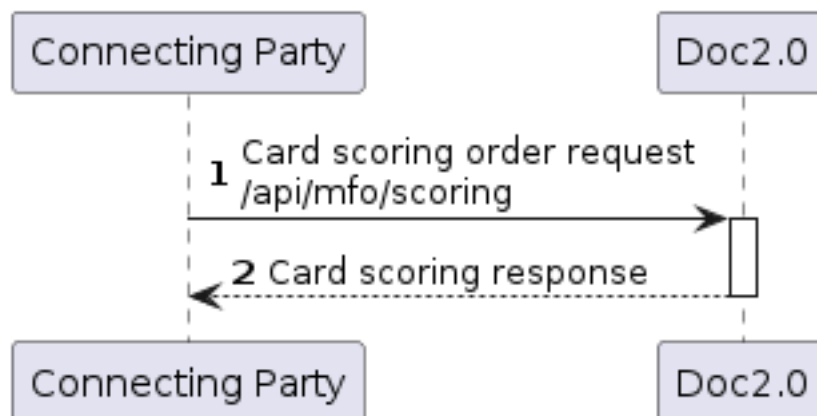
9.8 Card Scoring

- [Card Scoring](#)
 - [Card Scoring Flow](#)
- [Card Scoring Form](#)
 - [Card Scoring Form Flow](#)

9.8.1 Card Scoring

Card scoring is the additional security check of Customer's (Payer or Receiver) bank card which helps to prevent fraud and assess potential risks. Direct Card Scoring Flow contains card number, card mask (first 6 and last 4 digits) or card token (Card Reference ID) in initial API request.

Card Scoring Flow

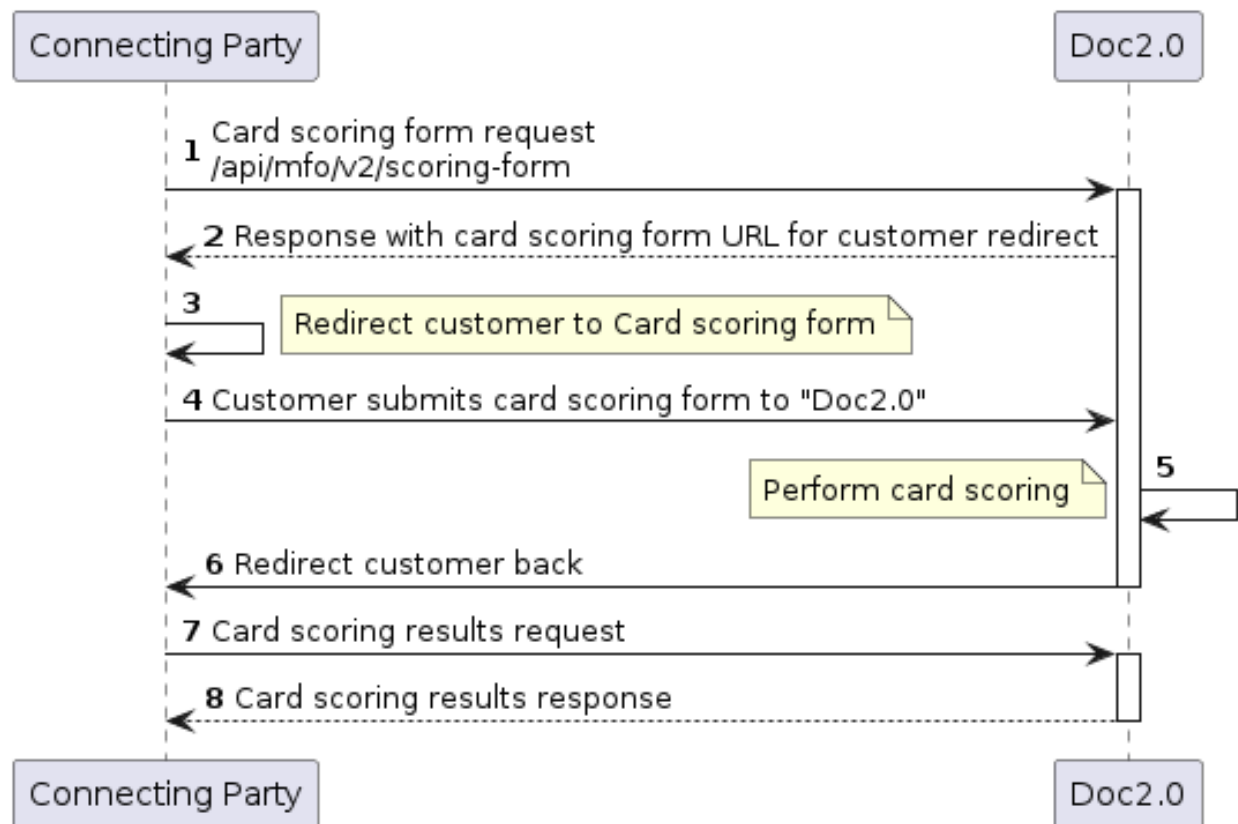


- (1) Connecting party initiates HTTPS POST request to the [/api/mfo/scoring](#).
- (2) Doc2.0 system returns response with card scoring response data set.

9.8.2 Card Scoring Form

Card scoring is the additional security check of Customer's (Payer or Receiver) bank card which helps to prevent fraud and assess potential risks. Card Scoring Form Flow allows to get HTML form to redirect Customer for card scoring procedure. Card scoring form accepts the card number which will be scored and then redirects Customer back to Connecting Party.

Card Scoring Form Flow



- (1) Connecting Party initiates card scoring form via HTTPS POST request to the </api/mfo/v2/scoring-form>.
- (2) Doc2.0 system responds with card scoring form URL for customer redirect where customer must be redirected to input card information for card scoring request.
- (3) Connecting Party redirects the customer to provided card scoring form URL.
- (4) Customer fills the card scoring form and submits it for card scoring to Doc2.0.
- (5) Doc2.0 processes card scoring request.
- (7) Connecting Party requests </api/mfo/v2/scoring-result/> and gets response with card scoring information.

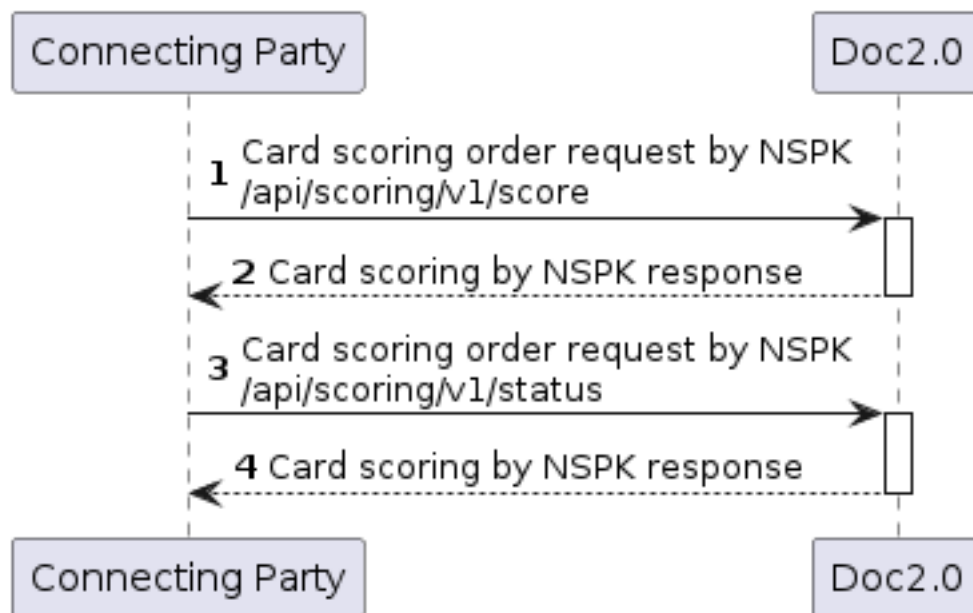
9.9 Card Scoring by NSPK

- [Introduction](#)
- [Card Scoring Flow](#)

9.9.1 Introduction

Card scoring is the additional security check of Customer's (Payer or Receiver) bank card by NSPK which helps to prevent fraud and assess potential risks. Direct Card Scoring Flow contains card number, card mask (first 6 and last 4 digits) or card token (Card Reference ID) in initial API request.

9.9.2 Card Scoring Flow



- (1) Connecting party initiates card scoring by NSPK with HTTPS POST request to the [/api/scoring/v1/score](#).
- (2) Doc2.0 system returns response with Order ID assigned to NSPK card scoring request.
- (3) Connecting party gets the result with HTTPS GET request to the [/api/scoring/v1/status](#). Connecting party should use Scoring status API calls to get the scoring status until final scoring status is received.
- (4) Doc2.0 system returns response with card scoring response data set.

9.10 Data migration

Sensitive information, such as cardholder data, must be encrypted for PCI-compliant migration. Encryption is made with Payneteasy PGP key.

After [importing](#)⁴⁶ Doc2.0 PGP key, run the following command to encrypt the files before sending. `file_name.gpg` will be created:

```
gpg --encrypt --recipient 3C7266AE file_name
```

Doc2.0 Public PGP Key

Key ID: 91D14F90

Key type: RSA

Key size: 4096 bits

Fingerprint: C519 7432 AF07 595A C081 3B83 3501 1DD3 CBC2 E0AB

User ID: Doc2.0 Import Key (PCI) <security@Doc2.0.ru>

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQINBGLNircBEADNI7fzBBc2NyhzhKGB1gI94ASwNqA37dxPaC8bW1hxThaZV0Aj
jPc1EbMT9UdSjJgHZafvAqYo0EUfJqCS1BFnhjTrHiWWBr1dKQ35dMkUB4UWmius
vungosQeFgAy70eY4MS+2r05DYaBylHHhc/0TZ+v0U0aQp8ch2Uu2C6N7wYT97nV
DC0373S5t0N36m6k1pXLYuTSzXX9ZgZLLYvK1gyItvBqziT7AsSHeGFFjCFygcCL
i3qrmT2S4JBDXQTDaSCWG36aGTerMvkGlxwixnpj6nw1E6GJsApQNY7qahn2rY3UK
40Fk7BniHLbFmR4AK5RvDWNtgu3jDdc1z9dS/6n3beM9HEXly3TjpmLZDa1CxhtH
W3o69m/3BbzL0jgW6XQX1QoCjuv1ejw6Hm1b5dphyab2xXfEXS40jghuTKTMK7TN
gAs0n8siD4zTedeWdo4rUUX/WATtyRAyFBDLz0RFxj4iRezGda1XfT0IgRqPc/3U
ggQh0T6BQuevdtuHjdSNY36TEJH0Pi28xMOPMMpMVA5nPBHX/L+I6Z8Z9/pGDZPa
mQwlg5XebtL2t3AakGQ8goobImavQrNgDPQU+h3maNAkBzKirqLVcAKjqG/KX7v
AMNUzzXCRqc/yGEobJuSa0wTpf2pJhHqExrAgbn3FEBiu42kY/wE20jyLQARAQAB
tDRQYXluZXRlYXN5IEltcG9ydCBLZXkgKFBDS5kgPHNlY3VyaXR5QHhew5ldGVh
c3kucnU+iQI/BBMBAgApBQJizYq3AhsDBQkDwmcABwsJCAcDAgEGFQgCCQoLBBYC
AwECHgECF4AACGkQNQEd08vC4KubdA/9G0okz/M0p6Nz7DE2u7Hz00VoP50MHQ+U
TdXINLYG0PWJGricF5ohw5jYp/iwgumAb9lXJBc0/h3LZBemW/H0/HCID+wFJT6p
kSoQ2ZGKL37MRdJ6kf9H+VMjGJYS0V0JoJlxWiwUhBjmjeWcAGx880vej9IzZsa0
F9hCgsB/PiTm95wdqpEfH0Cy+C5r03PoW0sa8v4bpMwL3PaNxiFaD+uorJX4k0F
EvGI0krm3o1C6gkZJR2f6RLnazDFLW0KmXU0ik1MEosKYCprY2Bk3wBZ0GskyLLY
jckppVdzfeX66vYUlcZr57m1k3GSZzCQH15fBFoX+yUY+IiowRVqjmneFP6rv9MY
CsvcL/2vdJ80u4pyRmRcxJiYhoNSxowWTenbhj40KHUfw7YH3/VLY2UI/LBedEMF
Dx0CqtHSNVogeb5RzzIbhySeXAut79In8AAsvtNT/OMF5ZrZtlnBi6wqZ0/invgv
UkdCngSL0+TZA0ySwfR0iNZaZJmZ/DfrBuP9p82HZRyK7X9GB/5Ziqs8WSt0IzJr
2iQVB9TCKIrE7kWHfA7HNzfWYgWU32lx/2R+0cH4zwuhFH2rKfrBR6Ekg18F0+m
tuYqMMBCagF0U60hFqTtTiQ3cz1A8inNlVkv9AHI/2R15w9hRb88139TtV7aeK2
jbYlHaIwmZC5Ag0EYs2KtwEQAMwaMSjHenq5rU/vVNhhisKKvrzrzHtRFiSnshs8
cWwtr2hyAVRb+loZa5trBs6paL/X+V/fAm7LWBSjt0z1gyZfgamf0PHvmWl5JHiK
06x22qYLS/F404d/tDwUPdPJPx/FXYu/q50B4IsJ0zfoKW2Zwp9hCUoY67EM+8c+
```

(continues on next page)

⁴⁶ <https://www.gnupg.org/gph/en/manual.html#AEN84>

(continued from previous page)

```

NF5pTqbZ8Fp7zM1oHIkhYRga2qXRW5LxhC8Xojc/nZp65qW+r2JVlBECa/A20L8
FsMM0bkdG2W6MWN8zXPayvGinx+PYRne9h8WiyJN9N44aG4No0kd4mXP09vGXAg
8dhNZxKMxmW0q0Bcme6aIVGJQtSEEjTL3qqSxYj8kUdIiR4Fznf5cYvjbzSgXRSi
tJdQu5e0k28c99t4TtnLzR1B6qrPDtwjtApoxw5YsjZKbKwe0m+DqzhRVS0soQE5
DC6CFkmdWybz+20/UNA4qcJIq4bAHYfYskFHkmiix71YLIthWq6F9+Fpnq8ZQp5
kX+TqxDyAqhIrYh13kJ/DSstpPe+GjebFWPNkvfXxJ8iwn0Uq0zuVerpbWfYF8SC
B0lpxMncoJgx5dIM0QRFcw17rv1+GK8Ga0+8avhZ5hqd4raUvxlY7ZInwsMn70sN
Jc2hH70fYF9U3Ww4o6p2IY/bh8V0Pu9nMJILXFuSN7PdcNpSGk3tdQgJ32seAbc
OWfnABEBAAGJAiUEGAECaA8FamLnircCGwwFCQPCZwAACgkQNQEd08vC4KtZvBAA
kyJd2aNJhpyQQMVZHUdnXwL/W16r+Hw0+gg1nXBTj0swNiNKq9mk78IWKJW+XIEC
5RucAq4c2pg8lCH0wuIpPexEKsHTE6PjFnx6DT6ykaZjZQNiPJAL7QoPu0eUGW2L
vBG4/qLJNKzQQQUHBLMV8MlQhNdx5wBbs8s6UGQ0Q+YUzYCBZ4thQb8SfRSNDGc3
0DitLJD+hSJGeDQ38/1brF8hCQuJc0MYpKv7rWdpBDXY0oWhDWckQo0F4+Ke+8ou
IUMWc/toyIHcjRP6XpihC2utFfQE4hlugRn2l0I8vkCQkEidXqYzZvfA7zqdmjX
LYWkISsF6jd7td/ScN8h9KW58Rq/fTiQ5eXsWbV5fY0wMsJJ6uXL/8/yfvJxw/Zp
YkHA+gIMvdmxTxP3qd+W9Hq+vToD8MbGqPFRvB29+xF6VmzB2KFyvNbiGnwSR7Ke
FuFbKkIoly1hlc7s/NlmaU2LS4ByvPh0Q6h7AJ07jJL+/wu2kDoQbCLZ85nRGath
j2b1ZsftaZk4vdZ0rRoTovta5gVYJjxMYwFlBqoucXZF3VXoiBz0Hec60h9YsK3a
GG01yxMkDDJPUSl5p43dfJIg9nVvpB37tFp+vn70gu01SLYsXChQWdNdw3BQT4pL
Plb3Yd+pl7shbg9LTSafrv7nNukxPzZmVklFL9w3Ec=
=m5/V
-----END PGP PUBLIC KEY BLOCK-----

```

9.11 Insurance

- [Introduction](#)
- [Example of insurance flow on the payment form](#)

9.11.1 Introduction

When a borrower visits the website or payment form of an MFI to repay a loan, they can insure the funds stored on their card, protecting them from theft and unauthorized access. In the current implementation, the funds are accepted by the MFI, which provides the service of including the Client in the list of insured individuals. This operation is charged according to the agreement. The MFI must add an offer to its payment form to justify the fee for this service. The data filled in by the Client (or pre-filled by the MFI via API) is added to the application for inclusion in the list of insured individuals. Before the operation is completed, there is an option to preview the document and make corrections if necessary.

Insurance is supported for the following types of transactions: [Payment via form](#), [Pre-authorization via form](#), [Recurring payment](#), [Recurring payment with pre-authorization](#), [Transfer](#).

To obtain the insurance document after completing the transaction, use [/api/v2/card-insurance-document](#).

Note:

This document can also be sent to the Customer's e-mail address.

9.11.2 Example of insurance flow on the payment form

To activate the insurance form, the corresponding operation must be enabled on the terminal.

It is also possible to enable or disable the editing of the insurance form field at the Connecting Party's request.

After initiating the transaction, the payment form requires entering card details and clicking on [Learn more](#).

A form will open where the following details need to be filled in: last name, first name, middle name, date of birth, passport series and number, passport issue date, issued by, registration address, phone number, e-mail.

At the bottom of the insurance data entry form, there is information about the insurance expiration date.

To complete data entry, click the Protect button.

Clicking the Cancel button will close the form, but the entered data will be saved upon reopening.

After filling out the form, a checkbox will appear next to the field I consent to the card protection service against fraud.

Оплачивайте безопасно кредитной картой

Номер карты:

Срок действия карты:

Имя держателя карты:

CVV2/CVC2:

☐ Даю своё согласие на подключение
услуги по защите банковской карты
от мошеннических действий

Стоимость: **250.00 рублей**

Услуга оказывается страховой N.

Страховая компания возместит сумму до
100 000 рублей в случае мошеннических
действий по карте.

Срок страхования: 30 календарных дней.
Услуга вступает в силу по истечении 14
календарных дней с даты оплаты с
возможностью отмены в течение 30 дней
с момента оформления.

[Скачать полис](#)

[Узнать больше информации](#) >



Безопасное соединение

Оплатить

[Отмена](#)

 **Защита карты**

Паспортные данные

100%

ЗАПОЛНЕНИЕ АНКЕТЫ

Фамилия

Павленко

Имя

Марина

Отчество

Павленко

Дата рождения

14.11.1990

Серия и номер паспорта

4510 973453

Дата выдачи паспорта

Защитить

Отмена

Ваша карта будет под защитой 1
месяц

Защитить

Отмена

To modify the entered data or check insurance terms, click [Learn more](#).

Next, complete the transaction by clicking the Pay button.

Оплачивайте безопасно кредитной картой

Номер карты:

Срок действия карты:

Имя держателя карты:

CVV2/CVC2:

☒ Даю своё согласие на подключение услуги по защите банковской карты от мошеннических действий

Стоимость: **250.00 рублей**

Услуга оказывается страховой N.

Страховая компания возместит сумму до 100 000 рублей в случае мошеннических действий по карте.

Срок страхования: 30 календарных дней.
Услуга вступает в силу по истечении 14 календарных дней с даты оплаты с возможностью отмены в течение 30 дней с момента оформления.

[📄 Скачать полис](#)

[Узнать больше информации >](#)



Безопасное соединение

Оплатить

[Отмена](#)



Ваша карта будет под защитой!
Срок действия - 1 месяц.

[← Вернуться](#)

[Редактировать](#)

[Отменить защиту](#)

[Оплатить](#)

REFERENCE**10.1 FAQ**

- Gather useful information
- Frequently asked questions during the integration
 - What does INVALID_CONTROL_CODE error mean?
 - What is the difference between ENDPOINTID or ENDPOINTGROUPID?
 - Is it possible to use Synchronous API?
 - How to implement 3-D Secure (Verified by Visa / MasterCard SecureCode) on Native mobile app?
 - What is client_orderid?
 - How to get an approved transaction on sandbox environment?
 - Why response is formatted with (0xA)?
 - Is it possible to use redirect_url to get transaction status for backoffice?
 - Is it possible to use GET instead of POST to submit data to Doc2.0 gateway?
 - What does INVALID_CALLBACK_URL mean?
 - How often can transaction status be polled?
 - Why transaction has been declined with base 64 fail reason?

10.1.1 Gather useful information

Please gather this information and provide it to support to quickly resolve issues which happened during the integration

- Please indicate whether sandbox or production environments were used.
- Please indicate whether or not the functionality worked at one time or not. If so, please indicate since when it stopped working.
- Please add relevant log information (orderid, serial-number, client_orderid, ENDPOINTID, ENDPOINTGROUPID) when any is available.

10.1.2 Frequently asked questions during the integration

What does INVALID_CONTROL_CODE error mean?

The calculation of the signature may be different for different types of API calls. The most common reasons are:

- Wrong credentials. Check credentials received from Payment Gateway manager.
- API calls to sandbox had production environment URL or vice versa.
- Production credentials (ENDPOINTID, ENDPOINTGROUPID, control key, etc) were used on sandbox or vice versa.
- Parameters for signature calculation are in incorrect order.
- The amount of the transaction is not converted to minimal decimal units.

The best way to check signature calculation is to use the request builder sections in API documentation:

- Choose the correct URL from the combobox with API calls selection, if present.
- After pressing the Sign button, please check signature value for the String to sign whether the same string is being signed or not.
- If the signing string looks correct, signature value must be the same as in Signature field.
- Login to UI and find the particular API requests and responses on the Integration panel screen in Tools section.

What is the difference between ENDPOINTID or ENDPOINTGROUPID?

Doc2.0 gateway can support any currency.

- ENDPOINTID supports only one currency.
- ENDPOINTGROUPID can support as many currencies as needed. Full list of currencies should be sent to Payment Gateway manager for project configuration.

Due to the different nature of these two entities it is impossible to use ENDPOINTID instead of ENDPOINTGROUPID and vice versa.

Is it possible to use Synchronous API?

All APIs are asynchronous, see [Async-Response](#). If using synchronous API is required, it should be implemented on Connecting Party side. Ask Payment Gateway manager to refine if it is possible to make synchronous calls in your integration.

How to implement 3-D Secure (Verified by Visa / MasterCard SecureCode) on Native mobile app?

Even in native app embedded WebView component should be used in order to open banking ACS server URL.

To process the transactions without redirect to 3-D Secure page Connecting Party's application must meet the following requirements:

- Payers are identified and there is a low fraud probability, in this case Connecting Party can ask Payment Gateway manager to open non3D account.
- Connecting Party have PCI DSS certificate and can gather cardholder data on their side, in this case Connecting Party can ask Payment Gateway manager to open account with Direct API support to avoid unnecessary redirects.

What is client_orderid?

This transaction identifier is generated on Connecting Party side. It is also referred as merchant-order-id in some APIs.

How to get an approved transaction on sandbox environment?

For different types of API Doc2.0 have different mechanisms for getting approved transactions and transactions in other statuses. For e-commerce transactions Doc2.0 is using different CVVs, see [test_cards](#). For mPOS transactions see [mpos](#).

Why response is formatted with (0xA)?

To improve readability of the response in most APIs, Doc2.0 adds (0xA) character at the end of each parameters value. To process the response from the Payment Gateway please, implement the handling mechanism for this character.

Is it possible to use `redirect_url` to get transaction status for backoffice?

It is not advisable.

All parameters for the final redirect are passed through the customer's browser. Parameters can be lost during the transmission over the internet. To get guaranteed result from Doc2.0 gateway see [Connecting Party Callbacks](#).

Is it possible to use GET instead of POST to submit data to Doc2.0 gateway?

Unfortunately it is not possible.

Transmitting parameters using HTTP GET is strictly forbidden to avoid possible data leaks.

What does `INVALID_CALLBACK_URL` mean?

Please use full public URL with `http://` or `https://`

For example, please pass <http://www.example.com> or <https://www.example.com> instead of `www.example.com`. Only following ports are allowed:

- 80, 8080 for HTTP
- 443, 8443 for HTTPS

How often can transaction status be polled?

Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.

Normally it takes up to 5 seconds to process the non3D transaction and up to 60 minutes for 3D transaction. Transactions made via some payment methods or payment channels, especially processed in offline mode, might stay in non-final status for several hours or even days, so there is no point in polling transaction status often for such cases. Connecting Party should rely on callback to get transaction final status, see [Connecting Party Callbacks](#).

Why transaction has been declined with base 64 fail reason?

Please use `urldecode` to process html parameter in Doc2.0 gateway response.

Doc2.0 is trying to help Connecting Party to correctly handle API responses. Base64 test gives Connecting Party an ability to check if 3DS page provided by bank will be correctly handled on production environment. If Connecting Party is getting this error, this means that `urldecode` probably was not used to parse the Doc2.0 gateway response or probably it works in incorrect way (for example in some versions of PHP it is needed to use `rawurldecode` instead).

10.2 3DS Overview

- [Doc2.0 3DS Flow](#)
- [3DS Implementation Scenarios](#)
- [3DS Decision Making Schema](#)
 - [Non3D Flow](#)
 - [3DS 2.x.0 Frictionless Flow](#)
 - * [3DS Method HTML Page Example](#)
 - * [Process 3DS Method Notification](#)
 - * [3DS Method Done HTML Page Example](#)
 - [3DS 2.x.0 Challenge Flow](#)
 - * [CReq HTML Page Example](#)
 - [3DS 1.0.2 Authentication Flow](#)
 - * [PaReq HTML Page Example](#)
 - [Simplified Authentication Flow](#)
- [Alternative cardholder authentication](#)
- [General diagram for form transactions](#)
- [General diagram for direct integrations](#)

10.2.1 Doc2.0 3DS Flow

Reference:

3-D Secure is a protocol designed to be an additional security layer for online credit and debit card transactions.

3-D Secure protocol has two main versions.

1.0 – which was introduced in 2001 and is still used by some acquirers (Services based on this protocol were adopted by different card schemes giving it different names: Verified by Visa or Visa Secure used by Visa, SecureCode used by Mastercard, ProtectBy used by Discover, J/Secure used by JCB International, American Express SafeKey used by American Express).

2.X – which was introduced in 2016 to strengthen and supersede the version 1.0 and it was aimed at complying with the new EU authentication requirements and resolving some short-comings of the original protocol.

Despite low level message exchange differences in the essence protocol version 2.X differs from 1.0 by introducing an option for the issuer of the card to perform an automatic risk assessment based on the additional contextual data sent from the payer browser or mobile app, whereas 1.0 version always requires payer to pass an authentication challenge. Correspondingly (and according to EMVCo spec) protocol version 2.X includes Frictionless flow (gathering contextual data and performing risk assessment) and Challenge flow (passing authentication challenge). The normal Payer's web flow which passes 3DS 2.X Authentication can be illustrated as below.

1. Payers fill out a payment form provided by Connection Party or Payment Gateway side and sends data.

Order summary

Test Order Description

Total: 10.42 USD

Pay securely with credit card

Credit or Debit card number:

Card expiration date:
01 / January ▼ 2022 ▼

Card printed name:

Card security code (CVV2/CVC2):

VISA

SECURE CONNECTION

[Process payment](#) [Cancel](#)

Note: For Transfer transactions, payment form might look as described below.

Order summary

Test

Total: 1.00 USD

Pay securely with credit card

Credit or Debit card number:

Card expiration date:

01 / January

2023

Card printed name:

Card security code (CVV2/CVC2):

Destination card number:



VISA

 **SECURE CONNECTION**

[Process payment](#) [Cancel](#)

- Contextual data is gathered and sent to the Issuer in a hidden process within the browser (Frictionless flow).

Order summary

Test Order Description

Total: 10.42 USD



Please wait, your payment is being processed...

This page refreshes automatically each 5 seconds...
Next time it will refresh in 3 seconds.

[Check result](#)

3. If Issuer finds the transaction risky Payer's browser gets redirected to the Issuer web page to pass authentication challenge (Challenge flow).



Authenticate using password

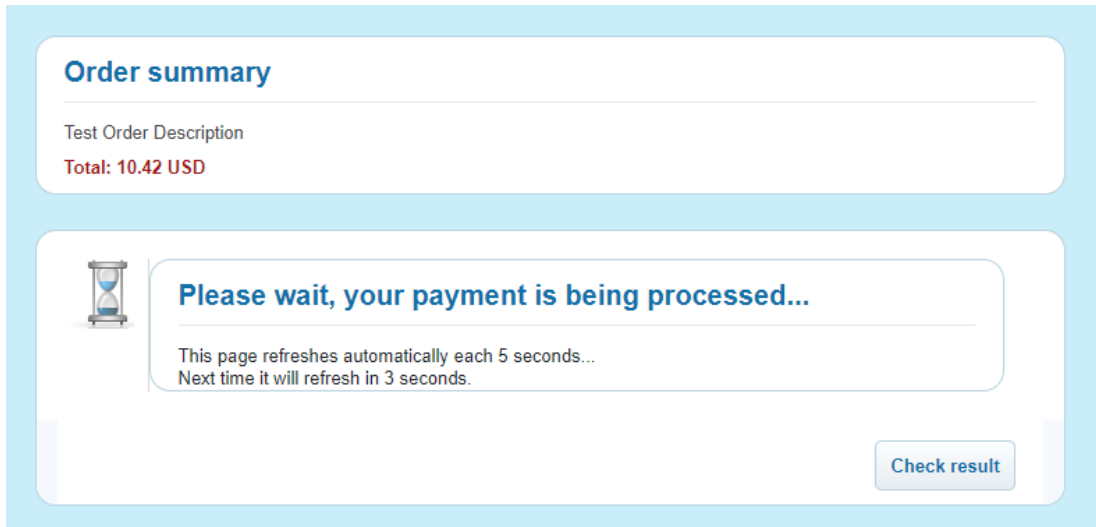
Please enter your password below.

Amount: EUR 3.00
Date: 20180314 09:52:32
Card Number: XXXX XXXX XXXX 0173
Personal Message:
Password:

[Help](#) [Cancel](#) [Submit](#)

If you experience any problems, please contact our Helpdesk on the back of your card.

4. When Payer passes checks the payment authorization is made saving the details related to the 3DS Authentication on the acquirer side.



5. Payer's browser gets redirected to the Connecting Party website to the resultant page.

Thank you

You've made a purchase

June 8, 2022 at 7:18:26 AM PDT

Item	Quantity	Price
Developer Registration Fee	1	\$25.00
		Tax \$0.00

Total **\$25.00**

Payment method

Visa ••••7087

Order number

PDS.6614-0389-4709-00899

[VIEW ONLINE](#)

[HELP CENTER](#)

[CONTACT US](#)

Due to the fact that both protocol versions still used the Connecting party have to deal with both of them as well as specific integrations when acquirer handles 3D Secure communication on its end. For this purpose, Payment Gateway provides universal/generic approach based on the API commands which allow the Connecting party to handle all mentioned cases. The Connecting Party has to start with the 3DS Decision Making Schema of the relevant Use-Case which in turn refers to the more detailed diagram describing specifics of the data handling and Payer's browser redirects related to the corresponding 3DS flow.

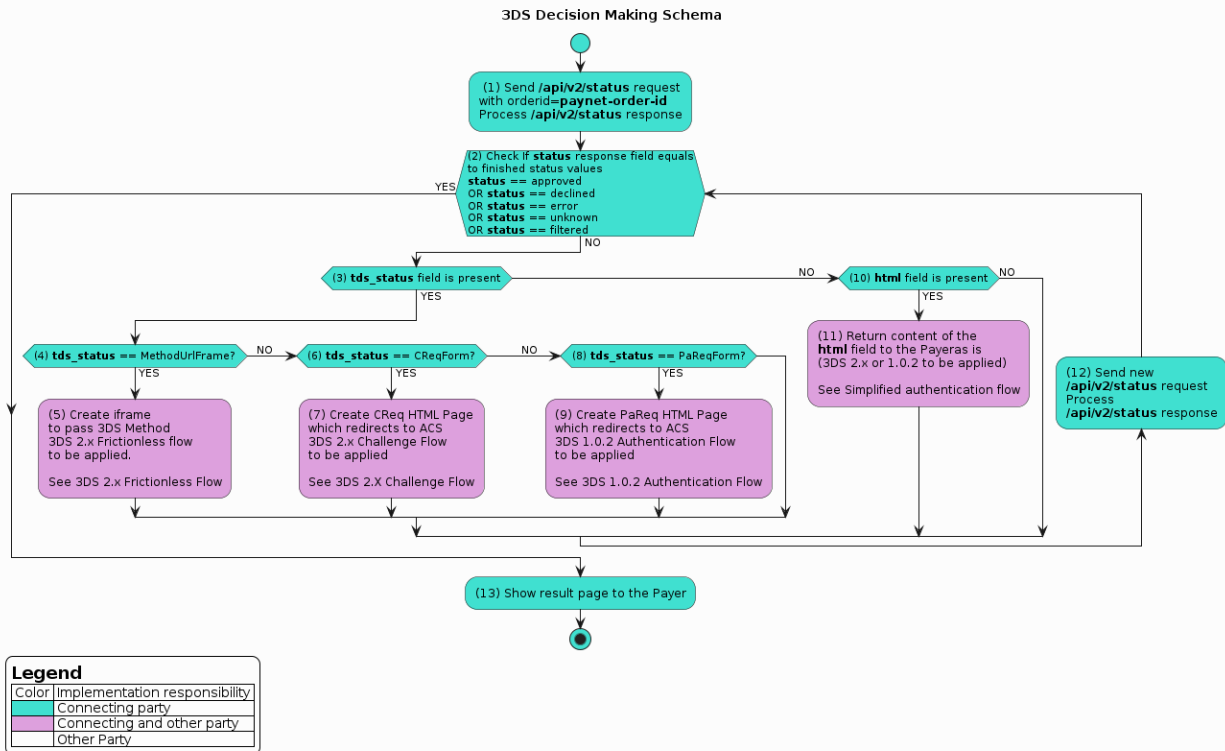
10.2.2 3DS Implementation Scenarios

Implementation Scenario	Instructions
3DS is not initiated (non3D).	3DS Authentication is not initiated. Follow 3DS Decision Making Schema of relevant Use-Case and implement Non3D Flow.
3DS is initiated and performed by Connecting Party (for PSPs and Acquirers).	3DS Authentication is initiated and performed by Connecting Party, results are uploaded to Payment Gateway in initial request. Follow 3DS Decision Making Schema of relevant Use-Case and implement Non3D Flow (3DS is performed before the initial request to Payment Gateway).
3DS is initiated by Payment Gateway and performed on Connecting Party side.	3DS Authentication is initiated by Payment Gateway. For 3DS 1.0.2, Payer Authentication is performed on Connecting Party side, results are uploaded to Payment Gateway. For 3DS 2.x, 3DS Method and Challenge are performed on Connecting Party side, results are uploaded to Payment Gateway. Additional fields for 3DS Method can be provided by Connecting Party in initial request to speed up the process for 3DS 2.x (tds_status=MethodUrlFrame will be skipped if 3DS Method is not supported by ACS). Follow 3DS Decision Making Schema of relevant Use-Case and implement all 3DS Flows.
3DS is initiated and performed by Payment Gateway.	3DS Authentication is initiated by Payment Gateway, all 3DS scenarios are performed on Payment Gateway side. Follow 3DS Decision Making Schema of relevant Use-Case, implement Simplified Authentication Flow (ignore tds_status parameter presence, use html or redirect-to parameters to redirect the Payer to Payment Gateway) and Non3D Flow (if Payer's card is not enrolled or processing is Non3D).

10.2.3 3DS Decision Making Schema

Connecting party has to implement all steps marked in green and purple. Below are the description for steps which reference specific API commands according to the step ID:

- (1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (5) If tds_status is MethodUrlFrame see [Frictionless 3DS 2.x](#).
- (7) If tds_status is CReqForm see [3DS 2.X Challenge](#).
- (9) If tds_status is PaReqForm see [3DS 1.0.2 Authentication](#).



(11) If tds status is not present, but html field is present, see [Simplified authentication](#).

(12) The same as point (1).

Non3D Flow

Transaction should be considered as non3D (no 3DS authentication) if all conditions are met:

1. steps 1-2-(12)-13 of [3DS decision making schema](#) were followed.
2. tds_status, html and redirect-to parameters were not present.
3. transaction received final status (approved, declined, error, filtered).

Note: Please note that transaction status “unknown” might appear for both 3DS and non3D transactions. See details in [Statuses](#).

3DS 2.x.0 Frictionless Flow

Connecting party has to implement all steps marked in green. Below are the description for steps which reference specific API commands according to the step ID:

- (1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (2) The same as point (1).
- (4) To construct 3DS Method HTML page see [example](#) below.
- (9) To process 3DS Method Notification see [Process 3DS Method Notification](#).
- (11) To construct 3DS Method Done HTML page see [example](#).
- (18) To upload method url result see </api/3ds/v1/upload-method-url-result/>.
- (19) The same as point (1).

3DS Method HTML Page Example

3DS Method HTML Page example below performs 3DS Method initiation through iframe technique and holding of the Payer's browser until the 3DS Method is not finished on the Connecting Party server. Once it is finished or timeout is reached Connecting Party has to follow the next steps following 3DS Decision Making Schema and redirect Payer's browser to the corresponding URL (either Challenge Flow or payment result page). Normal timeout value is 30 sec.

3DS Method HTML Page consists of the following parameters:

Parameter	Description
tds-method-url-frame-3ds-method-url	ACS 3DS Method URL is received by the Connecting Party in the /api/v2/status/ response.
tdsMethodUploadUrl	Connecting Party's URL, where the result comes after the completion of the flow.
threeDSMethodData	JSON value containing threeDSServerTransID and threeDSMethodNotificationURL. See below how to construct threeDSMethodData .

Construct threeDSMethodData

In order to **construct threeDSMethodData** the Connecting Party has to use the following parameters.

Parameter	Description
threeDSServerTransID	Universally unique transaction identifier assigned by the 3DS Server to identify a single transaction. Received in the /api/v2/status/ response as tds-method-url-frame-3ds-server-trans-id parameter.

continues on next page

Table 3 – continued from previous page

Parameter	Description
threeDSMethodNotificationURL	The URL that will receive the notification of 3DS Method completion from the ACS. The Connecting Party is responsible for providing valid threeDSMethodNotificationURL. Normally when the Connecting Party receives any request on this URL is has to return 3DS Method Done HTML Page (see example below).

threeDSMethodData construction example

1. Construct threeDSMethodData JSON.

```
{
  "threeDSServerTransID": "3d671629-a410-4a5d-9288-b38ceadd41f2",
  "threeDSMethodNotificationURL": "https://connectingparty.com/3ds-method-complete/"
}
```

2. Apply base64 url encoding to resultant JSON.

```
eyJ0aHJlZURTU2VydMvyVHJhbnNJRCI6IjNkNjc5NjI5LWE0MTAtNGE1ZC05Mjg4LWIzOGNlYWwRkNDFmMiIsInRocmVlRFNNZXRo
```

Generate Fingerprint

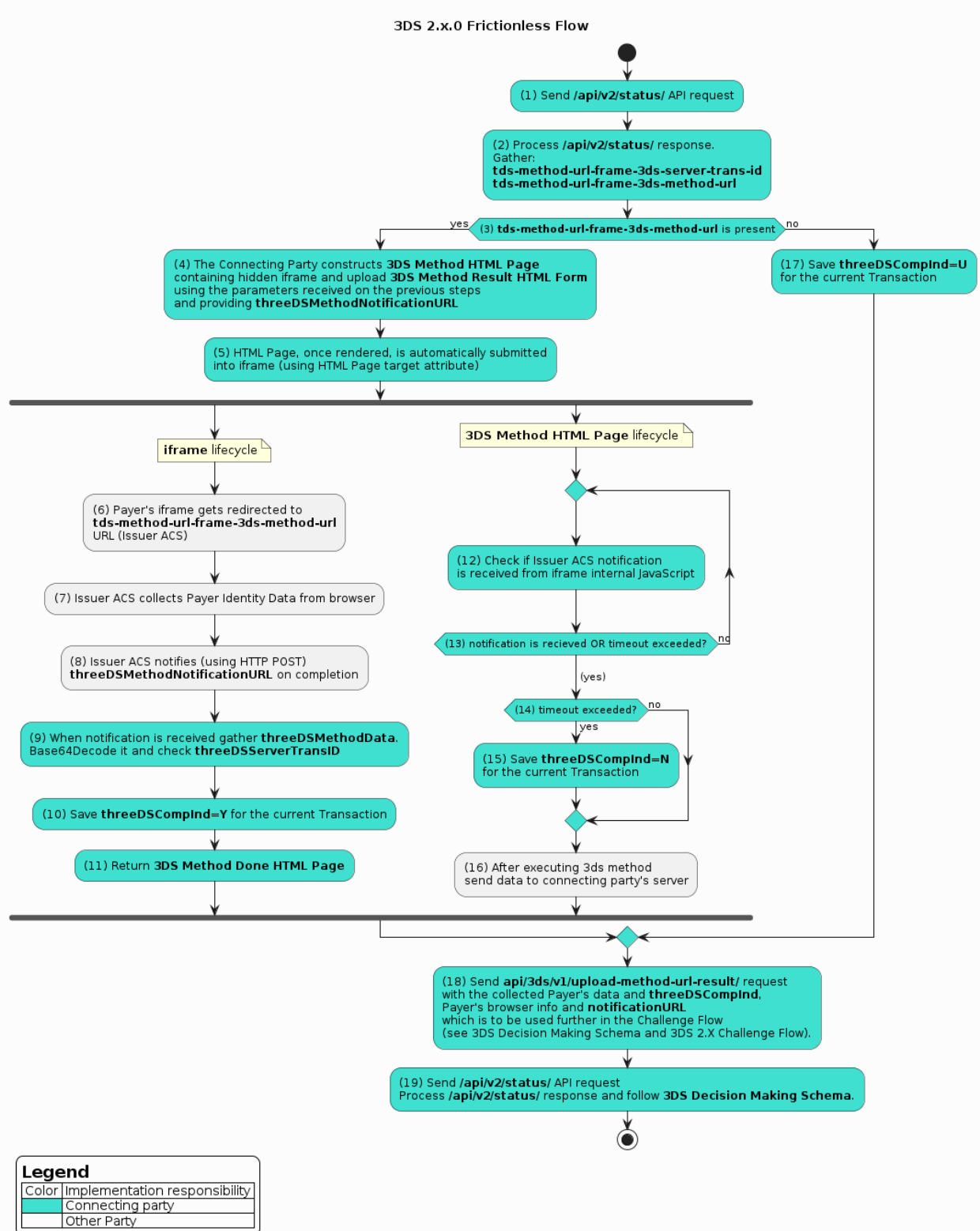
The 3DS Method can be optionally used by issuers to gather browser fingerprints using JavaScript. This is done by loading a URL in a hidden iframe, before the authentication. This iframe will then execute some fingerprinting JavaScript, before POST'ing to the prespecified URL belonging to the requestor. The 3DS Method fingerprint result is tied to the authentication by the threeDSServerTransID.

```
function gatherBrowserData() {
  var colorDepth = screen.colorDepth; // 24
  var javaEnabled = navigator.javaEnabled(); // true
  var browserLanguage = navigator.language; // en_US
  var screenHeight = screen.height; // 1080
  var screenWidth = screen.width; // 1920
  var userAgent = navigator.userAgent; // Mozilla/5.0 (Windows NT 6.1; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36
  var browserTimezoneZoneOffset = new Date().getTimezoneOffset(); // 0
}
```

Construct 3DS Method HTML page example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8"/>
  <title>ACS v2 3DS Method ...</title>
  <style>
    .progress{
      position: absolute;
      height: 10px;
      width: 100%;
    }
    .progress .color{
      position: absolute;
      background-color: #444;
      width: 100%;
      height: 10px;
      animation: progres 11s linear;
    }
  </style>
</head>
<body>
  <div class="progress">
    <div class="color"></div>
  </div>
</body>
</html>
```

(continues on next page)



(continued from previous page)

```

}
@keyframes progres{
  0%{
    width: 0%;
    background-color: #ddd;
  }
  25%{
    width: 50%;
    background-color: #ccc;
  }
  50%{
    width: 75%;
    background-color: #bbb;
  }
  75%{
    width: 85%;
    background-color: #aaa;
  }
  100%{
    width: 100%;
    background-color: #777;
  }
};
</style>
<script>
var fallbackTimeout = null;
function processForm() {
  document.title = "ACS v2 AReq ...";
  document.getElementById( "browserJavaEnabled" ).value = navigator.
  ↪ javaEnabled();
  document.getElementById( "browserJavascriptEnabled" ).value = true;
  document.getElementById( "browserLanguage" ).value = navigator.language;
  document.getElementById( "browserColorDepth" ).value = screen.colorDepth;
  document.getElementById( "browserScreenHeight" ).value = screen.height;
  document.getElementById( "browserScreenWidth" ).value = screen.width;
  document.getElementById( "browserTZ" ).value = new Date().
  ↪ getTimezoneOffset();
  document.autoForm.submit();
}
function onPostMessage(event) {
  if(!event.data.hasOwnProperty('methodNotification')) {
    return;
  }
  if(fallbackTimeout != null) {
    clearTimeout(fallbackTimeout);
    fallbackTimeout = null;
  }
  document.getElementById( "threeDSCompInd" ).value = 'Y';
  processForm();
}
function onPageLoaded() {
  fallbackTimeout = setTimeout(processForm, 10 * 1000);
  document.methodForm.submit();
  window.addEventListener('message', onPostMessage);
}
</script>

```

(continues on next page)

(continued from previous page)

```

</head>
<body onload="onPageLoaded()">
<div class="progress">
  <div class="color"></div>
</div>
<iframe style="width:0; height:0; border:0;" name="methodFrame"></iframe>
<form name="methodForm" target="methodFrame" action="[=tds-method-url-frame-3ds-method-
url]" method="POST">
  <input type="hidden" name="threeDSMethodData" value="[=threeDSMethodData]">
</form>
<form name="autoForm" action="[=tdsMethodUploadUrl]" method="post">
  <input type="hidden" name="threeDSServerTransID" value="[=threeDSServerTransID]" />
  <input type="hidden" name="threeDSCompInd" id="threeDSCompInd" value="N" />
  <input type="hidden" name="browserJavaEnabled" id="browserJavaEnabled"
  value="" />
  <input type="hidden" name="browserJavascriptEnabled" id="browserJavascriptEnabled
  value="" />
  <input type="hidden" name="browserLanguage" id="browserLanguage"
  value="" />
  <input type="hidden" name="browserColorDepth" id="browserColorDepth"
  value="" />
  <input type="hidden" name="browserScreenHeight" id="browserScreenHeight"
  value="" />
  <input type="hidden" name="browserScreenWidth" id="browserScreenWidth"
  value="" />
  <input type="hidden" name="browserTZ" id="browserTZ"
  value="" />
  <noscript>
    <input type="submit" name="submit" value="Upload 3DS Method Result" />
  </noscript>
</form>
</body>
</html>

```

Process 3DS Method Notification

When 3DS Method is completed, the Connecting Party receives HTTP POST request at threeDSMethodNotificationURL with threeDSMethodData, which contains threeDSServerTransID (in base64 encoded JSON).

1. Get threeDSMethodData.

```
threeDSMethodData=eyJ0aHJlZURTU2VydMvyVHJhbnNJRCI6IjNkNjcXNjI5LWE0MTAtNGE1ZC05Mjg4LWIzOGNlYWVRkNDFmMi
```

2. Apply base64 url decoding to get JSON, which contains threeDSServerTransID.

```
{"threeDSServerTransID": "3d671629-a410-4a5d-9288-b38ceadd41f2"}
```


3DS Method Done HTML Page Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8"/>
  <title>ACS v2 3DS Method Notification Handler...</title>
  <script>
    window.parent.postMessage({ methodNotification: "COMPLETE" }, "*");
  </script>
</head>
<body>
<p>This should not be displayed</p>
</body>
</html>
```

3DS 2.x.0 Challenge Flow

- (1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (2) The same as point (1).
- (3) To create CReq HTML Page see [example](#).
- (5) To implement CRes redirect see [CRes redirect](#).
- (7) To upload CRes result see </api/3ds/v1/upload-cres-result/>.
- (8) The same as point (1).

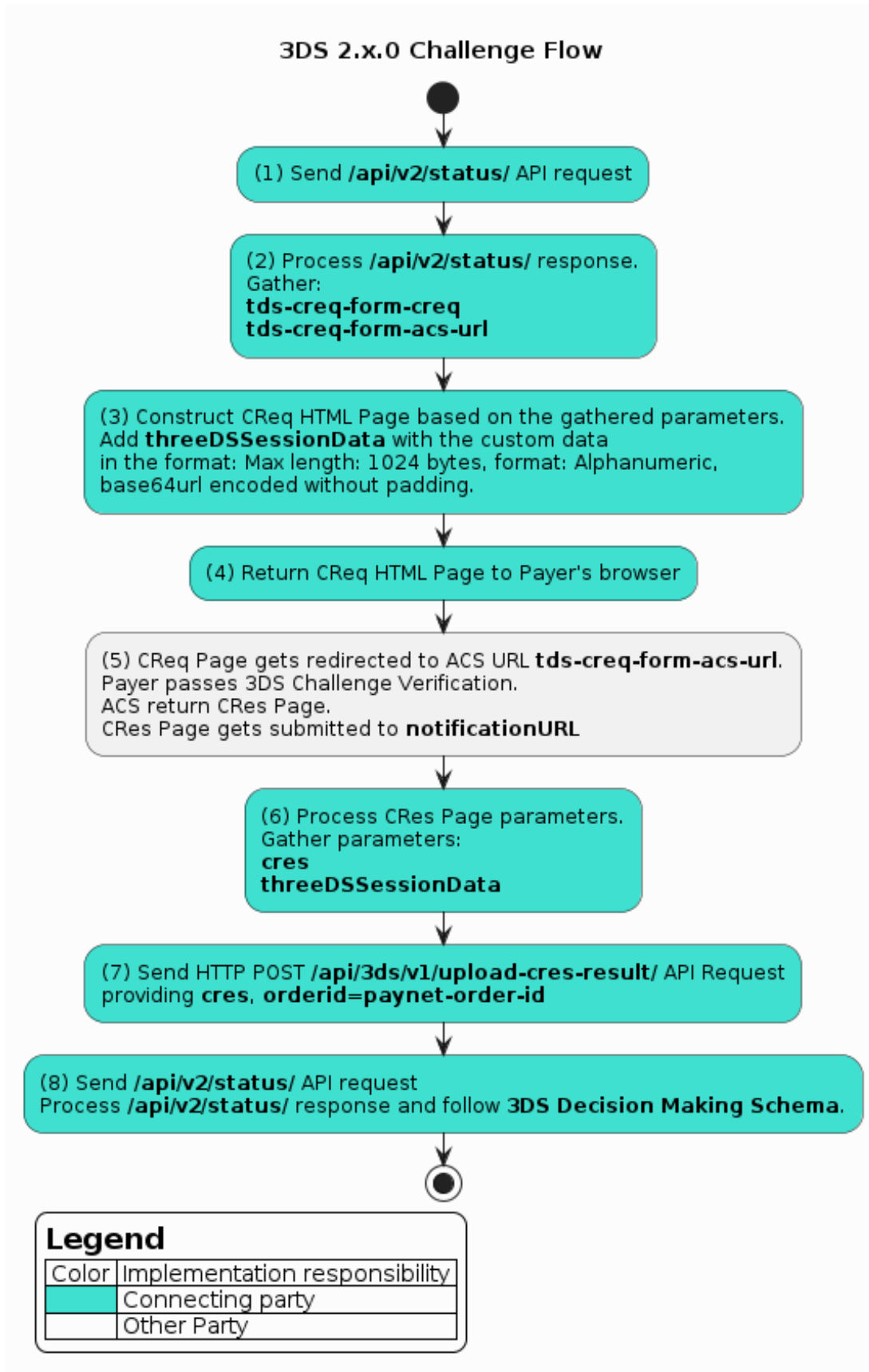
CReq HTML Page Example

CReq HTML Page redirects the Payer's browser to ACS Server URL, provided in tds-creq-form-acs-url parameter. The result CRes value will be returned from ACS to notificationURL provided by Connecting Party in </api/3ds/v1/upload-method-url-result> request during 3DS 2.x.0 Frictionless Flow.

Field	Description	Necessity
creq	ACS 3DS CReq data, which received by the Connecting Party in the /api/v2/status/ response. The same as tds-creq-form-creq.	Required
threeDSSESSIONData	value which will be posted back within CRes to notificationURL at the end of the process. Max length: 1024 bytes, format: Alphanumeric, Base64url encoded without padding.	Optional

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <title>Redirecting ...</title>
  <script type="text/javascript" language="javascript">
    function makeSubmit() {
      document.returnform.submit();
    }
  </script>
</head>
<body>
  <p>Redirecting ...</p>
</body>
</html>
```

(continues on next page)



(continued from previous page)

```

    }
  </script>
</head>
<body onLoad="makeSubmit()">
<form name="returnform" action="https://acs.bank-domain.com/mdpayacs/creq" method=
  "POST">
  <input type="hidden" name="creq" value=
  "eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6ImM1NDA5N2VhLTc0ZTctNDE2My05MTQ4LTNjMTY1NTg3NGIwMCIsImFjc1RyYW5zS
  ">
  <input type="hidden" name="threeDSsessionData" value="NjY4MDU3NQ">
  <noscript>
    <input type="submit" name="submit" value="Press this button to continue"/>
  </noscript>
</form>
</body>
</html>

```

3DS 1.0.2 Authentication Flow

- (1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (2) The same as point (1).
- (3) To construct PaReq HTML Page see [example](#).
- (5) To implement PaRes redirect see [PaRes redirect](#).
- (10) To upload PaRes result see </api/3ds/v1/upload-pares-result/>.
- (11) The same as point (10).

PaReq HTML Page Example

PaReq HTML Page redirects the Payer's browser to ACS Server URL, provided in tds-pareq-form-acs-url parameter.

PaReq HTML Page consists of the following parameters:

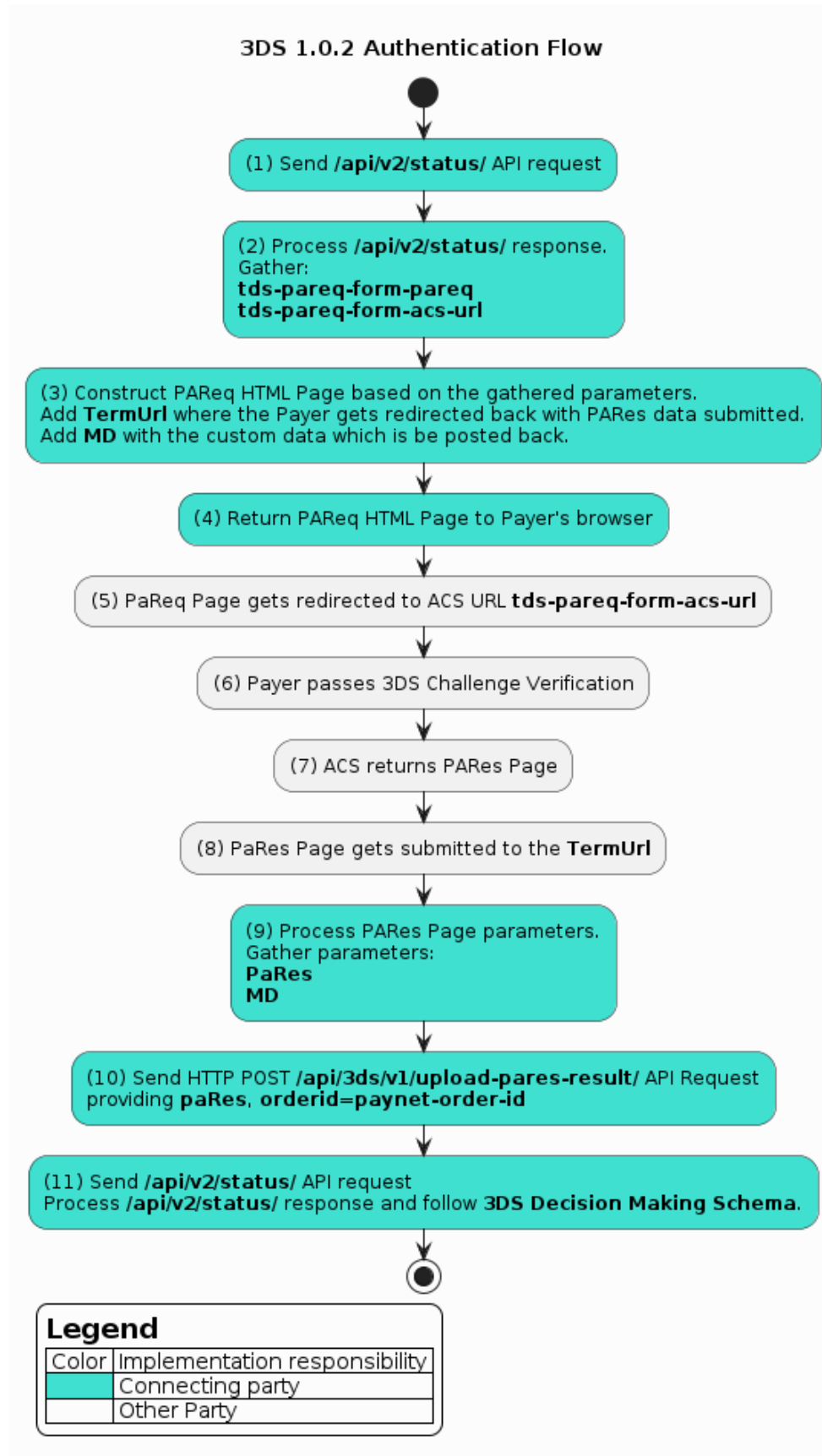
Field	Description	Necessity
tds-pareq-form-acs-url	ACS 3DS PaReq URL is received by the Connecting Party in the /api/v2/status/ response.	Required
MD	Connecting Party Data, which comes back to your termination page.	Optional
PaReq	ACS 3DS PaReq data, which received by the Connecting Party in the /api/v2/status/ response. The same as tds-pareq-form-pareq.	Required
TermURL	URL of termination page, where the Payer gets redirected back with PaRes data submitted.	Required

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">

```

(continues on next page)



(continued from previous page)

```

<title>Loading acs..</title>
<script type="text/javascript" language="javascript">
  function makeSubmit() {
    document.returnform.submit();
  }
</script>
</head>
<body onLoad="makeSubmit()">
<form name="returnform" action="$tds-pareq-form-accs-url" method="POST">
  <input type="hidden" name="MD" value="some_merchant_data"/>
  <input type="hidden" name="PaReq" value="$tds-pareq-form-pareq"/>
  <input type="hidden" name="TermUrl" value="https://termination.page"/>
  <noscript>
    <input type="submit" name="submit" value="Submit"/>
  </noscript>
</form>
</body>
</html>

```

Simplified Authentication Flow

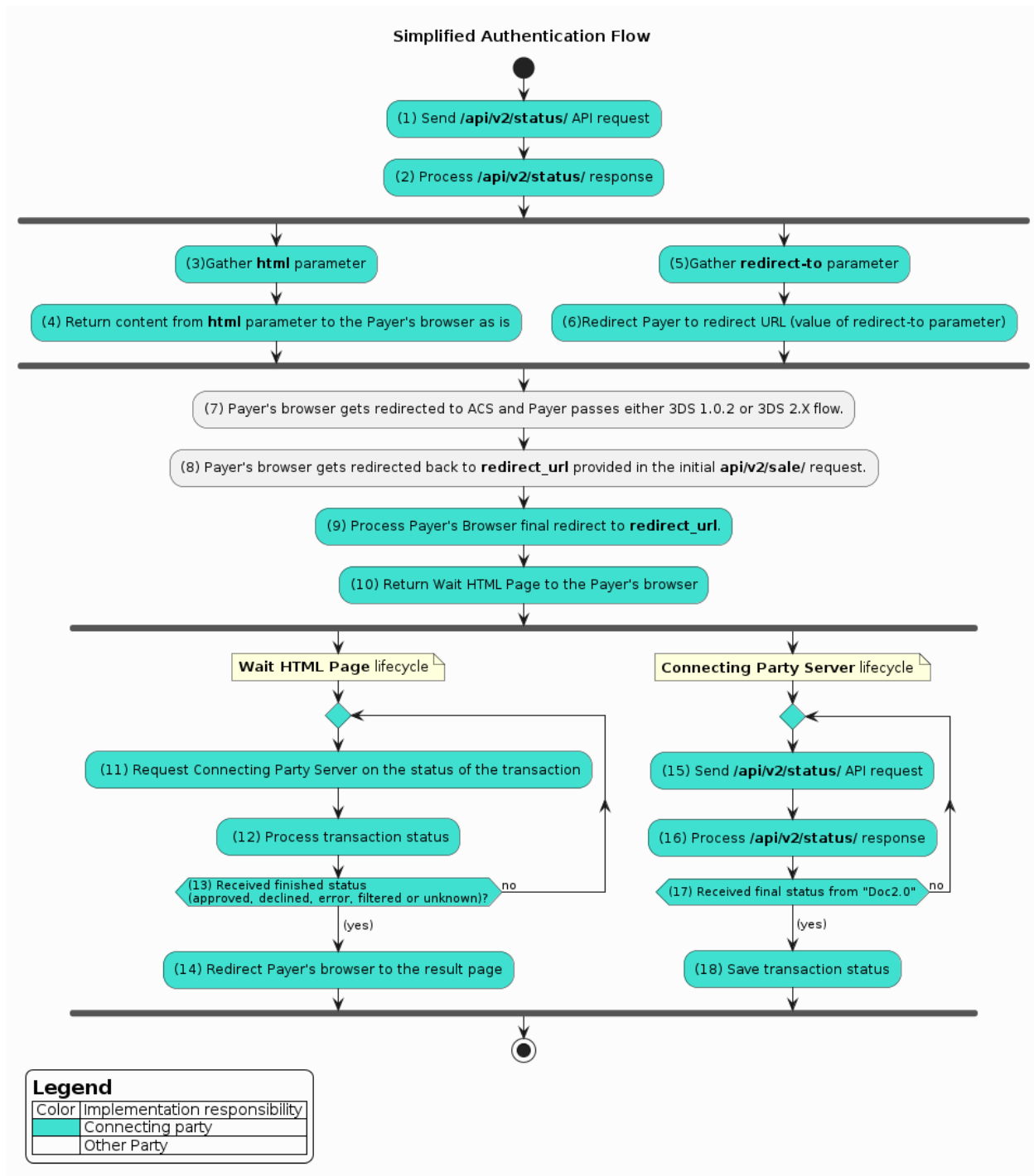
- (1) To implement order status request see </api/v2/status/>. Status should be requested multiple times with 3-5 seconds interval until final status will be received in response.
- (2) The same as point (1).
- (6) To implement final redirect see [Final redirect](#).
- (7) The HTML wait page on Connecting Party side can have custom design and should communicate with Connecting Party server as described on the diagram.
- (12) and (13) The same as point (1).

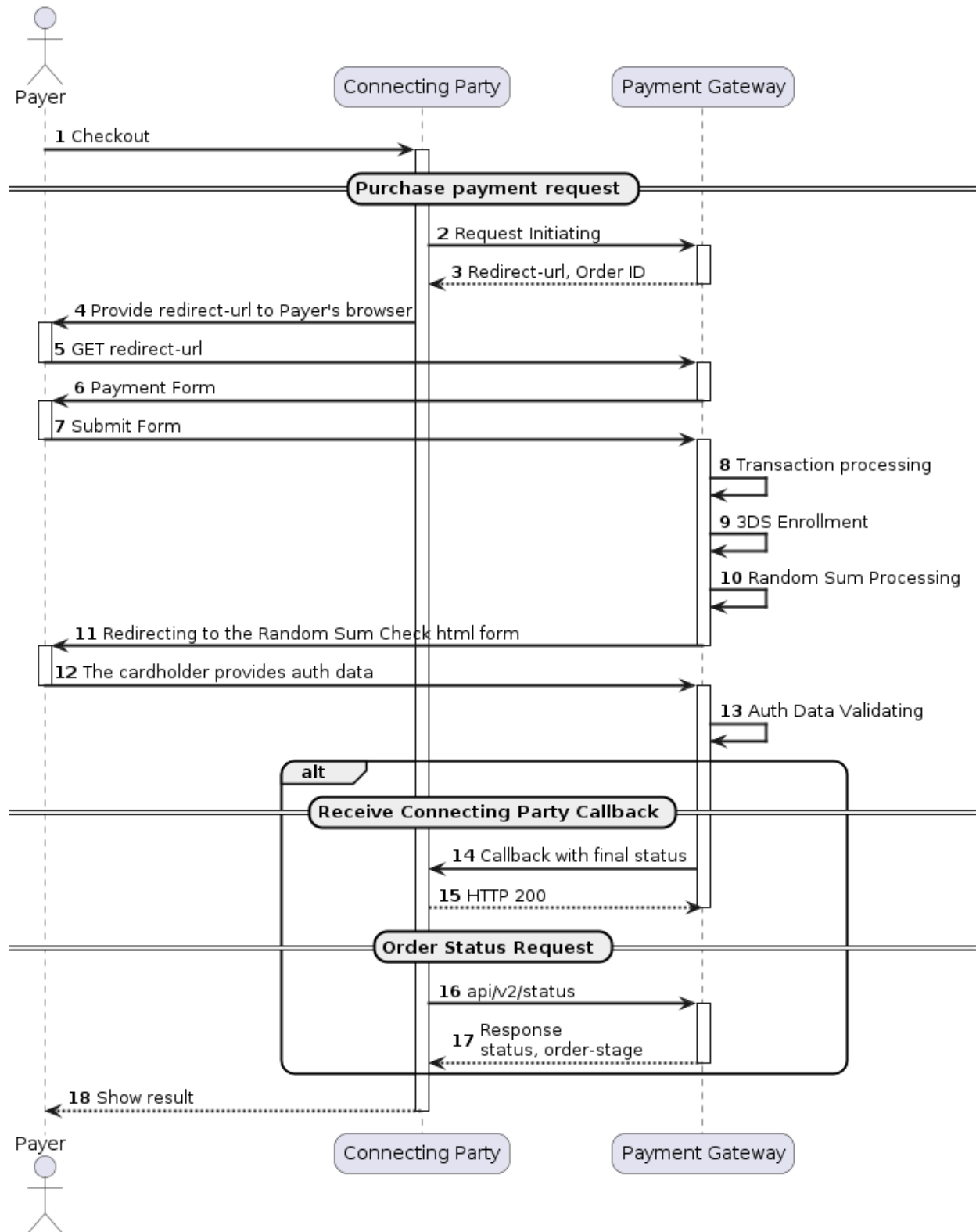
10.2.4 Alternative cardholder authentication

Payment Gateway supports alternative methods for cardholder authentication if card is not enrolled to 3DS (negative 3DS enrollment response). One of such methods is random sum check. In this method, Payment Gateway initiates an additional preauthorization transaction to hold a random small amount on cardholder's account and sends a special form for the Payer to enter the amount being held. If the amount is correct, Payment Gateway continues to process the initial transaction. The small amount hold is cancelled automatically.

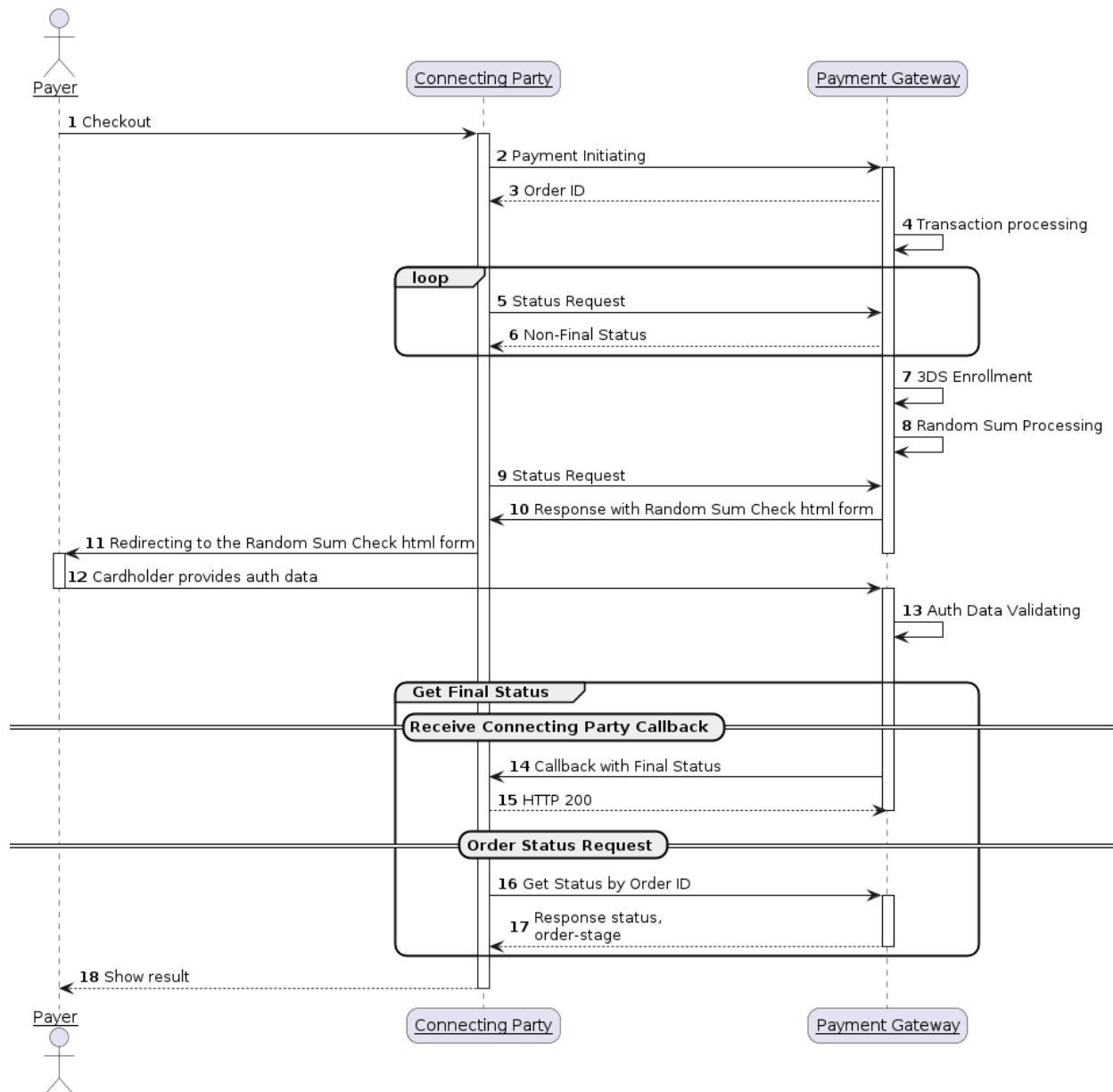
10.2.5 General diagram for form transactions

- (2) See relevant Use-Case to get details about the exact API command to initiate transaction.
- (13) To implement callback with final status handling see [Connecting Party Callbacks](#).
- (15) Please see [Status List](#) to get more information about Final Statuses.





10.2.6 General diagram for direct integrations



(2) See relevant Use-Case to get details about the exact API command to initiate transaction.

(9) Status Request continues in parallel all the time starting from step 5.

(11) To implement callback with final status handling see [Connecting Party Callbacks](#).

(13) Please see [Status List](#) to get more information about Final Statuses.

10.3 Forms Customization

- [Payment Page Display Logic](#)
- [Payment Page Customization](#)
 - [Payment Page Example](#)
 - [Payment Page Autofill](#)
 - [Payment Page Template Macros](#)
- [Prefilled Cardholder Data in Payment Page](#)
 - [Prefilled Cardholder Data Payment Page Example](#)
 - [Prefilled Cardholder Data Payment Page Macros](#)
- [Wait Page Customization](#)
 - [Wait Page Template](#)
 - [Wait Page Template Macros](#)
- [Finish Page Customization](#)
 - [Finish Page Template](#)
 - [Finish Page Macros](#)
- [Payment Cashier Page Customization](#)
 - [Parallel Form Template Sample](#)
 - [Parallel Form Macros](#)
 - [Parallel Form Scripts](#)

10.3.1 Payment Page Display Logic

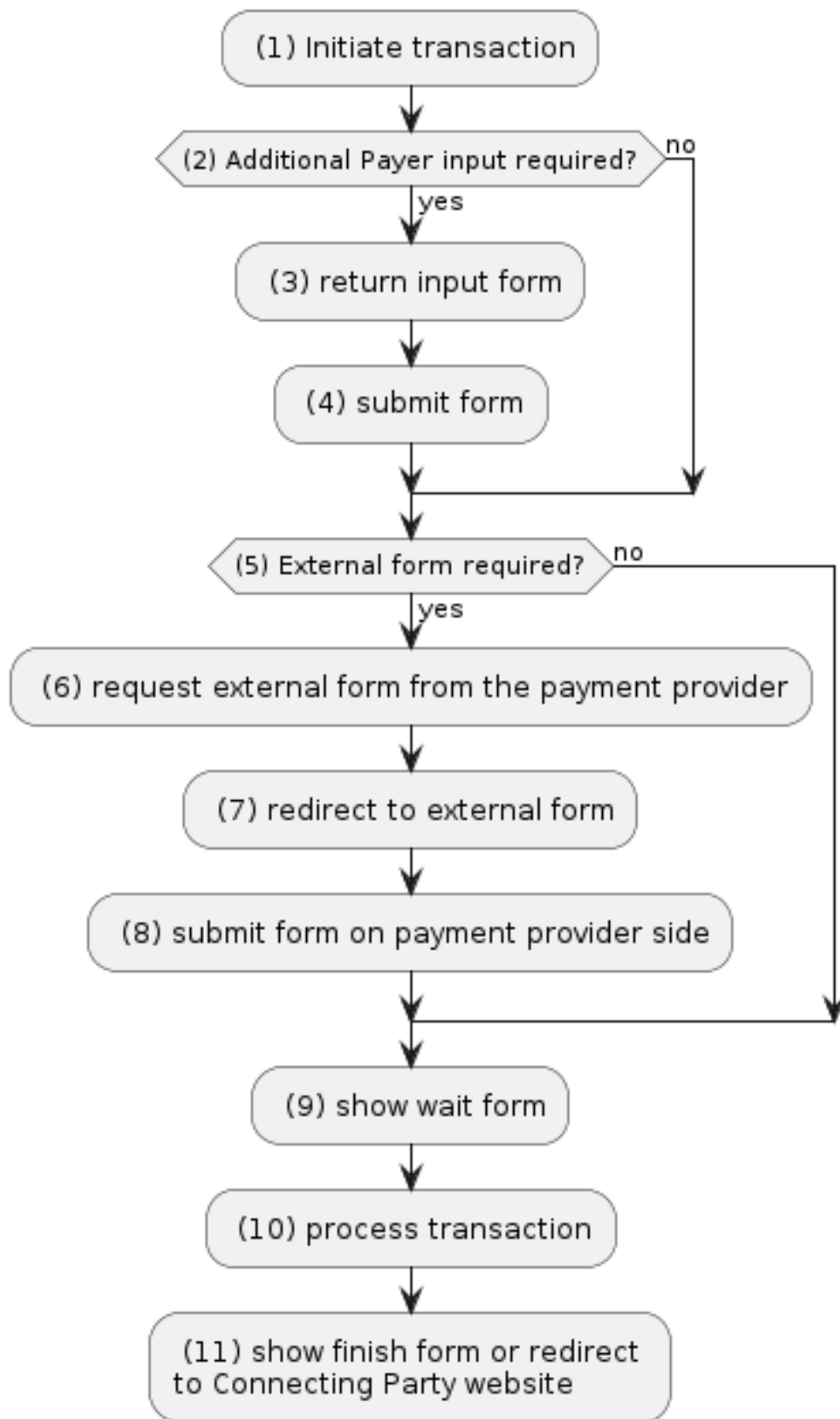
(1) Transactions can be initiated with API requests, batch upload or via Virtual terminal on UI. Each payment method has its own payment page display logic.

Note: Payment Cashier might have multiple transactions initiated within the same payment session, because it can have multiple payment methods available for the Payer. When Payer selects payment tab, respective auxiliary transaction is initiated. See [Payment Cashier Page Customization](#).

(3) For credit card payment method form is displayed on Payment Gateway side and can be customized. See [Payment Page Customization](#). Some other payment methods may have additional forms on Payment Gateway side. Contact Doc2.0 support for details.

(7) Some payment methods require the Payer to be redirected to their own form. That form is not hosted by Payment Gateway and can't be customized.

(9) Until transaction reaches final status, Payment Gateway displays [Wait Form](#) for the Payer.



(11) After the transaction reaches the final status, Payment Gateway displays [Finish Form](#) for the Payer or [Redirect](#) to Connecting Party website.

10.3.2 Payment Page Customization

Payment Page Example

```
<html>
<head>
<script type="text/javascript">
  function isCCValid(r){var n=r.length;if(n>19||13>n)return!1;
    for(i=0,s=0,m=1,l=n;i<l;i++)d=parseInt(r.substring(l-i-1,l-i),10)*m,s+=d>=10?d
    ↪ %10+1:d,l==m?m++:m--;
    return s%10==0?!0:!1}
</script>
</head>
<body>
<h3>Order #${MERCHANT_ORDER_ID} - ${ORDERDESCRIPTION}</h3>
<h3>Total amount: ${AMOUNT} ${CURRENCY} to ${MERCHANT}</h3>
<form action="${ACTION}" method="post">
  <div>Cardholder name: <input name="${CARDHOLDER}" type="text" maxlength="64"/></div>
  <div><label for="cc-number">Credit Card Number</label> <input id="cc-number" name="$
  ↪ {CARDNO}" type="text" maxlength="19" autocomplete="cc-number"/></div>
  <div>Card verification value: <input name="${CVV2}" type="text" maxlength="4"
  ↪ autocomplete="off"/></div>
  <div>
    Expiration date:
    <select class="expiry-month" name="${EXPMONTH}" size="1" autocomplete="cc-exp-
    ↪ month" >
      <option value="01">January</option><option value="02">February</option><option
    ↪ value="03">March</option>
      <option value="04">April</option><option value="05">May</option><option value=
    ↪ "06">June</option>
      <option value="07">July</option><option value="08">August</option><option value=
    ↪ "09">September</option>
      <option value="10">October</option><option value="11">November</option><option
    ↪ value="12">December</option>
    </select>
    <select class="expiry-year" id="cc-exp-year" name="${EXPYEAR}" size="1"
    ↪ autocomplete="cc-exp-year">
      ${EXPIRE_YEARS}
    </select>
  </div>
  <div><label for="dest-number">Destination card number:</label> <input id="dest-
  ↪ number" name="${DESTINATIONCARDNO}" type="text" maxlength="19" autocomplete="off"/>
  ↪ </div>
  ${INTERNAL_SECTION}
  #if(${card_error})
  <div style="color: red;">${card_error}</div>
  #end
  <input name="submit" onclick="return isCCValid(document.getElementById('cardnumber').
  ↪ value);" type="submit" value="Pay"/>
</form>
</body>
</html>
```

Note: The code described below allows to submit destination card data in [Transfer](#) type transactions:

```
<div><label for="dest-number">Destination card number:</label>
<input id="dest-number" name="${DESTINATIONCARDNO}" type="text" maxlength="19"
  ↳autocomplete="off"/></div>
```

Payment Page Autofill

If Connecting Party wants to use autofill in payment form, certain element attributes `<id>` `<autocomplete>` `<label for>` should be hardcoded in the following manner:

```
<label for="cc-number">Credit Card Number</label><span class="form-label-comment">The
  ↳13-19 digits on the front of your card</span>
<input class="card-number-field" id="cc-number" name="${CARDNO}" type="text"
  ↳maxlength="19" autocomplete="cc-number" />
```

If autofill in the transfer form must be used, certain element attributes `<id>` `<autocomplete>` `<label for>` should be hardcoded in the following manner:

```
#if ($INPUT_SOURCE_CARD_CARDHOLDER)
<!-- Card printed name field -->
<li class="form-li">
  <label class="form-label" for="cc-name">Card printed name:</label>
  <input class="form-name-field" id="cc-name" name="${CARDHOLDER}" type="text"
    ↳maxlength="50" autocomplete="cc-name" value="${CARDHOLDER_VALUE}"/>
</li>
#end
#if ($INPUT_SOURCE_CARD_CVV2)
<!-- CVV field -->
<li class="form-li">
  <label class="form-label" for="${CVV2}">Card security code (CVV2/CVC2):</label>
  <input class="form-cvv-field" name="${CVV2}" id="${CVV2}" type="password"
    ↳maxlength="4" autocomplete="off"/>
</li>
#end
#end
#if ($INPUT_DESTINATION_CARD)
<!-- Destination Card Number field -->
#if ($!DESTINATIONCARDNO)
<li class="form-li">
  <label class="form-label" for="${DESTINATIONCARDNO}">Destination card number:</
    ↳label>
  <input class="form-number-field" id="${DESTINATIONCARDNO}" name="$
    ↳{DESTINATIONCARDNO}" type="text" maxlength="19" autocomplete="off"/>
</li>
#end
#end
```

Doc2.0 has default payment form template which supports autocomplete. In case if Connecting Party wants to add additional fields for autocomplete, this [specification](#)⁴⁷ should be used for naming references.

⁴⁷ <https://html.spec.whatwg.org/multipage/form-control-infrastructure.html#autofill/>

Payment Page Template Macros

Field Name Macro	Field Value Macro	Description
<code>\${APPLE_PAY}</code>	n/a	Apple Pay buy button will be available if this macros is used. Not available for Transfer transactions.
<code>\${GOOGLE_PAY}</code>	n/a	Google Pay buy button will be available if this macros is used. Not available for Transfer transactions.
<code>\${CARDNO}</code>	<code>\${CARD-NO-VALUE}</code>	Payer's credit card number.
<code>\${EXPMONTH}</code>	n/a	Credit card expiration month.
<code>\${EXPYEAR}</code>	n/a	Credit card expiration year.
<code>\${CVV2}</code>	<code>\${CVV2VALUE}</code>	Card security code. Example: 432
<code>\${CARDHOLDER}</code>	<code>\${CARD-HOLDER-VALUE}</code>	Card printed name.
<code>\${MERCHANT}</code>	n/a	End point display name.
<code>\${SKIN_VERSION}</code>	n/a	CSS skin version.
<code>\${ORDERDESCRIPTION}</code>	n/a	Order description.
<code>\${CUSTOMER_FIRST_NAME}</code>	n/a	Payer first name sent by Connecting Party via input parameters.
<code>\${CUSTOMER_LAST_NAME}</code>	n/a	Payer last name sent by Connecting Party via input parameters.
<code>\${CUSTOMER_EMAIL}</code>	n/a	Payer E-mail address sent by Connecting Party via input parameters.
<code>\${DESTINATIONCARDNO}</code>	n/a	Destination credit card number. Available for Transfer transactions.
<code>\${DESTINATION_CARD_TYPE}</code>	n/a	Destination card type. Available for Transfer transactions.
<code>\${DESTINATION_LAST_FOUR_DIGITS}</code>	n/a	Destination card last four digits. Available for Transfer transactions.
<code>\${AMOUNT}</code>	n/a	Amount.
<code>\${CURRENCY}</code>	n/a	Currency.
<code>\${DESTINATION_PURPOSE}</code>	n/a	Purpose sent by Connecting Party via input parameters.
<code>\${PAYNET_ORDER_ID}</code>	n/a	Doc2.0 order id.
<code>\${MERCHANT_ORDER_ID}</code>	n/a	Connecting Party order id.
<code>\${refresh_interval}</code>	n/a	Refresh interval recommended by system.
<code>\${uuid}</code>	n/a	Internal.
<code>\${INTERNAL_SECTION}</code>	n/a	Internal for iFrame integration.
<code>\${CUSTOMER_IP_COUNTRY_ISO_CODE}</code>	n/a	Payer country defined by IP Address.
<code>\${PREFERRED_LANGUAGE}</code>	n/a	Payer language sent by Connecting Party via input parameters. Not available for Account Verification.
<code>\${BROWSER_LANGUAGE}</code>	n/a	Payer language defined by browser settings.

continues on next page

Table 6 – continued from previous page

Field Name Macro	Field Value Macro	Description
<code>\${CUSTOMER_LANGUAGE}</code>	n/a	Payer language send by Connecting Party via input parameters or defined by browser settings if first is not set.
<code>\${MERCHANT_FORM_DATA}</code>	n/a	Parameters sent in MERCHANT_FORM_DATA API parameter are parsed into macros with the same name, the parameter is url-encoded, example: test-param%3Dtest1%26mynewparam%3Dtest2 and is parsed into \$MFD_testparam = test1 and \$MFD_mynewparam = test2 macros in the form. Parameter name characters[a-zA-Z0-9], parameter value characters[a-zA-Z0-9], control characters [=&], 2MB max size. Not available for Account Verification. For example, this parameter can be used to display payment form in light/dark mode depending on the value passed by Connecting Party (e.g. pass merchant_form_data=theme%3Ddark in request and \$MFD_theme macro placeholder on payment form will be changed to dark.
<code>\${MIN_AMOUNT}</code>	n/a	This macro has the value provided in minimum-transaction-amount parameter in initial request. It can be used to validate the transaction amount before payment form is submitted. Contact support manager to enable this feature.
<code>\${MAX_AMOUNT}</code>	n/a	This macro has the value provided in maximum-transaction-amount parameter in initial request. It can be used to validate the transaction amount before payment form is submitted. Contact support manager to enable this feature.
<code>\${CUSTOMER_ZIP_CODE}</code>	n/a	Generates a ZIP code if it was not received from the Payer. Not available for Account Verification.

10.3.3 Prefilled Cardholder Data in Payment Page

Payer's cardholder data, used for processing of a previous transaction, can be tokenized for future use with request to [/api/v2/create-card-ref/](#). The obtained token cardrefid then can be passed in the next requests to relevant API. If this parameter is sent, all cardholder data input will be immutable and prefilled on the payment page. CVV is not tokenized and must be provided by Payer on this payment page. Below is the example of form customization to work with prefilled cardholder data in payment page.

Note: This feature is available only for [Sale](#) and [Preauth](#) transaction types and cannot be used in [Payment Cashier](#).

Prefilled Cardholder Data Payment Page Example

```
<html>
<head>
</head>
<body>
  <h3>Order #${MERCHANT_ORDER_ID} - ${ORDERDESCRIPTION}</h3>
  <h3>Total amount: ${AMOUNT} ${CURRENCY} to ${MERCHANT}</h3>
  <h4>Card Holder: ${CARDHOLDER} </h4>
  <h4>Card Number: ${CARD_BIN} ..... ${CARD_LAST4DIGITS} </h4>
  <h4>Exp year: ${EXPYEAR_VALUE} </h4>
  <h4>Exp month: ${EXPMONTH_VALUE} </h4>
  <form action="${ACTION}" method="post">
    <div>Cvv: <input name="${CVV2}" type="text" maxlength="4" autocomplete="off"/></div>
    <div>
      ${INTERNAL_SECTION}
      #if(${card_error})
        <div style="color: red;">${card_error}</div>
      #end
      <input name="submit" type="submit" value="Pay"/>
    </form>
  </body>
</html>
```

Prefilled Cardholder Data Payment Page Macros

Field Name Macro	Field Value Macro	Description
\${CARDHOLDER}	n/a	Cardholder name.
\${CARD_BIN}	n/a	Full card number.
\${CARD_LAST4DIGITS}	n/a	Last 4 digits of the card.
\${EXPMONTH_VALUE}	n/a	Card expiry month value.
\${EXPYEAR_VALUE}	n/a	Card expiry year value.
\${CVV2}	n/a	CVV2 card value.

10.3.4 Wait Page Customization

Wait Page Template

Wait Page is the form where the Payer's stays on until transaction reaches it's final status.

```
<html>  
<head>  
    <script type="text/javascript">  
        function fc(t) {  
            document.getElementById("seconds-remaining").innerHTML = t;  
            (t > 0) ? setTimeout(function(){fc(--t);}, 1000) : document.checkform.  
↵submit();}  
    </script>  
</head>  
<body onload="fc(!$refresh_interval)">  
    <h3>Order #${MERCHANT_ORDER_ID} - ${ORDERDESCRIPTION}</h3>  
    <h3>Total amount: ${AMOUNT} ${CURRENCY} to ${MERCHANT}</h3>  
    Please wait, your payment is being processed, remaining <span id="seconds-  
↵remaining">&nbsp;</span> seconds.  
    <form name="checkform" method="post">  
        <input type="hidden" name="tmp" value="${uuid}" />  
        ${INTERNAL_SECTION}  
        <input type="submit" value="Check" />  
    </form>  
</body>  
</html>
```

Note: The code described below do not regard to the [Account Verification](#):

Total amount: \$!AMOUNT \$!CURRENCY to \$!MERCHANT

Wait Page Template Macros

Field Name Macro	Field Value Macro	Description
\$(refresh_interval)	n/a	Refresh interval recommended by system.
\$(MERCHANT_ORDER_ID)	n/a	Connecting Party order id.
\$(ORDERDESCRIPTION)	n/a	Order description.
\$(AMOUNT)	n/a	Amount. Not available for Account verification.
\$(CURRENCY)	n/a	Currency. Not available for Account verification.
\$(MERCHANT)	n/a	End point display name.
\$(uuid)	n/a	Internal.
\$(INTERNAL_SECTION)	n/a	Internal for iFrame integration.

continues on next page

Table 8 – continued from previous page

Field Name Macro	Field Value Macro	Description
<code>\${SKIN_VERSION}</code>	n/a	CSS skin version.
<code>\${PAYNET_ORDER_ID}</code>	n/a	Doc2.0 order id.
<code>\${CUSTOMER_IP_COUNTRY_ISO_CODE}</code>	n/a	Payer country defined by IP Address.
<code>\${PREFERRED_LANGUAGE}</code>	n/a	Payer language send by Connecting Party via input parameters.
<code>\${BROWSER_LANGUAGE}</code>	n/a	Payer language defined by browser settings.
<code>\${CUSTOMER_LANGUAGE}</code>	n/a	Payer language send by Connecting Party via input parameters or defined by browser settings if first is not set.

10.3.5 Finish Page Customization

Finish Page Template

A form on which the Payer is shown the [final status](#) of his request.

```
<html>
  <head>
  </head>
  <body>
    <h3>Processing of the payment has finished</h3>
    <h3>Order Invoice: ${MERCHANT_ORDER_ID}</h3>
    <h3>Order ID: ${PAYNET_ORDER_ID}</h3>
    <h3>Status: ${STATUS}</h3>
    #if($ERROR_MESSAGE)
      <h3>Error: ${ERROR_MESSAGE}</h3>
    #end
  </body>
</html>
```

Finish Page Macros

Field Name Macro	Field Value Macro	Description
<code>\${STATUS}</code>	n/a	Order status.
<code>\${PAYNET_ORDER_ID}</code>	n/a	System order id.
<code>\${MERCHANT_ORDER_ID}</code>	n/a	Connecting Party order id.
<code>\${ERROR_MESSAGE}</code>	n/a	Contains the reason for decline or error details.
<code>\${SKIN_VERSION}</code>	n/a	CSS skin version.
<code>\${CUSTOMER_IP_COUNTRY_ISO_CODE}</code>	n/a	Payer country defined by IP Address.
<code>\${PREFERRED_LANGUAGE}</code>	n/a	Payer language send by Connecting Party via input parameters.
<code>\${BROWSER_LANGUAGE}</code>	n/a	Payer language defined by browser settings.

continues on next page

Table 9 – continued from previous page

Field Name Macro	Field Value Macro	Description
<code>\${CUSTOMER_LANGUAGE}</code>	n/a	Payer language send by Connecting Party via input parameters or defined by browser settings if first is not set.
<code>\${AMOUNT}</code>	n/a	Amount. Not available for Account verification.
<code>\${CURRENCY}</code>	n/a	Currency. Not available for Account verification.
<code>\${DESCRIPTION}</code>	n/a	Transaction description. Not available for Account verification.
<code>\${DATE}</code>	n/a	Transaction date. Not available for Account verification.
<code>\${PAYNET_PROCESSING_DATE}</code>	n/a	Doc2.0 processing date. Not available for Account verification.
<code>\${RRN}</code>	n/a	Retrieval Reference Number. Not available for Account verification.
<code>\${AUTH_CODE}</code>	n/a	Authorization Code. Not available for Account verification.
<code>\${CARD_TYPE}</code>	n/a	Card type. Not available for Account verification.
<code>\${LAST_FOUR_DIGITS}</code>	n/a	Last four digits of a card. Not available for Account verification.
<code>\${CUSTOMER_REDIRECT_URL}</code>	n/a	URL the cardholder will be redirected to upon completion of the transaction.

10.3.6 Payment Cashier Page Customization

Note: This page is relevant only for [Payment Cashier](#) integration.

Parallel Form Template Sample

Parallel Form is a form for the Payer to select one of the available payment methods to perform a transaction.

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
<!-- country selector -->
<span id="countryIsoCode" style="display: none;">${CUSTOMER_IP_COUNTRY_ISO_CODE}</span>
<div class="container">
  <input name="customer_language_code" id="iso" type="hidden" value="${CUSTOMER_
LANGUAGE}"/>
  <#if (!$HAS_PARALLEL_FORM_PAYMENT_METHODS)
    <select id="countrySelector" onchange="onCountrySelected(this);">
      <option value="EU">Europe</option>
  </select>
  </if>
</div>

```

(continues on next page)

(continued from previous page)

```

        <!-- You may add other countries for selection to this list -->
    </select>
#end
#if ($HAS_PARALLEL_FORM_PAYMENT_METHODS)
    <div class="form-tabs">
        <ul id="formTabs" class="form-tabs-ul">
            #foreach ($paymentMethod in $PARALLEL_FORM_PAYMENT_METHODS)
                <li id="$!{paymentMethod.identifier}-FORM-button" class="form-tab-
→button#if($velocityCount == 1) form-tab-button-active form-tab-button-active-$!
→{paymentMethod.identifier}#end" #if($paymentMethod.iframe) onclick="chooseIframe('$!
→{paymentMethod.identifier}');" #end>
                    $!{paymentMethod.name}
                </li>
            #end
        </ul>
        <select id="countrySelector" onchange="onCountrySelected(this);">
            <option value="EU">Europe</option>
            <!-- You may add other countries for selection to this list -->
        </select>
    </div>
    #foreach ($paymentMethod in $PARALLEL_FORM_PAYMENT_METHODS)
        <div class="ssl-header ssl-header-$!{paymentMethod.identifier}" id="merchant-
→description-$!{paymentMethod.identifier}" #if ($velocityCount > 1) style="display:
→none;"#end>
            <span class="ssl-text">${MERCHANT}</span>
            <div>
                <span class="flag-EN" id="flag-$!{paymentMethod.identifier}"></span>
                <select id="langSelector-$!{paymentMethod.identifier}" class=
→"langSelector langSelector-$!{paymentMethod.identifier}" onchange="selectLang('$!
→{paymentMethod.identifier}');">
                    <option value="EN">English</option>
                    <!-- You may add other languages for selection to this list -->
                </select>
            </div>
        </div>
        <script>
            merchantDescriptionIds.push("merchant-description-$!{paymentMethod.
→identifier}");
            paymentMethodsCountries["$!{paymentMethod.identifier}" ] = "$!
→{paymentMethod.countries}";
        </script>
    #end
#end
#if ($HAS_PARALLEL_FORM_PAYMENT_METHODS)
    #foreach ($paymentMethod in $PARALLEL_FORM_PAYMENT_METHODS)
        #if ($paymentMethod.iframe)
            <div class="frame-container-$!{paymentMethod.identifier} payment-
→method-iframe" id="$!{paymentMethod.identifier}-FORM" pneopen="#if ($velocityCount
→== 1)true#{else} false#end" pnesrc="$!paymentMethod.initSessionUrl" #if (
→$velocityCount > 1) style="display: none;"#end>
                <iframe src="#if ($velocityCount == 1)$!paymentMethod.
→initSessionUrl#end" scrolling="#if $!{paymentMethod.identifier} == 'QIWI' || $!
→{paymentMethod.identifier} == 'ASTROPAY')yes#{else}no#end">
                    </iframe>
                </div>
            #end
        #end
    #end

```

(continues on next page)

(continued from previous page)

```

#end
<form id="waitForm" action="${WAIT_FORM_URL}">
  #foreach( $p in $WAIT_FORM_PARAMETERS )
    <input type="hidden" name="$p.key" value="${p.value}">
  #end
  <!-- ${INTERNAL_SECTION} -->
</form>
#end
</div>
</body>
</html>

```

For more information see [Country Codes](#) and [Language Codes](#).

Parallel Form Macros

Field Name Macro	Field Value Macro	Description
\${CUSTOMER_IP_COUNTRY_ISO_CODE}	n/a	Payer's country defined by IP Address.
\${CUSTOMER_COUNTRY_CODE}	n/a	Billing country of customer sent in incoming API request.
\${CUSTOMER_LANGUAGE}	n/a	Payer language sent by Connecting Party via input parameters or defined by browser settings if first is not set.
\${HAS_PARALLEL_FORM_PAYMENT_METHODS}	n/a	Determines the availability of parallel form payment methods.
\$paymentMethod	n/a	Payment method object.
\${PARALLEL_FORM_PAYMENT_METHODS}	n/a	Array with available parallel form payment methods.
\${paymentMethod.identifier}	n/a	Payment method identifier.
\$velocityCount	n/a	Sequence number of the payment method.
\$paymentMethod.iframe	n/a	Iframe of payment method.
\${paymentMethod.name}	n/a	Payment method name.
\${MERCHANT}	n/a	End point display name.
\${paymentMethod.countries}	n/a	Countries which are set for this payment method.
\${paymentMethod.initSessionUrl}	n/a	URL for payment method session initialization.
\${WAIT_FORM_URL}	n/a	Waiting form URL.
\$p	n/a	Waiting form parameter.
WAIT_FORM_PARAMETERS	n/a	Waiting form parameters array.
\$p.key	n/a	Waiting form parameter key.
\${p.value}	n/a	Waiting form parameter value.
\${INTERNAL_SECTION}	n/a	Internal for iFrame integration.

Parallel Form Scripts

The following Parallel Form scripts are required for it's operating. The first script should be added into <head> tag of html document:

```
<script type="text/javascript">
var merchantDescriptionIds = [];
var paymentMethodsCountries = {};
function isCCValid(r) {
    var n = r.length;
    if (n > 19 || 13 > n) return !1;
    for (i = 0, s = 0, m = 1, l = n; i < l; i++) d = parseInt(r.substring(l - i - 1,
    ↪ l - i), 10) * m, s += d >= 10 ? d % 10 + 1 : d, 1 == m ? m++ : m--;
    return s % 10 == 0 ? !0 : !1
}

function runPayment(t) {
    if (isCCValid(t)) {
        return !0;
    } else {
        document.getElementById('cardnumber').style.borderColor = '#fb860f';
        return !1;
    }
}

function chooseIframe(paymentMethodName) {
    var id = paymentMethodName + '-FORM';
    var iframes = document.getElementsByTagName('IFRAME');
    for (var i = 0; i < iframes.length; i++) {
        var iframe = iframes[i].parentNode;
        var isCurrent = iframe.id == id;
        iframe.style.display = isCurrent ? 'block' : 'none';
        if (isCurrent && iframe.getAttribute("pneopen") == 'false') {
            ↪ iframe.firstElementChild.src = iframe.getAttribute("pnesrc") + "?country="
            ↪ " + document.getElementById("countryIsoCode").innerText;
            iframe.setAttribute("pneopen", 'true');
        }
        var buttonElem = document.getElementById(iframe.id + '-button');
        buttonElem.className = (iframe.id == id) ? 'form-tab-button form-tab-button-
        ↪ active form-tab-button-active-' + paymentMethodName : 'form-tab-button';
    }
    var descriptionId = 'merchant-description-' + paymentMethodName;
    for (var i = 0; i < merchantDescriptionIds.length; i++) {
        var isCurrent = merchantDescriptionIds[i] == descriptionId;
        ↪ document.getElementById(merchantDescriptionIds[i]).style.display = isCurrent ?
        ↪ 'flex' : 'none';
    }
}

function pneInit() {
    window.pneMasterSessionProcessed = function() {
        document.getElementById('waitForm').submit()
    }
    foreach( $paymentMethod in $PARALLEL_FORM_PAYMENT_METHODS )
        if( $paymentMethod.default ) chooseIframe('${paymentMethod.identifier}-FORM');
        break
end
```

(continues on next page)

(continued from previous page)

```

    end
}
</script>

```

Second script should be added to html code before the </body> close tag.

```

<script type="text/javascript">
function updateTabsVisibility() {
    var countryCode = getSelectedCountryCode();
    var paymentMethods = getPaymentMethods();
    var someIframeShown = false;
    for (var i = 0; i < paymentMethods.length; i++) {
        var paymentMethod = paymentMethods[i];
        var visible = isPaymentMethodVisible(paymentMethod, countryCode);
        updateTabVisibility(paymentMethod, visible);
        if (visible && !someIframeShown) {
            chooseIframe(paymentMethod);
            someIframeShown = true;
        }
    }
}

function getSelectedCountryCode() {
    return document.getElementById('countryIsoCode').innerText;
}

function setSelectedCountryCode(countryCode) {
    document.getElementById('countryIsoCode').innerText = countryCode;
}

function getPaymentMethods() {
    var lis = document.getElementById("formTabs").children;
    var result = [];
    for (var i = 0; i < lis.length; i++) {
        var li = lis[i];
        var id = li.id;
        if (id != null) {
            result.push(id.substring(0, id.indexOf("-FORM-button")));
        }
    }
    return result;
}

function isPaymentMethodVisible(paymentMethodId, countryCode) {
    if (paymentMethodsCountries[paymentMethodId] == "") {
        return true;
    } else {
        return paymentMethodsCountries[paymentMethodId].indexOf(countryCode) > -1;
    }
}

function updateTabVisibility(paymentMethod, visible) {
    document.getElementById(paymentMethod + '-FORM-button').style.display = visible ?
    'inline-block' : 'none';
}

```

(continues on next page)

(continued from previous page)

```

function onCountrySelected(selector) {
    setSelectedCountryCode(selector.value);
    updateTabsVisibility();
}

document.addEventListener('DOMContentLoaded', function() {
    pneInit();
    updateTabsVisibility();
    document.getElementById('countrySelector').value = getSelectedCountryCode();
});

function syncLangSelectors(ISO) {
    var methods = getPaymentMethods();
    for (var i = 0; i < methods.length; i++) {
        document.getElementById('flag-' + methods[i]).className = "flag-" + ISO;
        document.getElementById('langSelector-' + methods[i]).value = ISO;
    }
}

var ISO = document.getElementById('iso').value.toUpperCase();
ISO = 'EN';
document.getElementById('iso').value = ISO;
syncLangSelectors(ISO);

document.addEventListener('DOMContentLoaded', function() {
    //l10n();
});

function selectLang(method) {
    var ISO = document.getElementById('langSelector-' + method).value.toUpperCase();
    document.getElementById('iso').value = ISO;

    syncLangSelectors(ISO);
    //l10n();
}
</script>

```

If needed to determine the Payer billing country on the data sent via the API, \$!CUSTOMER_COUNTRY_CODE is used. If needed to determine the Payer billing country on the IP address from the Payer browser, use \$!CUSTOMER_IP_COUNTRY_ISO_CODE, billing country will also change according to IP.

The master set up by Billing country:

```

<span id="countryIsoCode" style="display: none;">$!CUSTOMER_COUNTRY_CODE</span>
<input name="customer_language_code" id="iso" type="hidden" value="$CUSTOMER_LANGUAGE
↵"/>
<div class="country-select">

```

10.4 Country and State Codes

- [Country Codes](#)
- [Mandatory State Codes](#)
- [Other State Codes](#)

10.4.1 Country Codes

- Alpha-2 country codes are used in country field.
- Alpha-3 codes are returned in status response and callback in card-country-alpha-three-code and destination-card-country-alpha-three-code parameters.
- ISO Codes are used in Payment Gateway configurations.

Country	Alpha-2 Code	Alpha-3 Code	ISO Code
Åland Islands	AX	ALA	248
Afghanistan	AF	AFG	004
Albania	AL	ALB	008
Algeria	DZ	DZA	012
American Samoa	AS	ASM	016
Andorra	AD	AND	020
Angola	AO	AGO	024
Anguilla	AI	AIA	660
Antarctica	AQ	ATA	010
Antigua and Barbuda	AG	ATG	028
Argentina	AR	ARG	032
Armenia	AM	ARM	051
Aruba	AW	ABW	533
Australia	AU	AUS	036
Austria	AT	AUT	040
Azerbaijan	AZ	AZE	031
Bahamas	BS	BHS	044
Bahrain	BH	BHR	048
Bangladesh	BD	BGD	050
Barbados	BB	BRB	052
Belarus	BY	BLR	112
Belgium	BE	BEL	056
Belize	BZ	BLZ	084
Benin	BJ	BEN	204
Bermuda	BM	BMU	060
Bhutan	BT	BTN	064
Bolivia, Plurinational State of	BO	BOL	068
Bosnia and Herzegovina	BA	BIH	070
Botswana	BW	BWA	072
Bouvet Island	BV	BVT	074
Brazil	BR	BRA	076
British Indian Ocean Territory	IO	IOT	086

continues on next page

Table 11 – continued from previous page

Country	Alpha-2 Code	Alpha-3 Code	ISO Code
Brunei Darussalam	BN	BRN	096
Bulgaria	BG	BGR	100
Burkina Faso	BF	BFA	854
Burundi	BI	BDI	108
Cambodia	KH	KHM	116
Cameroon	CM	CMR	120
Canada	CA	CAN	124
Cabo Verde	CV	CPV	132
Cayman Islands	KY	CYM	136
Central African Republic	CF	CAF	140
Chad	TD	TCD	148
Chile	CL	CHL	152
China	CN	CHN	156
Christmas Island	CX	CXR	162
Cocos (Keeling) Islands	CC	CCK	166
Colombia	CO	COL	170
Comoros	KM	COM	174
Congo	CG	COG	178
Congo, the Democratic Republic of the	CD	COD	180
Cook Islands	CK	COK	184
Costa Rica	CR	CRI	188
Cote d'Ivoire	CI	CIV	384
Croatia	HR	HRV	191
Cuba	CU	CUB	192
Cyprus	CY	CYP	196
Czech Republic	CZ	CZE	203
Denmark	DK	DNK	208
Djibouti	DJ	DJI	262
Dominica	DM	DMA	212
Dominican Republic	DO	DOM	214
Ecuador	EC	ECU	218
Egypt	EG	EGY	818
El Salvador	SV	SLV	222
Equatorial Guinea	GQ	GNQ	226
Eritrea	ER	ERI	232
Estonia	EE	EST	233
Ethiopia	ET	ETH	231
Falkland Islands (Malvinas)	FK	FLK	238
Faroe Islands	FO	FRO	234
Fiji	FJ	FJI	242
Finland	FI	FIN	246
France	FR	FRA	250
French Guiana	GF	GUF	254
French Polynesia	PF	PYF	258
French Southern Territories	TF	ATF	260
Gabon	GA	GAB	266
Gambia	GM	GMB	270
Georgia	GE	GEO	268

continues on next page

Table 11 – continued from previous page

Country	Alpha-2 Code	Alpha-3 Code	ISO Code
Germany	DE	DEU	276
Ghana	GH	GHA	288
Gibraltar	GI	GIB	292
Greece	GR	GRC	300
Greenland	GL	GRL	304
Grenada	GD	GRD	308
Guadeloupe	GP	GLP	312
Guam	GU	GUM	316
Guatemala	GT	GTM	320
Guernsey	GG	GGY	831
Guinea	GN	GIN	324
Guinea-Bissau	GW	GNB	624
Guyana	GY	GUY	328
Haiti	HT	HTI	332
Heard Island and McDonald Is-lands	HM	HMD	334
Holy See (Vatican City State)	VA	VAT	336
Honduras	HN	HND	340
Hong Kong	HK	HKG	344
Hungary	HU	HUN	348
Iceland	IS	ISL	352
India	IN	IND	356
Indonesia	ID	IDN	360
Iran, Islamic Republic of	IR	IRN	364
Iraq	IQ	IRQ	368
Ireland	IE	IRL	372
Isle of Man	IM	IMN	833
Israel	IL	ISR	376
Italy	IT	ITA	380
Jamaica	JM	JAM	388
Japan	JP	JPN	392
Jersey	JE	JEY	832
Jordan	JO	JOR	400
Kazakhstan	KZ	KAZ	398
Kenya	KE	KEN	404
Kosovo	QZ	QZZ	900
Kiribati	KI	KIR	296
Korea, Democratic People's Repub-lic of	KP	PRK	408
Korea, Republic of	KR	KOR	410
Kuwait	KW	KWT	414
Kyrgyzstan	KG	KGZ	417
Lao People's Democratic Republic	LA	LAO	418
Latvia	LV	LVA	428
Lebanon	LB	LBN	422
Lesotho	LS	LSO	426
Liberia	LR	LBR	430
Libya	LY	LBY	434
Liechtenstein	LI	LIE	438

continues on next page

Table 11 – continued from previous page

Country	Alpha-2 Code	Alpha-3 Code	ISO Code
Lithuania	LT	LTU	440
Luxembourg	LU	LUX	442
Macao	MO	MAC	446
Macedonia, the former Yugoslav Republic of	MK	MKD	807
Madagascar	MG	MDG	450
Malawi	MW	MWI	454
Malaysia	MY	MYS	458
Maldives	MV	MDV	462
Mali	ML	MLI	466
Malta	MT	MLT	470
Marshall Islands	MH	MHL	584
Martinique	MQ	MTQ	474
Mauritania	MR	MRT	478
Mauritius	MU	MUS	480
Mayotte	YT	MYT	175
Mexico	MX	MEX	484
Micronesia, Federated States of	FM	FSM	583
Moldova, Republic of	MD	MDA	498
Monaco	MC	MCO	492
Mongolia	MN	MNG	496
Montenegro	ME	MNE	499
Montserrat	MS	MSR	500
Morocco	MA	MAR	504
Mozambique	MZ	MOZ	508
Myanmar	MM	MMR	104
Namibia	NA	NAM	516
Nauru	NR	NRU	520
Nepal	NP	NPL	524
Netherlands	NL	NLD	528
Netherlands Antilles	AN	ANT	530
New Caledonia	NC	NCL	540
New Zealand	NZ	NZL	554
Nicaragua	NI	NIC	558
Niger	NE	NER	562
Nigeria	NG	NGA	566
Niue	NU	NIU	570
Norfolk Island	NF	NFK	574
Northern Mariana Islands	MP	MNP	580
Norway	NO	NOR	578
Oman	OM	OMN	512
Pakistan	PK	PAK	586
Palau	PW	PLW	585
Palestine, State of	PS	PSE	275
Panama	PA	PAN	591
Papua New Guinea	PG	PNG	598
Paraguay	PY	PRY	600
Peru	PE	PER	604
Philippines	PH	PHL	608

continues on next page

Table 11 – continued from previous page

Country	Alpha-2 Code	Alpha-3 Code	ISO Code
Pitcairn	PN	PCN	612
Poland	PL	POL	616
Portugal	PT	PRT	620
Puerto Rico	PR	PRI	630
Qatar	QA	QAT	634
Reunion	RE	REU	638
Romania	RO	ROU	642
Russian Federation	RU	RUS	643
Rwanda	RW	RWA	646
Saint Barthelemy	BL	BLM	652
Saint Helena, Ascension and Tristan da Cunha	SH	SHN	654
Saint Kitts and Nevis	KN	KNA	659
Saint Lucia	LC	LCA	662
Saint Martin (French part)	MF	MAF	663
Saint Pierre and Miquelon	PM	SPM	666
Saint Vincent and the Grenadines	VC	VCT	670
Samoa	WS	WSM	882
San Marino	SM	SMR	674
Sao Tome and Principe	ST	STP	678
Saudi Arabia	SA	SAU	682
Senegal	SN	SEN	686
Serbia	RS	SRB	688
Seychelles	SC	SYC	690
Sierra Leone	SL	SLE	694
Singapore	SG	SGP	702
Slovakia	SK	SVK	703
Slovenia	SI	SVN	705
Solomon Islands	SB	SLB	090
Somalia	SO	SOM	706
South Africa	ZA	ZAF	710
South Georgia and the South Sandwich Islands	GS	SGS	239
Spain	ES	ESP	724
Sri Lanka	LK	LKA	144
Sudan	SD	SDN	729
Suriname	SR	SUR	740
Svalbard and Jan Mayen	SJ	SJM	744
Swaziland	SZ	SWZ	748
Sweden	SE	SWE	752
Switzerland	CH	CHE	756
Syrian Arab Republic	SY	SYR	760
Taiwan, Province of China	TW	TWN	158
Tajikistan	TJ	TJK	762
Tanzania, United Republic of	TZ	TZA	834
Thailand	TH	THA	764
Timor-Leste	TL	TLS	626
Togo	TG	TGO	768
Tokelau	TK	TKL	772

continues on next page

Table 11 – continued from previous page

Country	Alpha-2 Code	Alpha-3 Code	ISO Code
Tonga	TO	TON	776
Trinidad and Tobago	TT	TTO	780
Tunisia	TN	TUN	788
Turkey	TR	TUR	792
Turkmenistan	TM	TKM	795
Turks and Caicos Islands	TC	TCA	796
Tuvalu	TV	TUV	798
Uganda	UG	UGX	800
Ukraine	UA	UKR	804
United Arab Emirates	AE	ARE	784
United Kingdom	GB	GBR	826
United States	US	USA	840
United States Minor Outlying Islands	UM	UMI	581
Uruguay	UY	URY	858
Uzbekistan	UZ	UZB	860
Vanuatu	VU	VUT	548
Venezuela, Bolivarian Republic of	VE	VEN	862
Viet Nam	VN	VNM	704
Virgin Islands, British	VG	VGB	092
Virgin Islands, U.S.	VI	VIR	850
Wallis and Futuna	WF	WLF	876
Western Sahara	EH	ESH	732
Yemen	YE	YEM	887
Zambia	ZM	ZMB	894
Zimbabwe	ZW	ZWE	716
Sint Maarten (Dutch part)	SX	SXM	534
Curaçao	CW	CUW	531
Bonaire, Sint Eustatius and Saba	BQ	BES	535
South Sudan	SS	SSD	728

10.4.2 Mandatory State Codes

Mandatory state codes used in state field for USA, Canada and Australia.

Country	State Code	State Name
AU	ACT	Australian Capital Territory
AU	NSW	New South Wales
AU	NT	Northern Territory
AU	QLD	Queensland
AU	SA	South Australia
AU	TAS	Tasmania
AU	VIC	Victoria
AU	WA	Western Australia
CA	AB	Alberta
CA	BC	British Columbia
CA	MB	Manitoba
CA	NB	New Brunswick

continues on next page

Table 12 – continued from previous page

Country	State Code	State Name
CA	NL	Newfoundland and Labrador
CA	NT	Northwest Territories
CA	NS	Nova Scotia
CA	NU	Nunavut
CA	ON	Ontario
CA	PE	Prince Edward Island
CA	QC	Quebec
CA	SK	Saskatchewan
CA	YT	Yukon
US	AL	Alabama
US	AK	Alaska
US	AS	American Samoa
US	AZ	Arizona
US	AR	Arkansas
US	CA	California
US	CO	Colorado
US	CT	Connecticut
US	DE	Delaware
US	DC	District of Columbia
US	FL	Florida
US	GA	Georgia
US	GU	Guam
US	HI	Hawaii
US	ID	Idaho
US	IL	Illinois
US	IN	Indiana
US	IA	Iowa
US	KS	Kansas
US	KY	Kentucky
US	LA	Louisiana
US	ME	Maine
US	MD	Maryland
US	MA	Massachusetts
US	MI	Michigan
US	MN	Minnesota
US	MS	Mississippi
US	MO	Missouri
US	MT	Montana
US	NE	Nebraska
US	NV	Nevada
US	NH	New Hampshire
US	NJ	New Jersey
US	NM	New Mexico
US	NY	New York
US	NC	North Carolina
US	ND	North Dakota
US	OH	Ohio
US	OK	Oklahoma

continues on next page

Table 12 – continued from previous page

Country	State Code	State Name
US	OR	Oregon
US	PA	Pennsylvania
US	PR	Puerto Rico
US	RI	Rhode Island
US	SC	South Carolina
US	SD	South Dakota
US	TN	Tennessee
US	TX	Texas
US	UT	Utah
US	VT	Vermont
US	VI	Virgin Islands
US	VA	Virginia
US	WA	Washington
US	WV	West Virginia
US	WI	Wisconsin
US	WY	Wyoming

10.4.3 Other State Codes

Optional state codes used in state field. Depending on the processing channel these codes can also be mandatory.

Country	ISO 3166-2 Code	State Name
AF	BDS	Badakhshan
AF	BDG	Badghis
AF	BGL	Baghlan
AF	BAL	Balkh
AF	BAM	Bamian
AF	FRA	Farah
AF	FYB	Faryab
AF	GHA	Ghazni
AF	GHO	Ghowr
AF	HEL	Helmand
AF	HER	Herat
AF	JOW	Jowzjan
AF	KAB	Kabul
AF	KAN	Kandahar
AF	KAP	Kapisa
AF	KHO	Khost
AF	KNR	Konar
AF	KDZ	Kondoz
AF	LAG	Laghman
AF	NAN	Nangrahar
AF	NIM	Nimruz
AF	NUR	Nurestan
AF	PIA	Paktia
AF	PKA	Paktika
AF	PAR	Parwan

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
AF	SAM	Samangan
AF	SAR	Sar-e Pol
AF	TAK	Takhar
AF	WAR	Wardak
AF	DAY	Dāykundī
AF	LOG	Lōgar
AF	PAN	Panjshayr
AF	URU	Uruzgān
AF	ZAB	Zābul
AL	01	Berat
AL	09	Diber
AL	02	Durres
AL	03	Elbasan
AL	04	Fier
AL	05	Gjirokaster
AL	06	Korce
AL	07	Kukes
AL	08	Lezhe
AL	10	Shkoder
AL	11	Tirane
AL	12	Vlore
DZ	01	Adrar
DZ	44	Ain Defla
DZ	46	Ain Temouchent
DZ	16	Alger
DZ	23	Annaba
DZ	05	Batna
DZ	08	Bechar
DZ	06	Bejaia
DZ	07	Biskra
DZ	09	Blida
DZ	34	Bordj Bou Arreridj
DZ	10	Bouira
DZ	35	Boumerdes
DZ	02	Chlef
DZ	25	Constantine
DZ	17	Djelfa
DZ	32	El Bayadh
DZ	39	El Oued
DZ	36	El Tarf
DZ	47	Ghardaia
DZ	24	Guelma
DZ	33	Illizi
DZ	18	Jijel
DZ	40	Khenchela
DZ	03	Laghouat
DZ	28	M'Sila
DZ	26	Medea
DZ	43	Mila

continues on next page

Table 13 - continued from previous page

Country	ISO 3166-2 Code	State Name
DZ	27	Mostaganem
DZ	45	Naama
DZ	31	Oran
DZ	30	Ouargla
DZ	48	Relizane
DZ	20	Saida
DZ	19	Setif
DZ	22	Sidi Bel Abbes
DZ	21	Skikda
DZ	41	Souk Ahras
DZ	12	Tebessa
DZ	14	Tiaret
DZ	37	Tindouf
DZ	42	Tipaza
DZ	38	Tissemsilt
DZ	15	Tizi Ouzou
DZ	13	Tlemcen
DZ	04	Oum el Bouaghi
DZ	11	Tamanrasset
DZ	29	Mascara
AD	07	Andorra la Vella
AD	02	Canillo
AD	03	Encamp
AD	08	Escaldes-Engordany
AD	04	La Massana
AD	05	Ordino
AD	06	Sant Julia de Loria
AO	BGO	Bengo
AO	BGU	Benguela
AO	BIE	Bie
AO	CAB	Cabinda
AO	CCU	Cuando-Cubango
AO	CNO	Cuanza Norte
AO	CUS	Cuanza Sul
AO	CNN	Cunene
AO	HUA	Huambo
AO	HUI	Huila
AO	LUA	Luanda
AO	LNO	Lunda Norte
AO	LSU	Lunda Sul
AO	MAL	Malange
AO	MOX	Moxico
AO	NAM	Namibe
AO	UIG	Uige
AO	ZAI	Zaire
AG	10	Barbuda
AG	11	Redonda
AG	03	Saint George
AG	04	Saint John

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
AG	05	Saint Mary
AG	06	Saint Paul
AG	07	Saint Peter
AG	08	Saint Philip
AR	B	Buenos Aires
AR	K	Catamarca
AR	H	Chaco
AR	U	Chubut
AR	X	Cordoba
AR	W	Corrientes
AR	E	Entre Rios
AR	P	Formosa
AR	Y	Jujuy
AR	L	La Pampa
AR	F	La Rioja
AR	M	Mendoza
AR	N	Misiones
AR	Q	Neuquen
AR	R	Rio Negro
AR	A	Salta
AR	J	San Juan
AR	D	San Luis
AR	Z	Santa Cruz
AR	S	Santa Fe
AR	G	Santiago del Estero
AR	V	Tierra del Fuego
AR	T	Tucuman
AR	C	Ciudad Autónoma de Buenos Aires
AM	AR	Ararat
AM	AV	Armavir
AM	KT	Kotayk'
AM	SU	Syunik'
AM	AG	Aragacotn
AM	ER	Erevan
AM	GR	Gegark'unik'
AM	LO	Loři
AM	SH	Širak
AM	TV	Tavuš
AM	VD	Vayoc Jor
AT	1	Burgenland
AT	3	Niederösterreich
AT	5	Salzburg
AT	6	Steiermark
AT	7	Tirol
AT	8	Vorarlberg
AT	9	Wien
AT	2	Kärnten
AT	4	Oberösterreich

continues on next page

Table 13 - continued from previous page

Country	ISO 3166-2 Code	State Name
AZ	ABS	Abseron
AZ	AGC	AgcabAdi
AZ	AGM	Agdam
AZ	AGS	Agdas
AZ	AGA	Agstafa
AZ	AGU	Agsu
AZ	AST	Astara
AZ	BAR	BArDA
AZ	BAB	BabAk
AZ	BA	Baki
AZ	BAL	BalakAn
AZ	BEY	Beylaqan
AZ	BIL	BilasuvAr
AZ	CAB	Cabrayil
AZ	CAL	Calilabab
AZ	CUL	Culfa
AZ	DAS	Daskasan
AZ	FUZ	Fuzuli
AZ	GAD	Gadabay
AZ	GA	Ganca
AZ	GOR	Goranboy
AZ	GOY	Goycay
AZ	HAC	Haciqabul
AZ	IMI	Imisli
AZ	ISM	Ismayilli
AZ	KAL	Kalbacar
AZ	KUR	Kurdamir
AZ	LAC	Lacin
AZ	LA	Lankaran
AZ	LA	Lankaran
AZ	LER	Lerik
AZ	MAS	Masalli
AZ	MI	Mingacevir
AZ	NA	Naftalan
AZ	NX	Naxcivan
AZ	NEF	Neftcala
AZ	OGU	Oguz
AZ	ORD	Ordubad
AZ	QAB	Qabala
AZ	QAX	Qax
AZ	QAZ	Qazax
AZ	QOB	Qobustan
AZ	QBA	Quba
AZ	QBI	Qubadli
AZ	QUS	Qusar
AZ	SAT	Saatli
AZ	SAB	Sabirabad
AZ	SAD	Sadarak
AZ	SAH	Sahbuz

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
AZ	SA	Saki
AZ	SA	Saki
AZ	SAL	Salyan
AZ	SMI	Samaxi
AZ	SKR	Samkir
AZ	SMX	Samux
AZ	SAR	Sarur
AZ	SIY	Siyazan
AZ	SM	Sumqayit
AZ	SUS	Susa
AZ	SUS	Susa
AZ	TAR	Tartar
AZ	TOV	Tovuz
AZ	UCA	Ucar
AZ	XAC	Xacmaz
AZ	XA	Xankandi
AZ	XIZ	Xizi
AZ	XCI	Xocali
AZ	XVD	Xocavand
AZ	YAR	Yardimli
AZ	YE	Yevlax
AZ	ZAN	Zangilan
AZ	ZAQ	Zaqatala
AZ	ZAR	Zardab
AZ	GYG	Göygöl
AZ	KAN	Kəngərli
AZ	SBN	Şabran
AZ	SR	Şirvan
BS	AK	Acklins
BS	BY	Berry Islands
BS	BI	Bimini
BS	BP	Black Point
BS	CI	Cat Island
BS	CO	Central Abaco
BS	CS	Central Andros
BS	CE	Central Eleuthera
BS	FP	City of Freeport
BS	EG	East Grand Bahama
BS	EX	Exuma
BS	GC	Grand Cay
BS	HI	Harbour Island
BS	HT	Hope Town
BS	IN	Inagua
BS	LI	Long Island
BS	MC	Mangrove Cay
BS	MG	Mayaguana
BS	MI	Moore's Island
BS	NO	North Abaco
BS	NS	North Andros

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
BS	NE	North Eleuthera
BS	RI	Ragged Island
BS	RC	Rum Cay
BS	SS	San Salvador
BS	SO	South Abaco
BS	SA	South Andros
BS	SE	South Eleuthera
BS	SW	Spanish Wells
BS	WG	West Grand Bahama
BS	CK	Crooked Island and Long Cay
BS	NP	New Providence
BH	13	Al 'Āsimah
BH	14	Al Janūbiyah
BH	15	Al Muḥarraḡ
BH	17	Ash Shamāliyah
BD	C	Dhaka
BD	D	Khulna
BD	E	Rajshahi
BD	G	Sylhet
BD	01	Bandarban
BD	02	Barguna
BD	03	Bogura
BD	04	Brahmanbaria
BD	05	Bagerhat
BD	06	Barishal
BD	07	Bhola
BD	08	Cumilla
BD	09	Chandpur
BD	10	Chattogram
BD	11	Cox's Bazar
BD	12	Chuadanga
BD	14	Dinajpur
BD	15	Faridpur
BD	16	Feni
BD	17	Gopalganj
BD	18	Gazipur
BD	19	Gaibandha
BD	20	Habiganj
BD	21	Jamalpur
BD	22	Jashore
BD	23	Jhenaidah
BD	24	Joypurhat
BD	25	Jhalakathi
BD	26	Kishoreganj
BD	28	Kurigram
BD	29	Khagrachhari
BD	30	Kushtia
BD	31	Lakshmipur

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
BD	32	Lalmonirhat
BD	33	Manikganj
BD	34	Mymensingh
BD	35	Munshiganj
BD	36	Madaripur
BD	37	Magura
BD	38	Moulvibazar
BD	39	Meherpur
BD	40	Narayanganj
BD	41	Netrakona
BD	42	Narsingdi
BD	43	Narail
BD	44	Natore
BD	45	Chapai Nawabganj
BD	46	Nilphamari
BD	47	Noakhali
BD	48	Naogaon
BD	49	Pabna
BD	50	Pirojpur
BD	51	Patuakhali
BD	52	Panchagarh
BD	53	Rajbari
BD	55	Rangpur
BD	56	Rangamati
BD	57	Sherpur
BD	58	Satkhira
BD	59	Sirajganj
BD	61	Sunamganj
BD	62	Shariatpur
BD	63	Tangail
BD	64	Thakurgaon
BD	A	Barishal
BD	B	Chattogram
BD	F	Rangpur
BD	H	Mymensingh
BB	01	Christ Church
BB	02	Saint Andrew
BB	03	Saint George
BB	04	Saint James
BB	05	Saint John
BB	06	Saint Joseph
BB	07	Saint Lucy
BB	08	Saint Michael
BB	09	Saint Peter
BB	10	Saint Philip
BB	11	Saint Thomas
BY	BR	Brestskaya (Brest)
BY	HO	Homyel'skaya (Homyel')
BY	HM	Horad Minsk

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
BY	HR	Hrodzyenskaya (Hrodna)
BY	MA	Mahilyowskaya (Mahilyow)
BY	MI	Minskaya
BY	VI	Vitsyebskaya (Vitsyebsk)
BE	VAN	Antwerpen
BE	WBR	Brabant Wallon
BE	WHT	Hainaut
BE	WLG	Liege
BE	VLI	Limburg
BE	WLX	Luxembourg
BE	WNA	Namur
BE	VOV	Oost-Vlaanderen
BE	VBR	Vlaams Brabant
BE	VWV	West-Vlaanderen
BE	BRU	“Bruxelles-Capitale, Région de”
BE	VLG	Vlaams Gewest
BE	WAL	“wallonne, Région”
BZ	BZ	Belize
BZ	CY	Cayo
BZ	CZL	Corozal
BZ	OW	Orange Walk
BZ	SC	Stann Creek
BZ	TOL	Toledo
BJ	AL	Alibori
BJ	AK	Atakora
BJ	AQ	Atlantique
BJ	BO	Borgou
BJ	CO	Collines
BJ	DO	Donga
BJ	KO	Kouffo
BJ	LI	Littoral
BJ	MO	Mono
BJ	OU	Oueme
BJ	PL	Plateau
BJ	ZO	Zou
BT	33	Bumthang
BT	22	Dagana
BT	GA	Gasa
BT	13	Haa
BT	11	Paro
BT	23	Punakha
BT	45	Samdrup Jongkhar
BT	14	Samtse
BT	31	Sarpang
BT	15	Thimphu
BT	41	Trashigang
BT	32	Trongsa
BT	21	Tsirang

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
BT	24	Wangdue Phodrang
BT	34	Zhemgang
BT	12	Chhukha
BT	42	Monggar
BT	43	Pema Gatshel
BT	44	Lhuentse
BT	TY	Trashi Yangtse
BO	H	Chuquisaca
BO	C	Cochabamba
BO	L	La Paz
BO	O	Oruro
BO	N	Pando
BO	P	Potosi
BO	S	Santa Cruz
BO	T	Tarija
BO	B	El Beni
BA	BIH	Federacija Bosne i Hercegovine
BA	BRC	Brčko distrikt
BA	SRP	Republika Srpska
BW	CE	Central
BW	GH	Ghanzi
BW	KG	Kgalagadi
BW	KL	Kgatleng
BW	KW	Kweneng
BW	NE	North East
BW	NW	North West
BW	SE	South East
BW	SO	Southern
BW	CH	Chobe
BW	FR	Francistown
BW	GA	Gaborone
BW	JW	Jwaneng
BW	LO	Lobatse
BW	SP	Selibe Phikwe
BW	ST	Sowa Town
BR	AC	Acre
BR	AL	Alagoas
BR	AP	Amapa
BR	AM	Amazonas
BR	BA	Bahia
BR	CE	Ceara
BR	DF	Distrito Federal
BR	ES	Espirito Santo
BR	GO	Goiás
BR	MA	Maranhao
BR	MT	Mato Grosso
BR	MS	Mato Grosso do Sul
BR	MG	Minas Gerais

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
BR	PA	Para
BR	PB	Paraíba
BR	PR	Paraná
BR	PE	Pernambuco
BR	PI	Piauí
BR	RN	Rio Grande do Norte
BR	RS	Rio Grande do Sul
BR	RJ	Rio de Janeiro
BR	RO	Rondonia
BR	RR	Roraima
BR	SC	Santa Catarina
BR	SP	São Paulo
BR	SE	Sergipe
BR	TO	Tocantins
BN	BE	Belait
BN	TE	Temburong
BN	TU	Tutong
BN	BM	Brunei-Muara
BG	01	Blagoevgrad
BG	02	Burgas
BG	08	Dobrich
BG	07	Gabrovo
BG	26	Haskovo
BG	09	Kardjali
BG	10	Kyustendil
BG	11	Lovech
BG	12	Montana
BG	13	Pazardjik
BG	14	Pernik
BG	15	Pleven
BG	16	Plovdiv
BG	17	Razgrad
BG	27	Shumen
BG	19	Silistra
BG	20	Sliven
BG	21	Smolyan
BG	23	Sofia
BG	22	Sofia - town
BG	24	Stara Zagora
BG	25	Targovishte
BG	03	Varna
BG	04	Veliko Tarnovo
BG	05	Vidin
BG	06	Vratza
BG	28	Yambol
BG	18	Ruse
BG	18	Ruse
BG	18	Ruse
BF	BAL	Bale

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
BF	BAM	Bam
BF	BAN	Banwa
BF	BAZ	Bazega
BF	BGR	Bougouriba
BF	BLG	Boulgou
BF	BLK	Boulkiemde
BF	COM	Comoe
BF	GAN	Ganzourgou
BF	GNA	Gnagna
BF	GOU	Gourma
BF	HOU	Houet
BF	IOB	Ioba
BF	KAD	Kadiogo
BF	KEN	Kenedougou
BF	KMD	Komondjari
BF	KOP	Kompienga
BF	KOS	Kossi
BF	KOP	Koulpelogo
BF	KOT	Kouritenga
BF	KOW	Kourweogo
BF	LER	Leraba
BF	LOR	Loroum
BF	MOU	Mouhoun
BF	NAO	Nahouri
BF	NAM	Namentenga
BF	NAY	Nayala
BF	NOU	Noumbiel
BF	OUB	Oubritenga
BF	ODU	Oudalan
BF	PAS	Passore
BF	PON	Poni
BF	SNG	Sanguie
BF	SMT	Sanmatenga
BF	SEN	Seno
BF	SIS	Sissili
BF	SOM	Soum
BF	SOR	Sourou
BF	TAP	Tapoa
BF	TUI	Tuy
BF	YAG	Yagha
BF	YAT	Yatenga
BF	ZIR	Ziro
BF	ZON	Zondoma
BF	ZOU	Zoundweogo
BF	01	Boucle du Mouhoun
BF	02	Cascades
BF	03	Centre
BF	04	Centre-Est
BF	05	Centre-Nord

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
BF	06	Centre-Ouest
BF	07	Centre-Sud
BF	08	Est
BF	09	Hauts-Bassins
BF	10	Nord
BF	11	Plateau-Central
BF	12	Sahel
BF	13	Sud-Ouest
BI	BB	Bubanza
BI	BR	Bururi
BI	CA	Cankuzo
BI	CI	Cibitoke
BI	GI	Gitega
BI	KR	Karuzi
BI	KY	Kayanza
BI	KI	Kirundo
BI	MA	Makamba
BI	MU	Muramvya
BI	MY	Muyinga
BI	MW	Mwaro
BI	NG	Ngozi
BI	RT	Rutana
BI	RY	Ruyigi
BI	BL	Bujumbura Rural
BI	BM	Bujumbura Mairie
BI	RM	Rumonge
KH	1	Banteay Meanchey
KH	3	Kampong Cham
KH	4	Kampong Chhnang
KH	7	Kampot
KH	8	Kandal
KH	9	Kaoh Kong
KH	23	Keb
KH	24	Pailin
KH	12	Phnom Penh
KH	13	Preah Vihear
KH	14	Prey Veng
KH	20	Svay Rieng
KH	10	Krâchéh
KH	11	Môndól Kiri
KH	15	Poŭthŭsăt
KH	16	Rôtânôkiri
KH	17	Siêmréab
KH	18	Preah Sihanouk
KH	19	Stueng Traeng
KH	2	Bătdămbăng
KH	21	Takêv
KH	22	Ôtdâr Méan Choăy
KH	25	Tbong Khmum

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
KH	5	Kâmpóng Spœ
KH	6	Kâmpóng Thum
CM	CE	Centre
CM	LT	Littoral
CM	AD	Adamaoua
CM	EN	Far North
CM	ES	Est
CM	NO	North
CM	NW	North-West
CM	OU	West
CM	SU	Sud
CM	SW	Sud-Ouest
CA	YT	Yukon
CV	BV	Boa Vista
CV	BR	Brava
CV	MA	Maio
CV	MO	Mosteiros
CV	PA	Paul
CV	PN	Porto Novo
CV	PR	Praia
CV	RG	Ribeira Grande
CV	SL	Sal
CV	CA	Santa Catarina
CV	CR	Santa Cruz
CV	SD	Sao Domingos
CV	SF	Sao Filipe
CV	SV	Sao Vicente
CV	TA	Tarrafal
CV	B	Ilhas de Barlavento
CV	CF	Santa Catarina do Fogo
CV	RB	Ribeira Brava
CV	RS	Ribeira Grande de Santiago
CV	S	Ilhas de Sotavento
CV	SM	São Miguel
CV	SO	São Lourenço dos Órgãos
CV	SS	São Salvador do Mundo
CV	TS	Tarrafal de São Nicolau
CF	BB	Bamingui-Bangoran
CF	BGF	Bangui
CF	BK	Basse-Kotto
CF	HM	Haut-Mbomou
CF	HK	Haute-Kotto
CF	LB	Lobaye
CF	MB	Mbomou
CF	NM	Nana-Mambere
CF	UK	Ouaka
CF	AC	Ouham
CF	OP	Ouham-Pende
CF	VK	Vakaga

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
CF	HS	Tö-Sangā / Mbaere-Kadeï
CF	KB	Gribingui
CF	KG	Kémo-Gribingui
CF	MP	Ombella-Mpoko
CF	SE	Sangha
TD	BA	Batha
TD	CB	Chari-Baguirmi
TD	GR	Guera
TD	KA	Kanem
TD	LC	Lac
TD	LC	Logone Occidental
TD	LR	Logone Oriental
TD	MC	Moyen-Chari
TD	OD	Ouaddai
TD	SA	Salamat
TD	TA	Tandjile
TD	BG	Bahr el Ghazal
TD	BO	Bürkū
TD	EE	Inidī ash Sharqī
TD	EO	Inidī al Gharbī
TD	HL	Ḥajjar Lamīs
TD	LO	Lūghūn al Gharbī
TD	MA	Māndūl
TD	ME	Māyū Kibbī ash Sharqī
TD	MO	Māyū Kibbī al Gharbī
TD	ND	Ville de Ndjamena
TD	SI	Silā
TD	TI	Tibesti
TD	WF	Wādī Fīrā'
CL	AI	Aisen del General Carlos Ibanez
CL	AN	Antofagasta
CL	AR	Araucania
CL	AT	Atacama
CL	BI	Bio-Bio
CL	CO	Coquimbo
CL	LI	Libertador General Bernardo O'Hi
CL	LL	Los Lagos
CL	MA	Magallanes y de la Antartica Chi
CL	ML	Maule
CL	RM	Region Metropolitana
CL	TA	Tarapaca
CL	VS	Valparaiso
CL	AP	Arica y Parinacota
CL	LR	Los Ríos
CL	NB	Ñuble
CN	GX	Guangxi

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
CN	GZ	Guizhou
CN	HA	Hainan
CN	HB	Hebei
CN	HL	Heilongjiang
CN	HE	Henan
CN	HK	Hong Kong
CN	HN	Hunan
CN	JX	Jiangxi
CN	JL	Jilin
CN	SH	Shaanxi
CN	SX	Shanxi
CN	AH	Anhui Sheng
CN	BJ	Beijing Shi
CN	CQ	Chongqing Shi
CN	FJ	Fujian Sheng
CN	GD	Guangdong Sheng
CN	GS	Gansu Sheng
CN	HI	Hainan Sheng
CN	JS	Jiangsu Sheng
CN	LN	Liaoning Sheng
CN	MO	Macau SAR (see also separate country code entry under MO)
CN	NM	Nei Mongol Zizhiqu
CN	NX	Ningxia Huizi Zizhiqu
CN	QH	Qinghai Sheng
CN	SC	Sichuan Sheng
CN	SD	Shandong Sheng
CN	SN	Shaanxi Sheng
CN	TJ	Tianjin Shi
CN	TW	Taiwan Sheng (see also separate country code entry under TW)
CN	XJ	Xinjiang Uygur Zizhiqu
CN	XZ	Xizang Zizhiqu
CN	YN	Yunnan Sheng
CN	ZJ	Zhejiang Sheng
CO	AMA	Amazonas
CO	ANT	Antioquia
CO	ARA	Arauca
CO	ATL	Atlantico
CO	BOL	Bolivar
CO	BOY	Boyaca
CO	CAL	Caldas
CO	CAQ	Caqueta
CO	CAS	Casanare
CO	CAU	Cauca
CO	CES	Cesar
CO	CHO	Choco

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
CO	COR	Cordoba
CO	CUN	Cundinamarca
CO	GUA	Guainia
CO	GUV	Guaviare
CO	HUI	Huila
CO	MAG	Magdalena
CO	MET	Meta
CO	NAR	Narino
CO	NSA	Norte de Santander
CO	PUT	Putumayo
CO	QUI	Quindio
CO	RIS	Risaralda
CO	SAP	San Andres y Providencia
CO	SAN	Santander
CO	SUC	Sucre
CO	TOL	Tolima
CO	VAC	Valle del Cauca
CO	VAU	Vaupes
CO	VID	Vichada
CO	DC	Distrito Capital de Bogotá
CO	LAG	La Guajira
KM	A	Anjouan
KM	G	Grande Comore
KM	M	Moheli
CG	11	Bouenza
CG	BZV	Brazzaville
CG	8	Cuvette
CG	15	Cuvette-Ouest
CG	5	Kouilou
CG	2	Lekoumou
CG	7	Likouala
CG	9	Niari
CG	14	Plateaux
CG	12	Pool
CG	13	Sangha
CG	16	Pointe-Noire
CD	BC	Bas-Congo
CD	EQ	Equateur
CD	KE	Kasai-Oriental
CD	KN	Kinshasa
CD	MA	Maniema
CD	NK	Nord-Kivu
CD	SK	Sud-Kivu
CD	BU	Bas-Uélé
CD	HK	Haut-Katanga
CD	HL	Haut-Lomami
CD	HU	Haut-Uélé
CD	IT	Ituri
CD	KC	Kasaï Central

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
CD	KG	Kwango
CD	KL	Kwilu
CD	KS	Kasaï
CD	LO	Lomami
CD	LU	Lualaba
CD	MN	Mai-Ndombe
CD	MO	Mongala
CD	NU	Nord-Ubangi
CD	SA	Sankuru
CD	SU	Sud-Ubangi
CD	TA	Tanganyika
CD	TO	Tshopo
CD	TU	Tshuapa
CR	A	Alajuela
CR	C	Cartago
CR	G	Guanacaste
CR	H	Heredia
CR	L	Limon
CR	P	Puntarenas
CR	SJ	San Jose
CI	AB	Abidjan
CI	YM	Yamoussoukro
CI	BS	Bas-Sassandra
CI	CM	Comoé
CI	DN	Denguélé
CI	GD	Gôh-Djiboua
CI	LC	Lacs
CI	LG	Lagunes
CI	MG	Montagnes
CI	SM	Sassandra-Marahoué
CI	SV	Savanes
CI	VB	Vallée du Bandama
CI	WR	Woroba
CI	ZZ	Zanzan
HR	01	Zagrebačka županija
HR	02	Krapinsko-zagorska županija
HR	03	Sisačko-moslavačka županija
HR	04	Karlovačka županija
HR	05	Varaždinska županija
HR	06	Koprivničko-križevačka županija
HR	07	Bjelovarsko-bilogorska županija
HR	08	Primorsko-goranska županija
HR	09	Ličko-senjska županija

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
HR	10	Virovitičko-podravská županija
HR	11	Požeško-slavonská županija
HR	12	Brodsko-posavská županija
HR	13	Zadarská županija
HR	14	Osječko-baranjská županija
HR	15	Šibensko-kninská županija
HR	16	Vukovarsko-srijemska županija
HR	17	Splitsko-dalmatinská županija
HR	18	Istarská županija
HR	19	Dubrovačko-neretvanská županija
HR	20	Međimurská županija
HR	21	Grad Zagreb
CU	09	Camaguey
CU	08	Ciego de Avila
CU	06	Cienfuegos
CU	12	Granma
CU	14	Guantanamo
CU	11	Holguin
CU	99	Isla de la Juventud
CU	03	La Habana
CU	10	Las Tunas
CU	04	Matanzas
CU	01	Pinar del Rio
CU	07	Sancti Spiritus
CU	13	Santiago de Cuba
CU	05	Villa Clara
CU	15	Artemisa
CU	16	Mayabeque
CY	01	Lefkosia
CY	02	Leymasun
CY	03	Larnaka
CY	04	Mağusa
CY	05	Pafos
CY	06	Keryneia
CZ	10	“Praha, Hlavní město”
CZ	20	Středočeský kraj
CZ	201	Benešov
CZ	202	Beroun
CZ	203	Kladno
CZ	204	Kolín
CZ	205	Kutná Hora
CZ	206	Mělník
CZ	207	Mladá Boleslav
CZ	208	Nymburk
CZ	209	Praha-východ

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
CZ	20A	Praha-západ
CZ	20B	Příbram
CZ	20C	Rakovník
CZ	31	Jihočeský kraj
CZ	311	České Budějovice
CZ	312	Český Krumlov
CZ	313	Jindřichův Hradec
CZ	314	Písek
CZ	315	Prachatice
CZ	316	Strakonice
CZ	317	Tábor
CZ	32	Plzeňský kraj
CZ	321	Domažlice
CZ	322	Klatovy
CZ	323	Plzeň-město
CZ	324	Plzeň-jih
CZ	325	Plzeň-sever
CZ	326	Rokycany
CZ	327	Tachov
CZ	41	Karlovarský kraj
CZ	411	Cheb
CZ	412	Karlovy Vary
CZ	413	Sokolov
CZ	42	Ústecký kraj
CZ	421	Děčín
CZ	422	Chomutov
CZ	423	Litoměřice
CZ	424	Louny
CZ	425	Most
CZ	426	Teplice
CZ	427	Ústí nad Labem
CZ	51	Liberecký kraj
CZ	511	Česká Lípa
CZ	512	Jablonec nad Nisou
CZ	513	Liberec
CZ	514	Semily
CZ	52	Královéhradecký kraj
CZ	521	Hradec Králové
CZ	522	Jičín
CZ	523	Náchod
CZ	524	Rychnov nad Kněžnou
CZ	525	Trutnov
CZ	53	Pardubický kraj
CZ	531	Chrudim
CZ	532	Pardubice
CZ	533	Svitavy
CZ	534	Ústí nad Orlicí
CZ	63	Kraj Vysočina
CZ	631	Havlíčkův Brod

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
CZ	632	Jihlava
CZ	633	Pelhřimov
CZ	634	Třebíč
CZ	635	Žďár nad Sázavou
CZ	64	Jihomoravský kraj
CZ	641	Blansko
CZ	642	Brno-město
CZ	643	Brno-venkov
CZ	644	Břeclav
CZ	645	Hodonín
CZ	646	Vyškov
CZ	647	Znojmo
CZ	71	Olomoucký kraj
CZ	711	Jeseník
CZ	712	Olomouc
CZ	713	Prostějov
CZ	714	Přerov
CZ	715	Šumperk
CZ	72	Zlínský kraj
CZ	721	Kroměříž
CZ	722	Uherské Hradiště
CZ	723	Vsetín
CZ	724	Zlín
CZ	80	Moravskoslezský kraj
CZ	801	Bruntál
CZ	802	Frýdek-Místek
CZ	803	Karviná
CZ	804	Nový Jičín
CZ	805	Opava
CZ	806	Ostrava-město
DK	81	Nordjylland
DK	82	Midtjylland
DK	83	Syddanmark
DK	84	Hovedstaden
DK	85	Sjælland
DJ	DI	Dikhil
DJ	DJ	Djibouti
DJ	OB	Obock
DJ	AR	ʿArtā
DJ	AS	ʿAlī Ṣabīḥ
DJ	TA	Tājūrah
DM	02	Saint Andrew
DM	03	Saint David
DM	04	Saint George
DM	05	Saint John
DM	06	Saint Joseph
DM	07	Saint Luke
DM	08	Saint Mark
DM	09	Saint Patrick

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
DM	10	Saint Paul
DM	11	Saint Peter
DO	02	Azua
DO	03	Baoruco
DO	04	Barahona
DO	05	Dajabon
DO	06	Duarte
DO	07	Elias Pina
DO	09	Espaillat
DO	30	Hato Mayor
DO	10	Independencia
DO	11	La Altagracia
DO	12	La Romana
DO	13	La Vega
DO	14	Maria Trinidad Sanchez
DO	28	Monsenor Nouel
DO	15	Monte Cristi
DO	29	Monte Plata
DO	16	Pedernales
DO	18	Puerto Plata
DO	20	Samana
DO	21	San Cristobal
DO	31	San Jose de Ocoa
DO	22	San Juan
DO	23	San Pedro de Macoris
DO	24	Sanchez Ramirez
DO	25	Santiago
DO	26	Santiago Rodriguez
DO	32	Santo Domingo
DO	27	Valverde
DO	01	Distrito Nacional (Santo Domingo)
DO	08	El Seibo
DO	17	Peravia
DO	19	Hermanas Mirabal
DO	33	Cibao Nordeste
DO	34	Cibao Noroeste
DO	35	Cibao Norte
DO	36	Cibao Sur
DO	37	El Valle
DO	38	Enriquillo
DO	39	Higuamo
DO	40	Ozama
DO	41	Valdesia
DO	42	Yuma
EC	A	Azuay
EC	B	Bolivar
EC	C	Carchi
EC	H	Chimborazo

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
EC	X	Cotopaxi
EC	O	El Oro
EC	E	Esmeraldas
EC	W	Galápagos
EC	G	Guayas
EC	I	Imbabura
EC	L	Loja
EC	R	Los Rios
EC	M	Manabí
EC	S	Morona Santiago
EC	N	Napo
EC	D	Orellana
EC	Y	Pastaza
EC	P	Pichincha
EC	U	Sucumbíos
EC	T	Tungurahua
EC	Z	Zamora Chinchipe
EC	F	Cañar
EC	SD	Santo Domingo de los Tsáchilas
EC	SE	Santa Elena
EG	DK	Ad Daqahliyah
EG	BA	Al Bahr al Ahmar
EG	BH	Al Buhayrah
EG	FYM	Al Fayyum
EG	GH	Al Gharbiyah
EG	ALX	Al Iskandariyah
EG	IS	Al Isma'iliyah
EG	GZ	Al Jizah
EG	MNF	Al Minufiyah
EG	MN	Al Minya
EG	C	Al Qahirah
EG	KB	Al Qalyubiyah
EG	WAD	Al Wadi al Jadid
EG	SUZ	As Suways
EG	SHR	Ash Sharqiyah
EG	ASN	Aswan
EG	AST	Asyut
EG	BNS	Bani Suwayf
EG	DT	Dumyat
EG	JS	Janub Sina'
EG	KFS	Kafr ash Shaykh
EG	MT	Matruh
EG	KN	Qina
EG	SIN	Shamal Sina'
EG	SHG	Suhaj
EG	LX	Al Uqsur
EG	PTS	Būr Sa'īd
SV	AH	Ahuachapan

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SV	CA	Cabanas
SV	CH	Chalatenango
SV	CU	Cuscatlan
SV	LI	La Libertad
SV	PA	La Paz
SV	UN	La Union
SV	MO	Morazan
SV	SM	San Miguel
SV	SS	San Salvador
SV	SV	San Vicente
SV	SA	Santa Ana
SV	SO	Sonsonate
SV	US	Usulután
GQ	AN	Provincia Annobon
GQ	BN	Provincia Bioko Norte
GQ	BS	Provincia Bioko Sur
GQ	CS	Provincia Centro Sur
GQ	KN	Provincia Kie-Ntem
GQ	LI	Provincia Litoral
GQ	WN	Provincia Wele-Nzas
GQ	C	Région Continentale
GQ	DJ	Djibloho
GQ	I	Región Insular
ER	MA	Central (Maekel)
ER	SK	Northern Red Sea (Semien-Keih-Bahri)
ER	DK	Southern Red Sea (Debub-Keih-Bahri)
ER	AN	'Anseba
ER	DU	Debub
ER	GB	Qāsh-Barkah
EE	130	Alutaguse
EE	141	Anija
EE	142	Antsla
EE	171	Elva
EE	184	Haapsalu
EE	191	Haljala
EE	198	Harku
EE	205	Hiiumaa
EE	214	Häädemeeste
EE	245	Jõelähtme
EE	247	Jõgeva
EE	251	Jõhvi
EE	255	Järva
EE	272	Kadrina
EE	283	Kambja
EE	284	Kanepi
EE	291	Kastre
EE	293	Kehtna

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
EE	296	Keila
EE	303	Kihnu
EE	305	Kiili
EE	317	Kohila
EE	321	Kohtla-Järve
EE	338	Kose
EE	353	Kuusalu
EE	37	Harjumaa
EE	39	Hiiumaa
EE	424	Loksa
EE	430	Lääneranna
EE	431	Lääne-Harju
EE	432	Luunja
EE	441	Lääne-Nigula
EE	442	Lüganuse
EE	446	Maardu
EE	45	Ida-Virumaa
EE	478	Muhu
EE	480	Mulgi
EE	486	Mustvee
EE	50	Jõgevamaa
EE	503	Märjamaa
EE	511	Narva
EE	514	Narva-Jõesuu
EE	52	Järvamaa
EE	528	Nõo
EE	557	Otepää
EE	56	Läänemaa
EE	567	Paide
EE	586	Peipsiääre
EE	60	Lääne-Virumaa
EE	615	Põhja-Sakala
EE	618	Põltsamaa
EE	622	Põlva
EE	624	Pärnu
EE	638	Põhja-Pärnumaa
EE	64	Põlvamaa
EE	651	Raasiku
EE	653	Rae
EE	661	Rakvere
EE	663	Rakvere
EE	668	Rapla
EE	68	Pärnumaa
EE	689	Ruhnu
EE	698	Rõuge
EE	708	Räpina
EE	71	Raplamaa
EE	712	Saarde
EE	714	Saaremaa

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
EE	719	Saku
EE	726	Saue
EE	732	Setomaa
EE	735	Sillamäe
EE	74	Saaremaa
EE	784	Tallinn
EE	79	Tartumaa
EE	792	Tapa
EE	793	Tartu
EE	796	Tartu
EE	803	Toila
EE	809	Tori
EE	81	Valgamaa
EE	824	Tõrva
EE	834	Türi
EE	84	Viljandimaa
EE	855	Valga
EE	87	Võrumaa
EE	890	Viimsi
EE	897	Viljandi
EE	899	Viljandi
EE	901	Vinni
EE	903	Viru-Nigula
EE	907	Vormsi
EE	917	Võru
EE	919	Võru
EE	928	Väike-Maarja
ET	AA	Addis Ababa
ET	AF	Afar
ET	DD	Dire Dawa
ET	OR	Oromia
ET	SO	Somali
ET	SN	Southern Nations - Nationalities and Peoples Region
ET	TI	Tigray
ET	AM	Amara
ET	BE	Bīnshangul Gumuz
ET	GA	Gambela Peoples
ET	HA	Hārerī Hizb
FJ	C	Central Division
FJ	E	Eastern Division
FJ	N	Northern Division
FJ	R	Rotuma
FJ	W	Western Division
FJ	01	Ba
FJ	02	Bua
FJ	03	Cakaudrove
FJ	04	Kadavu
FJ	05	Lau

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
FJ	06	Lomaiviti
FJ	07	Macuata
FJ	08	Nadroga and Navosa
FJ	09	Naitasiri
FJ	10	Namosi
FJ	11	Ra
FJ	12	Rewa
FJ	13	Serua
FJ	14	Tailevu
FI	01	Landskapet Åland
FI	02	Södra Karelen
FI	03	Södra Österbotten
FI	04	Södra Savolax
FI	05	Kajanaland
FI	06	Kanta-Häme
FI	07	Mellersta Österbotten
FI	08	Mellersta Finland
FI	09	Kymmenedalen
FI	10	Lappland
FI	11	Pirkanmaa
FI	12	Pohjanmaa
FI	13	Pohjois-Karjala
FI	14	Pohjois-Pohjanmaa
FI	15	Pohjois-Savo
FI	16	Päijät-Häme
FI	17	Satakunta
FI	18	Uusimaa
FI	19	Varsinais-Suomi
FR	20R	Corse
FR	01	Ain
FR	02	Aisne
FR	03	Allier
FR	04	Alpes-de-Haute-Provence
FR	05	Hautes-Alpes
FR	06	Alpes-Maritimes
FR	07	Ardèche
FR	08	Ardenne
FR	09	Ariège
FR	10	Aube
FR	11	Aude
FR	12	Aveyron
FR	13	Bouches-du-Rhône
FR	14	Calvados
FR	15	Cantal
FR	16	Charente
FR	17	Charente-Maritime
FR	18	Cher
FR	19	Corrèze
FR	21	Côte-d'Or

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
FR	22	Côtes-d’Armor
FR	23	Creuse
FR	24	Dordogne
FR	25	Doubs
FR	26	Drôme
FR	27	Eure
FR	28	Eure-et-Loir
FR	29	Finistère
FR	2A	Corse-du-Sud
FR	2B	Haute-Corse
FR	30	Gard
FR	31	Haute-Garonne
FR	32	Gers
FR	33	Gironde
FR	34	Hérault
FR	35	Ille-et-Vilaine
FR	36	Indre
FR	37	Indre-et-Loire
FR	38	Isère
FR	39	Jura
FR	40	Landes
FR	41	Loir-et-Cher
FR	42	Loire
FR	43	Haute-Loire
FR	44	Loire-Atlantique
FR	45	Loiret
FR	46	Lot
FR	47	Lot-et-Garonne
FR	48	Lozère
FR	49	Maine-et-Loire
FR	50	Manche
FR	51	Marne
FR	52	Haute-Marne
FR	53	Mayenne
FR	54	Meurthe-et-Moselle
FR	55	Meuse
FR	56	Morbihan
FR	57	Moselle
FR	58	Nièvre
FR	59	Nord
FR	60	Oise
FR	61	Orne
FR	62	Pas-de-Calais
FR	63	Puy-de-Dôme
FR	64	Pyrénées-Atlantiques
FR	65	Hautes-Pyrénées
FR	66	Pyrénées-Orientales
FR	67	Bas-Rhin
FR	68	Haut-Rhin

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
FR	69	Rhône
FR	70	Haute-Saône
FR	71	Saône-et-Loire
FR	72	Sarthe
FR	73	Savoie
FR	74	Haute-Savoie
FR	75	Paris
FR	76	Seine-Maritime
FR	77	Seine-et-Marne
FR	78	Yvelines
FR	79	Deux-Sèvres
FR	80	Somme
FR	81	Tarn
FR	82	Tarn-et-Garonne
FR	83	Var
FR	84	Vaucluse
FR	85	Vendée
FR	86	Vienne
FR	87	Haute-Vienne
FR	88	Vosges
FR	89	Yonne
FR	90	Territoire de Belfort
FR	91	Essonne
FR	92	Hauts-de-Seine
FR	93	Seine-Saint-Denis
FR	94	Val-de-Marne
FR	95	Val-d'Oise
FR	971	Guadeloupe
FR	972	Martinique
FR	973	Guyane (française)
FR	974	La Réunion
FR	976	Mayotte
FR	ARA	Auvergne-Rhône-Alpes
FR	BFC	Bourgogne-Franche-Comté
FR	BL	Saint-Barthélemy (see also separate country code entry under BL)
FR	BRE	Bretagne
FR	CP	Clipperton
FR	CVL	Centre-Val de Loire
FR	GES	Grand-Est
FR	GF	Guyane (française) (see also separate country code entry under GF)
FR	GP	Guadeloupe (see also separate country code entry under GP)
FR	HDF	Hauts-de-France
FR	IDF	Île-de-France

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
FR	MF	Saint-Martin (see also separate country code entry under MF)
FR	MQ	Martinique (see also separate country code entry under MQ)
FR	NAQ	Nouvelle-Aquitaine
FR	NC	Nouvelle-Calédonie (see also separate country code entry under NC)
FR	NOR	Normandie
FR	OCC	Occitanie
FR	PAC	Provence-Alpes-Côte-d’Azur
FR	PDL	Pays-de-la-Loire
FR	PF	Polynésie française (see also separate country code entry under PF)
FR	PM	Saint-Pierre-et-Miquelon (see also separate country code entry under PM)
FR	RE	La Réunion (see also separate country code entry under RE)
FR	TF	Terres australes françaises (see also separate country code entry under TF)
FR	WF	Wallis-et-Futuna (see also separate country code entry under WF)
FR	YT	Mayotte (see also separate country code entry under YT)
GA	1	Estuaire
GA	2	Haut-Ogooue
GA	3	Moyen-Ogooue
GA	4	Ngounie
GA	5	Nyanga
GA	6	Ogooue-Ivindo
GA	7	Ogooue-Lolo
GA	8	Ogooue-Maritime
GA	9	Woleu-Ntem
GM	B	Banjul
GM	M	Central River
GM	L	Lower River
GM	N	North Bank
GM	U	Upper River
GM	W	Western
GE	AB	Abkhazia
GE	AJ	Ajaria
GE	GU	Guria

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
GE	IM	Imereti
GE	KA	Kakheti
GE	KK	Kvemo Kartli
GE	MM	Mtskheta-Mtianeti
GE	RL	Racha Lechkhumi and Kvemo Svanet
GE	SZ	Samegrelo-Zemo Svaneti
GE	SJ	Samtskhe-Javakheti
GE	SK	Shida Kartli
GE	TB	Tbilisi
DE	BY	Bayern
DE	BE	Berlin
DE	BB	Brandenburg
DE	HB	Bremen
DE	HH	Hamburg
DE	HE	Hessen
DE	MV	Mecklenburg-Vorpommern
DE	NI	Niedersachsen
DE	NW	Nordrhein-Westfalen
DE	RP	Rheinland-Pfalz
DE	RP	Rhineland-Palatinate
DE	SL	Saarland
DE	SN	Sachsen
DE	ST	Sachsen-Anhalt
DE	SH	Schleswig-Holstein
DE	TH	Thüringen
DE	BW	Baden-Württemberg
GH	UE	Upper East Region
GH	UW	Upper West Region
GH	AA	Greater Accra
GH	AF	Ahafo
GH	AH	Ashanti
GH	BE	Bono East
GH	BO	Bono
GH	CP	Central
GH	EP	Eastern
GH	NE	North East
GH	NP	Northern
GH	OT	Oti
GH	SV	Savannah
GH	TV	Volta
GH	WN	Western North
GH	WP	Western
GR	69	Άγιον Όρος
GR	A	Anatolikí Makedonía kai Thráki
GR	B	Kentrikí Makedonía
GR	C	Dytikí Makedonía
GR	D	Ípeiros

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
GR	E	Thessalía
GR	F	Ionía Nísia
GR	G	Dytikí Elláda
GR	H	Stereá Elláda
GR	I	Attikí
GR	J	Pelopónnisos
GR	K	Vóreio Aigaío
GR	L	Nótio Aigaío
GR	M	Kríti
GL	AV	Avannaata Kommunia
GL	KU	Kommune Kujalleq
GL	QE	Qeqqata Kommunia
GL	QT	Kommune Qeqertalik
GL	SM	Kommuneqarfik Sermer-sooq
GD	01	Saint Andrew
GD	02	Saint David
GD	03	Saint George
GD	04	Saint John
GD	05	Saint Mark
GD	06	Saint Patrick
GD	10	Southern Grenadine Islands
GT	AV	Alta Verapaz
GT	BV	Baja Verapaz
GT	CM	Chimaltenango
GT	CQ	Chiquimula
GT	PE	El Peten
GT	PR	El Progreso
GT	QC	El Quiche
GT	ES	Escuintla
GT	GU	Guatemala
GT	HU	Huehuetenango
GT	IZ	Izabal
GT	JA	Jalapa
GT	JU	Jutiapa
GT	QZ	Quetzaltenango
GT	RE	Retalhuleu
GT	SA	Sacatepequez
GT	SM	San Marcos
GT	SR	Santa Rosa
GT	SO	Solola
GT	SU	Suchitepequez
GT	TO	Totonicapan
GT	ZA	Zacapa
GN	BE	Beyla
GN	BF	Boffa
GN	B	Boke
GN	C	Conakry
GN	CO	Coyah

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
GN	DB	Dabola
GN	DL	Dalaba
GN	DI	Dinguiraye
GN	DU	Dubreka
GN	F	Faranah
GN	FO	Forecariah
GN	FR	Fria
GN	GA	Gaoual
GN	K	Kankan
GN	KE	Kerouane
GN	D	Kindia
GN	KS	Kissidougou
GN	KB	Koubia
GN	KN	Koundara
GN	KO	Kouroussa
GN	L	Labe
GN	LE	Lelouma
GN	LO	Lola
GN	MC	Macenta
GN	ML	Mali
GN	M	Mamou
GN	MD	Mandiana
GN	N	Nzerekore
GN	PI	Pita
GN	SI	Siguiri
GN	TE	Telimele
GN	TO	Tougue
GN	YO	Yomou
GN	GU	Guékédou
GW	BS	Bissau Region
GW	BL	Bolama Region
GW	CA	Cacheu Region
GW	GA	Gabu Region
GW	OI	Oio Region
GW	QU	Quinara Region
GW	TO	Tombali Region
GW	BA	Bafatá
GW	BM	Biombo
GW	L	Leste
GW	N	Norte
GW	S	Sul
GY	BA	Barima-Waini
GY	CU	Cuyuni-Mazaruni
GY	DE	Demerara-Mahaica
GY	EB	East Berbice-Corentyne
GY	ES	Essequibo Islands-West Demerara
GY	MA	Mahaica-Berbice
GY	PM	Pomeroon-Supenaam

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
GY	PT	Potaro-Siparuni
GY	UD	Upper Demerara-Berbice
GY	UT	Upper Takutu-Upper Essequibo
HT	AR	Artibonite
HT	CE	Centre
HT	GA	Grand'Anse
HT	ND	Nord
HT	NE	Nord-Est
HT	NO	Nord-Ouest
HT	OU	Ouest
HT	SD	Sud
HT	SE	Sud-Est
HT	NI	Nippes
HN	AT	Atlantida
HN	CH	Choluteca
HN	CL	Colon
HN	CM	Comayagua
HN	CP	Copan
HN	CR	Cortes
HN	EP	El Paraiso
HN	FM	Francisco Morazan
HN	GD	Gracias a Dios
HN	IN	Intibuca
HN	IB	Islas de la Bahia (Bay Islands)
HN	LP	La Paz
HN	LE	Lempira
HN	OC	Ocotepeque
HN	OL	Olancho
HN	SB	Santa Barbara
HN	VA	Valle
HN	YO	Yoro
HU	BK	Bacs-Kiskun
HU	BA	Baranya
HU	BE	Bekes
HU	BC	Bekescsaba
HU	BZ	Borsod-Abauj-Zemplen
HU	BU	Budapest
HU	CS	Csongrad
HU	DE	Debrecen
HU	DU	Dunaujvaros
HU	EG	Eger
HU	FE	Fejer
HU	GY	Gyor
HU	GS	Gyor-Moson-Sopron
HU	HB	Hajdu-Bihar
HU	HE	Heves
HU	HV	Hodmezovasarhely

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
HU	JN	Jasz-Nagykun-Szolnok
HU	KV	Kaposvar
HU	KE	Kecskemet
HU	KE	Komarom-Esztergom
HU	MI	Miskolc
HU	NK	Nagykanizsa
HU	NO	Nograd
HU	NY	Nyiregyhaza
HU	PE	Pecs
HU	PE	Pest
HU	SO	Somogy
HU	SN	Sopron
HU	SZ	Szabolcs-Szatmar-Bereg
HU	SZ	Szeged
HU	SF	Szekesfehervar
HU	SK	Szolnok
HU	SH	Szombathely
HU	TB	Tatabanya
HU	TO	Tolna
HU	VA	Vas
HU	VE	Veszprem
HU	ZA	Zala
HU	ZE	Zalaegerszeg
HU	ER	Érd
HU	ST	Salgótarján
IS	7	Austurland
IS	3	Vesturland
IS	1	Höfuðborgarsvæði
IS	2	Suðurnes
IS	4	Vestfirðir
IS	5	Norðurland vestra
IS	6	Norðurland eystra
IS	8	Suðurland
IS	AKH	Akrahreppur
IS	AKN	Akraneskaupstaður
IS	AKU	Akureyrarbær
IS	ARN	Árneshreppur
IS	ASA	Ásahreppur
IS	BFJ	Borgarfjarðarhreppur
IS	BLA	Bláskógabyggð
IS	BLO	Blönduósbær
IS	BOG	Borgarbyggð
IS	BOL	Bolungarvíkurkaupstaður
IS	DAB	Dalabyggð
IS	DAV	Dalvíkurbyggð
IS	DJU	Djúpavogshreppur
IS	EOM	Eyja- og Miklaholtshreppur
IS	EYF	Eyjafjarðarsveit
IS	FJD	Fjarðabyggð

continues on next page

Table 13 - continued from previous page

Country	ISO 3166-2 Code	State Name
IS	FJL	Fjallabyggð
IS	FLA	Flóahreppur
IS	FLD	Fljótsdalshérað
IS	FLR	Fljótsdalshreppur
IS	GAR	Garðabær
IS	GOG	Grímsnes- og Grafningshreppur
IS	GRN	Grindavíkurbær
IS	GRU	Grundarfjarðarbær
IS	GRY	Grýtubakkahreppur
IS	HAF	Hafnarfjarðarkaupstaður
IS	HEL	Helgafellssveit
IS	HRG	Hörgársveit
IS	HRU	Hrunamannahreppur
IS	HUT	Húnavatnshreppur
IS	HUV	Húnaþing vestra
IS	HVA	Hvalfjarðarsveit
IS	HVE	Hveragerðisbær
IS	ISA	Ísafjarðarbær
IS	KAL	Kaldrananeshreppur
IS	KJO	Kjósarhreppur
IS	KOP	Kópavogsbær
IS	LAN	Langesbyggð
IS	MOS	Mosfellsbær
IS	MYR	Mýrdalshreppur
IS	NOR	Norðurþing
IS	RGE	Rangárþing eystra
IS	RGY	Rangárþing ytra
IS	RHH	Reykholahreppur
IS	RKN	Reykjanesbær
IS	RKV	Reykjavíkurborg
IS	SBH	Svalbarðshreppur
IS	SBT	Svalbarðsstrandarhreppur
IS	SDN	Suðurnesjabær
IS	SDV	Súðavíkurhreppur
IS	SEL	Seltjarnarnesbær
IS	SEY	Seyðisfjarðarkaupstaður
IS	SFA	Sveitarfélagið Árborg
IS	SHF	Sveitarfélagið Hornafjörður
IS	SKF	Skaftárhreppur
IS	SKG	Skagabyggð
IS	SKO	Skorradalshreppur
IS	SKU	Skútustaðahreppur
IS	SNF	Snæfellsbær
IS	SOG	Skeiða- og Gnúpverjahreppur
IS	SOL	Sveitarfélagið Ölfus
IS	SSF	Sveitarfélagið Skagafjörður
IS	SSS	Sveitarfélagið Skagaströnd

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
IS	STR	Strandabyggð
IS	STY	Stykkishólmsbær
IS	SVG	Sveitarfélagið Vogar
IS	TAL	Tálknafjarðarhreppur
IS	THG	Þingeyjarsveit
IS	TJO	Tjörneshreppur
IS	VEM	Vestmannaeyjabær
IS	VER	Vesturbyggð
IS	VOP	Vopnafjarðarhreppur
IN	BR	Berat
IN	DL	Delvine
IN	GJ	Gjirokaster
IN	KA	Kavaje
IN	SK	Skrapar
IN	TR	Tirane
IN	AN	Andaman and Nicobar Islands
IN	AP	Andhra Pradesh
IN	AR	Arunāchal Pradesh
IN	AS	Assam
IN	CH	Chandīgarh
IN	CT	Chhattīsgarh
IN	DH	Dādra and Nagar Haveli and Damān and Diu
IN	GA	Goa
IN	HP	Himāchal Pradesh
IN	HR	Haryāna
IN	JH	Jhārkhand
IN	JK	Jammu and Kashmīr
IN	KL	Kerala
IN	LA	Ladākh
IN	LD	Lakshadweep
IN	MH	Mahārāshtra
IN	ML	Meghālaya
IN	MN	Manipur
IN	MP	Madhya Pradesh
IN	MZ	Mizoram
IN	NL	Nāgāland
IN	OR	Odisha
IN	PB	Punjab
IN	PY	Puducherry
IN	RJ	Rājasthān
IN	TG	Telangāna
IN	TN	Tamil Nādu
IN	UP	Uttar Pradesh
IN	UT	Uttarākhand
IN	WB	West Bengal
ID	AC	Aceh
ID	BA	Bali

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
ID	BT	Banten
ID	BE	Bengkulu
ID	GO	Gorontalo
ID	JK	Jakarta Raya
ID	JA	Jambi
ID	JB	Jawa Barat
ID	JT	Jawa Tengah
ID	JI	Jawa Timur
ID	KB	Kalimantan Barat
ID	KS	Kalimantan Selatan
ID	KT	Kalimantan Tengah
ID	KI	Kalimantan Timur
ID	BB	Kepulauan Bangka Belitung
ID	LA	Lampung
ID	ML	Maluku
ID	MU	Maluku Utara
ID	NB	Nusa Tenggara Barat
ID	NT	Nusa Tenggara Timur
ID	PP	Papua
ID	RI	Riau
ID	SN	Sulawesi Selatan
ID	ST	Sulawesi Tengah
ID	SG	Sulawesi Tenggara
ID	SA	Sulawesi Utara
ID	SB	Sumatera Barat
ID	SS	Sumatera Selatan
ID	SU	Sumatera Utara
ID	YO	Yogyakarta
ID	JW	Jawa
ID	KA	Kalimantan
ID	KR	Kepulauan Riau
ID	KU	Kalimantan Utara
ID	NU	Nusa Tenggara
ID	PB	Papua Barat
ID	SL	Sulawesi
ID	SM	Sumatera
ID	SR	Sulawesi Barat
IR	24	Ardabil
IR	18	Bushehr
IR	10	Esfahan
IR	07	Fars
IR	01	Gilan
IR	27	Golestan
IR	13	Hamadan
IR	22	Hormozgan
IR	16	Ilam
IR	08	Kerman
IR	05	Kermanshah
IR	06	Khuzestan

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
IR	15	Lorestan
IR	00	Markazi
IR	02	Mazandaran
IR	26	Qazvin
IR	25	Qom
IR	20	Semnan
IR	23	Tehran
IR	21	Yazd
IR	19	Zanjan
IR	03	Āzārbāyjān-e Shārqī
IR	04	Āzārbāyjān-e Ghārbī
IR	09	Khorāsān-e Razavī
IR	11	Sīstān va Balūchestān
IR	12	Kordestān
IR	14	Chahār Mahāl va Bakhtiārī
IR	17	Kohgilūyeh va Bowyer Aḥ-mad
IR	28	Khorāsān-e Shomālī
IR	29	Khorāsān-e Jonūbī
IR	30	Alborz
IQ	AN	Al Anbar
IQ	BA	Al Basrah
IQ	MU	Al Muthanna
IQ	QA	Al Qadisyah
IQ	NA	An Najaf
IQ	AR	Arbil
IQ	SU	As Sulaymaniyah
IQ	BB	Babil
IQ	BG	Baghdad
IQ	DA	Dahuk
IQ	DQ	Dhi Qar
IQ	DI	Diyala
IQ	MA	Maysan
IQ	NI	Ninawa
IQ	SD	Salah ad Din
IQ	WA	Wasit
IQ	KA	Karbalā'
IQ	KI	Kirkūk
IE	CW	Carlow
IE	CN	Cavan
IE	CE	Clare
IE	CO	Cork
IE	DL	Donegal
IE	D	Dublin
IE	G	Galway
IE	KE	Kerry
IE	KE	Kildare
IE	KK	Kilkenny
IE	LS	Laois

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
IE	LM	Leitrim
IE	LK	Limerick
IE	LD	Longford
IE	LH	Louth
IE	MO	Mayo
IE	MH	Meath
IE	MO	Monaghan
IE	OY	Offaly
IE	RN	Roscommon
IE	SO	Sligo
IE	TA	Tipperary
IE	WD	Waterford
IE	WH	Westmeath
IE	WX	Wexford
IE	WW	Wicklow
IE	C	Connaught
IE	L	Leinster
IE	M	Munster
IE	U	Ulster
IL	HA	Haifa
IL	JM	Jehuda Mountains
IL	TA	Tel Aviv (Gosh Dan)
IL	D	HaDarom
IL	M	HaMerkaz
IL	Z	HaTsafon
IT	AG	Agrigento
IT	AL	Alessandria
IT	AN	Ancona
IT	AR	Arezzo
IT	AP	Ascoli Piceno
IT	AT	Asti
IT	AV	Avellino
IT	BA	Bari
IT	BL	Belluno
IT	BN	Benevento
IT	BG	Bergamo
IT	BI	Biella
IT	BO	Bologna
IT	BZ	Bolzano
IT	BS	Brescia
IT	BR	Brindisi
IT	CA	Cagliari
IT	CL	Caltanissetta
IT	CB	Campobasso
IT	CE	Caserta
IT	CT	Catania
IT	CZ	Catanzaro
IT	CH	Chieti
IT	CO	Como

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
IT	CS	Cosenza
IT	CR	Cremona
IT	KR	Crotone
IT	CN	Cuneo
IT	EN	Enna
IT	FE	Ferrara
IT	FI	Firenze
IT	FG	Foggia
IT	FC	Forli-Cesena
IT	FR	Frosinone
IT	GE	Genova
IT	GO	Gorizia
IT	GR	Grosseto
IT	IM	Imperia
IT	IS	Isernia
IT	AQ	L'Aquila
IT	SP	La Spezia
IT	LT	Latina
IT	LE	Lecce
IT	LC	Lecco
IT	LI	Livorno
IT	LO	Lodi
IT	LU	Lucca
IT	MC	Macerata
IT	MN	Mantova
IT	MS	Massa-Carrara
IT	MT	Matera
IT	ME	Messina
IT	MI	Milano
IT	MO	Modena
IT	NA	Napoli
IT	NO	Novara
IT	NU	Nuoro
IT	OR	Oristano
IT	PD	Padova
IT	PA	Palermo
IT	PR	Parma
IT	PV	Pavia
IT	PG	Perugia
IT	PU	Pesaro e Urbino
IT	PE	Pescara
IT	PC	Piacenza
IT	PI	Pisa
IT	PT	Pistoia
IT	PN	Pordenone
IT	PZ	Potenza
IT	PO	Prato
IT	RG	Ragusa
IT	RA	Ravenna

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
IT	RC	Reggio Calabria
IT	RE	Reggio Emilia
IT	RI	Rieti
IT	RN	Rimini
IT	RM	Roma
IT	RO	Rovigo
IT	SA	Salerno
IT	SS	Sassari
IT	SV	Savona
IT	SI	Siena
IT	SR	Siracusa
IT	SO	Sondrio
IT	TA	Taranto
IT	TE	Teramo
IT	TR	Terni
IT	TO	Torino
IT	TP	Trapani
IT	TN	Trento
IT	TV	Treviso
IT	TS	Trieste
IT	UD	Udine
IT	VA	Varese
IT	VE	Venezia
IT	VB	Verbano-Cusio-Ossola
IT	VC	Vercelli
IT	VR	Verona
IT	VV	Vibo Valentia
IT	VI	Vicenza
IT	VT	Viterbo
IT	21	Piemonte
IT	23	Valle d'Aosta
IT	25	Lombardia
IT	32	Trentino-Südtirol
IT	34	Veneto
IT	36	Friuli Venezia Giulia
IT	42	Liguria
IT	45	Emilia-Romagna
IT	52	Toscana
IT	55	Umbria
IT	57	Marche
IT	62	Lazio
IT	65	Abruzzo
IT	67	Molise
IT	72	Campania
IT	75	Puglia
IT	77	Basilicata
IT	78	Calabria
IT	82	Sicilia
IT	88	Sardegna

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
IT	BT	Barletta-Andria-Trani
IT	FM	Fermo
IT	MB	Monza e Brianza
IT	SU	Sud Sardegna
JM	01	Kingston
JM	02	Saint Andrew
JM	03	Saint Thomas
JM	04	Portland
JM	05	Saint Mary
JM	06	Saint Ann
JM	07	Trelawny
JM	08	Saint James
JM	09	Hanover
JM	10	Westmoreland
JM	11	Saint Elizabeth
JM	12	Manchester
JM	13	Clarendon
JM	14	Saint Catherine
JP	23	Aichi
JP	05	Akita
JP	02	Aomori
JP	12	Chiba
JP	38	Ehime
JP	18	Fukui
JP	40	Fukuoka
JP	07	Fukushima
JP	21	Gifu
JP	34	Hiroshima
JP	01	Hokkaido
JP	28	Hyogo
JP	08	Ibaraki
JP	17	Ishikawa
JP	03	Iwate
JP	37	Kagawa
JP	46	Kagoshima
JP	14	Kanagawa
JP	39	Kochi
JP	43	Kumamoto
JP	26	Kyoto
JP	24	Mie
JP	04	Miyagi
JP	45	Miyazaki
JP	20	Nagano
JP	42	Nagasaki
JP	29	Nara
JP	15	Niigata
JP	44	Oita
JP	33	Okayama
JP	47	Okinawa

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
JP	27	Osaka
JP	41	Saga
JP	11	Saitama
JP	25	Shiga
JP	32	Shimane
JP	22	Shizuoka
JP	09	Tochigi
JP	36	Tokushima
JP	13	Tokyo
JP	31	Tottori
JP	16	Toyama
JP	30	Wakayama
JP	06	Yamagata
JP	35	Yamaguchi
JP	19	Yamanashi
JP	10	Gunma
JO	AM	'Amman
JO	AJ	Ajlun
JO	KA	Al Karak
JO	MA	Al Maфраq
JO	AT	At Tafilah
JO	AZ	Az Zarqa'
JO	IR	Irbid
JO	JA	Jarash
JO	MA	Ma'an
JO	MD	Madaba
JO	AQ	Al 'Aqabah
JO	BA	Al Balqā'
JO	MN	Ma'an
KZ	ALA	Almaty
KZ	AKM	Aqmola oblysy
KZ	AKT	Aqtöbe oblysy
KZ	ALM	Almaty oblysy
KZ	AST	Nur-Sultan
KZ	ATY	Atyrauskaya oblast'
KZ	KAR	Qaraghandy oblysy
KZ	KUS	Qostanay oblysy
KZ	KZY	Qyzylorda oblysy
KZ	MAN	Mangystauskaya oblast'
KZ	PAV	Pavlodarskaya oblast'
KZ	SEV	Soltüstik Qazaqstan oblysy
KZ	SHY	Šimkent
KZ	VOS	Vostočno-Kazakhstanskaya oblast'
KZ	YUZ	Türkistan oblysy
KZ	ZAP	Zapadno-Kazakhstanskaya oblast'
KZ	ZHA	Zhambylskaya oblast'
KE	01	Baringo

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
KE	02	Bomet
KE	03	Bungoma
KE	04	Busia
KE	05	Elgeyo/Marakwet
KE	06	Embu
KE	07	Garissa
KE	08	Homa Bay
KE	09	Isiolo
KE	10	Kajiado
KE	11	Kakamega
KE	12	Kericho
KE	13	Kiambu
KE	14	Kilifi
KE	15	Kirinyaga
KE	16	Kisii
KE	17	Kisumu
KE	18	Kitui
KE	19	Kwale
KE	20	Laikipia
KE	21	Lamu
KE	22	Machakos
KE	23	Makueni
KE	24	Mandera
KE	25	Marsabit
KE	26	Meru
KE	27	Migori
KE	28	Mombasa
KE	29	Murang'a
KE	30	Nairobi City
KE	31	Nakuru
KE	32	Nandi
KE	33	Narok
KE	34	Nyamira
KE	35	Nyandarua
KE	36	Nyeri
KE	37	Samburu
KE	38	Siaya
KE	39	Taita/Taveta
KE	40	Tana River
KE	41	Tharaka-Nithi
KE	42	Trans Nzoia
KE	43	Turkana
KE	44	Uasin Gishu
KE	45	Vihiga
KE	46	Wajir
KE	47	West Pokot
KI	G	Gilbert Islands
KI	L	Line Islands
KI	P	Phoenix Islands

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
KP	04	Chagang-do
KP	09	Hamgyong-bukto
KP	08	Hamgyong-namdo
KP	06	Hwanghae-bukto
KP	05	Hwanghae-namdo
KP	07	Kangwon-do
KP	03	P'yongan-bukto
KP	02	P'yongan-namdo
KP	01	Phyeongyang
KP	10	Ryangkangto
KP	13	Rasŏn
KP	14	Namp'o
KR	11	Seoul-teukbyeolsi
KR	26	Busan-gwangyeoksi
KR	27	Daegu-gwangyeoksi
KR	28	Incheon-gwangyeoksi
KR	29	Gwangju-gwangyeoksi
KR	30	Daejeon-gwangyeoksi
KR	31	Ulsan-gwangyeoksi
KR	41	Gyeonggi-do
KR	42	Gangwon-do
KR	43	Chungcheongbuk-do
KR	44	Chungcheongnam-do
KR	45	Jeollabuk-do
KR	46	Jeollanam-do
KR	47	Gyeongsangbuk-do
KR	48	Gyeongsangnam-do
KR	49	Jeju-teukbyeoljachido
KR	50	Sejong
KW	AH	Al Ahmadi
KW	FA	Al Farwaniyah
KW	HA	Hawalli
KW	JA	Al Jahra'
KW	KU	Al 'Aşimah
KW	MU	Mubarak al Kabir
KG	B	Batken
KG	GB	Bishkek
KG	C	Chu
KG	J	Jalal-Abad
KG	N	Naryn
KG	O	Osh
KG	T	Talas
KG	Y	Ysyk-Kol
KG	GO	Osh Shaary
LA	AT	Attapu
LA	BK	Bokeo
LA	BL	Bolikhamxai
LA	CH	Champasak
LA	HO	Houaphan

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
LA	KH	Khammouan
LA	LM	Louang Namtha
LA	LP	Louangphabang
LA	OU	Oudomxai
LA	PH	Phongsali
LA	SL	Salavan
LA	SV	Savannakhet
LA	VT	Vientiane
LA	VT	Vientiane
LA	XA	Xaignabouli
LA	XS	Xaisomboun
LA	XE	Xekong
LA	XI	Xiangkhoang
LV	DGV	Daugavpils
LV	JEL	Jelgava
LV	JUR	Jurmala
LV	LPX	Liepaja
LV	REZ	Rezekne
LV	RIX	Riga
LV	VEN	Ventspils
LV	001	Aglonas novads
LV	002	Aizkraukles novads
LV	003	Aizputes novads
LV	004	Aknīstes novads
LV	005	Alojas novads
LV	006	Alsungas novads
LV	007	Alūksnes novads
LV	008	Amatas novads
LV	009	Apes novads
LV	010	Auces novads
LV	011	Adažu novads
LV	012	Babītes novads
LV	013	Baldones novads
LV	014	Baltinavas novads
LV	015	Balvu novads
LV	016	Bauskas novads
LV	017	Beverīnas novads
LV	018	Brocēnu novads
LV	019	Burtnieku novads
LV	020	Carnikavas novads
LV	021	Cesvaines novads
LV	022	Cēsu novads
LV	023	Ciblas novads
LV	024	Dagdas novads
LV	025	Daugavpils novads
LV	026	Dobeles novads
LV	027	Dundagas novads
LV	028	Durbes novads
LV	029	Engures novads

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
LV	030	Ērgļu novads
LV	031	Garkalnes novads
LV	032	Grobiņas novads
LV	033	Gulbenes novads
LV	034	Iecavas novads
LV	035	Ikšķiles novads
LV	036	Ilūkstes novads
LV	037	Inčukalna novads
LV	038	Jaunjelgavas novads
LV	039	Jaunpiebalgas novads
LV	040	Jaunpils novads
LV	041	Jelgavas novads
LV	042	Jēkabpils novads
LV	043	Kandavas novads
LV	044	Kārsavas novads
LV	045	Kocēnu novads
LV	046	Kokneses novads
LV	047	Krāslavas novads
LV	048	Krimuldas novads
LV	049	Krustpils novads
LV	050	Kuldīgas novads
LV	051	Ķeguma novads
LV	052	Ķekavas novads
LV	053	Lielvārdes novads
LV	054	Limbažu novads
LV	055	Līgatnes novads
LV	056	Līvānu novads
LV	057	Lubānas novads
LV	058	Ludzas novads
LV	059	Madonas novads
LV	060	Mazsalacas novads
LV	061	Mālpils novads
LV	062	Mārupes novads
LV	063	Mērsraga novads
LV	064	Naukšēnu novads
LV	065	Neretas novads
LV	066	Nīcas novads
LV	067	Ogres novads
LV	068	Olaines novads
LV	069	Ozolnieku novads
LV	070	Pārgaujas novads
LV	071	Pāvilostas novads
LV	072	Pļaviņu novads
LV	073	Preiļu novads
LV	074	Priekules novads
LV	075	Priekuļu novads
LV	076	Raunas novads
LV	077	Rēzeknes novads
LV	078	Riebiņu novads

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
LV	079	Rojas novads
LV	080	Ropažu novads
LV	081	Rucavas novads
LV	082	Rugāju novads
LV	083	Rundāles novads
LV	084	Rūjienas novads
LV	085	Salas novads
LV	086	Salacgrīvas novads
LV	087	Salaspils novads
LV	088	Saldus novads
LV	089	Saulkrastu novads
LV	090	Sējas novads
LV	091	Siguldas novads
LV	092	Skrīveru novads
LV	093	Skrundas novads
LV	094	Smiltenes novads
LV	095	Stopiņu novads
LV	096	Strenču novads
LV	097	Talsu novads
LV	098	Tērvetes novads
LV	099	Tukuma novads
LV	100	Vaiņodes novads
LV	101	Valkas novads
LV	102	Varakļānu novads
LV	103	Vārkavas novads
LV	104	Vecpiebalgas novads
LV	105	Vecumnieku novads
LV	106	Ventspils novads
LV	107	Viesītes novads
LV	108	Viļakas novads
LV	109	Viļānu novads
LV	110	Zilupes novads
LV	JKB	Jēkabpils
LV	VMR	Valmiera
LB	AK	Aakkâr
LB	BI	Al Biqā'
LB	JA	Al Janūb
LB	NA	An Nabaṭīyah
LB	AS	Ash Shimāl
LB	BH	Baalbek-Hermel
LB	BA	Beyrouth
LB	BI	Béqaa
LB	BH	B'alabak-Al Hirmil
LB	JL	Jabal Lubnān
LB	AS	Liban-Nord
LB	JA	Liban-Sud
LB	JL	Mont-Liban
LB	NA	Nabatīyé
LS	D	Berea

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
LS	C	Leribe
LS	E	Mafeteng
LS	A	Maseru
LS	F	Mohale's Hoek
LS	J	Mokhotlong
LS	H	Qacha's Nek
LS	G	Quthing
LS	K	Thaba-Tseka
LS	B	Botha-Bothe
LR	BM	Bomi
LR	BG	Bong
LR	GB	Grand Bassa
LR	CM	Grand Cape Mount
LR	GG	Grand Gedeh
LR	GK	Grand Kru
LR	LO	Lofa
LR	MG	Margibi
LR	MY	Maryland
LR	MO	Montserrado
LR	NI	Nimba
LR	RI	River Cess
LR	SI	Sinoe
LR	GP	Gbarpolu
LR	RG	River Gee
LY	JA	Al Jabal al Akhdar
LY	JU	Al Jufrah
LY	KF	Al Kufrah
LY	NQ	An Nuqat al Khams
LY	ZA	Az Zawiyah
LY	BA	Banghazi
LY	DR	Darnah
LY	MI	Misratah
LY	MQ	Murzuq
LY	SB	Sabha
LY	SR	Surt
LY	BU	Al Buṭnān
LY	GT	Ghāt
LY	JG	Al Jabal al Gharbī
LY	JI	Al Jafārah
LY	MB	Al Marqab
LY	MJ	Al Marj
LY	NL	Nālūt
LY	TB	Ṭarābulus
LY	WA	Al Wāḥāt
LY	WD	Wādī al Ḥayāt
LY	WS	Wādī ash Shāṭi'
LI	01	Balzers
LI	02	Eschen
LI	03	Gamprin

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
LI	04	Mauren
LI	05	Planken
LI	06	Ruggell
LI	07	Schaan
LI	08	Schellenberg
LI	09	Triesen
LI	10	Triesenberg
LI	11	Vaduz
LT	AL	Alytus
LT	16	Kaunas
LT	KL	Klaipeda
LT	25	Marijampole
LT	33	Panevezys
LT	44	Siauliai
LT	TA	Taurage
LT	TE	Telsiai
LT	UT	Utena
LT	58	Vilnius
LT	01	Akmenė
LT	02	Alytaus miestas
LT	04	Anykščiai
LT	05	Birštono
LT	06	Biržai
LT	07	Druskininkai
LT	08	Elektrėnai
LT	09	Ignalina
LT	10	Jonava
LT	11	Joniškis
LT	12	Jurbarkas
LT	13	Kaišiadorys
LT	14	Kalvarijos
LT	15	Kauno miestas
LT	17	Kazlų Rūdos
LT	18	Kėdainiai
LT	19	Kelmė
LT	20	Klaipėdos miestas
LT	22	Kretinga
LT	23	Kupiškis
LT	24	Lazdijai
LT	26	Mažeikiai
LT	27	Molėtai
LT	28	Neringa
LT	29	Pagėgiai
LT	30	Pakruojis
LT	31	Palangos miestas
LT	32	Panevėžio miestas
LT	34	Pasvalys
LT	35	Plungė
LT	36	Prienai

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
LT	37	Radviliškis
LT	38	Raseiniai
LT	39	Rietavo
LT	40	Rokiškis
LT	41	Šakiai
LT	42	Šalčininkai
LT	43	Šiaulių miestas
LT	45	Šilalė
LT	46	Šilutė
LT	47	Širvintos
LT	48	Skuodas
LT	49	Švenčionys
LT	52	Trakai
LT	53	Ukmergė
LT	55	Varėna
LT	56	Vilkaviškis
LT	57	Vilniaus miestas
LT	59	Visaginas
LT	60	Zarasai
LT	KU	Kauno apskritis
LT	MR	Marijampolės apskritis
LT	PN	Panevėžio apskritis
LT	SA	Šiaulių apskritis
LT	VL	Vilniaus apskritis
LU	CA	Capellen
LU	CL	Clervaux
LU	DI	Diekirch
LU	EC	Echternach
LU	ES	Esch-sur-Alzette
LU	GR	Grevenmacher
LU	LU	Luxembourg
LU	ME	Mersch
LU	RD	Redange
LU	GR	Remich
LU	VD	Vianden
LU	WI	Wiltz
MK	101	Veles
MK	102	Gradsko
MK	103	Demir Kapija
MK	104	Kavadarci
MK	105	Lozovo
MK	106	Negotino
MK	107	Rosoman
MK	108	Sveti Nikole
MK	109	Čaška
MK	201	Berovo
MK	202	Vinica
MK	203	Delčevo
MK	204	Zrnovci

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MK	205	Karbinci
MK	206	Kočani
MK	207	Makedonska Kamenica
MK	208	Pehčevo
MK	209	Probištip
MK	210	Češinovo-Obleševo
MK	211	Štip
MK	301	Vevčani
MK	303	Debar
MK	304	Debrca
MK	307	Kičevo
MK	308	Makedonski Brod
MK	310	Ohrid
MK	311	Plasnica
MK	312	Struga
MK	313	Centar Župa
MK	401	Bogdanci
MK	402	Bosilovo
MK	403	Valandovo
MK	404	Vasilevo
MK	405	Gevgelija
MK	406	Dojran
MK	407	Konče
MK	408	Novo Selo
MK	409	Radoviš
MK	410	Strumica
MK	501	Bitola
MK	502	Demir Hisar
MK	503	Dolneni
MK	504	Krivogaštani
MK	505	Kruševo
MK	506	Mogila
MK	507	Novaci
MK	508	Prilep
MK	509	Resen
MK	601	Bogovinje
MK	602	Brvenica
MK	603	Vrapčište
MK	604	Gostivar
MK	605	Želino
MK	606	Jegunovce
MK	607	Mavrovo i Rostuše
MK	608	Tearce
MK	609	Tetovo
MK	701	Kratovo
MK	702	Kriva Palanka
MK	703	Kumanovo
MK	704	Lipkovo
MK	705	Rankovce

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MK	706	Staro Nagoričane
MK	801	Aerodrom †
MK	802	Aračinovo
MK	803	Butel †
MK	804	Gazi Baba †
MK	805	Gjorče Petrov †
MK	806	Zelenikovo
MK	807	Ilinden
MK	808	Karpoš †
MK	809	Kisela Voda †
MK	810	Petrovec
MK	811	Saraj †
MK	812	Sopište
MK	813	Studeničani
MK	814	Centar †
MK	815	Čair †
MK	816	Čučer-Sandevo
MK	817	Šuto Orizari †
MG	T	Antananarivo
MG	D	Antsiranana
MG	F	Fianarantsoa
MG	M	Mahajanga
MG	A	Toamasina
MG	U	Toliara
MW	BA	Balaka
MW	BL	Blantyre
MW	CK	Chikwawa
MW	CR	Chiradzulu
MW	CT	Chitipa
MW	DE	Dedza
MW	DO	Dowa
MW	KR	Karonga
MW	KS	Kasungu
MW	LK	Likoma
MW	LI	Lilongwe
MW	MH	Machinga
MW	MG	Mangochi
MW	MC	Mchinji
MW	MU	Mulanje
MW	MW	Mwanza
MW	MZ	Mzimba
MW	NB	Nkhata Bay
MW	NK	Nkhotakota
MW	NS	Nsanje
MW	NU	Ntcheu
MW	NI	Ntchisi
MW	PH	Phalombe
MW	RU	Rumphi
MW	SA	Salima

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MW	TH	Thyolo
MW	ZO	Zomba
MW	C	Chapakati
MW	N	Northern Region
MW	NE	Neno
MW	S	Southern Region
MY	01	Johor
MY	02	Kedah
MY	03	Kelantan
MY	04	Melaka
MY	05	Negeri Sembilan
MY	06	Pahang
MY	08	Perak
MY	09	Perlis
MY	07	Pulau Pinang
MY	12	Sabah
MY	13	Sarawak
MY	10	Selangor
MY	11	Terengganu
MY	14	Wilayah Persekutuan Kuala Lumpur
MY	15	Wilayah Persekutuan Labuan
MY	16	Wilayah Persekutuan Putrajaya
MV	01	Addu
MV	03	Faadhippolhu
MV	08	Kolhumadulu
MV	26	Male Atoll
MV	12	Mulaku Atoll
MV	00	South Ari Atoll
MV	02	North Ari Atoll
MV	04	Felidhu Atoll
MV	05	Hahdhunmathi
MV	07	Thiladhunmathee Uthuruburi
MV	13	North Maalhosmadulu
MV	14	North Nilandhe Atoll
MV	17	South Nilandhe Atoll
MV	20	South Maalhosmadulu
MV	23	Thiladhunmathee Dhekunuburi
MV	24	North Miladhunmadulu
MV	25	South Miladhunmadulu
MV	27	North Huvadhu Atoll
MV	28	South Huvadhu Atoll
MV	29	Fuvammulah
MV	MLE	Male
ML	7	Gao

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
ML	1	Kayes
ML	8	Kidal
ML	2	Koulikoro
ML	5	Mopti
ML	4	Segou
ML	3	Sikasso
ML	6	Tombouctou
ML	10	Taoudénit
ML	9	Ménaka
ML	BKO	Bamako
MT	01	Attard
MT	02	Balzan
MT	03	Birgu
MT	04	Birkirkara
MT	05	Birzebbuga
MT	06	Bormla
MT	07	Dingli
MT	08	Fgura
MT	09	Floriana
MT	10	Fontana
MT	11	Gudja
MT	12	Gzira
MT	19	Iklin
MT	20	Isla
MT	21	Kalkara
MT	22	Kercem
MT	23	Kirkop
MT	24	Lija
MT	25	Luqa
MT	26	Marsa
MT	27	Marsaskala
MT	28	Marsaxlokk
MT	29	Mdina
MT	31	Mgarr
MT	32	Mosta
MT	33	Mqabba
MT	34	Msida
MT	35	Mtarfa
MT	36	Munxar
MT	37	Nadur
MT	38	Naxxar
MT	39	Paola
MT	40	Pembroke
MT	41	Pieta
MT	42	Qala
MT	43	Qormi
MT	44	Qrendi
MT	47	Safi
MT	48	San Giljan

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MT	49	San Gwann
MT	50	San Lawrenz
MT	52	Sannat
MT	53	Santa Lucija
MT	54	Santa Venera
MT	55	Siggiewi
MT	56	Sliema
MT	57	Swieqi
MT	59	Tarxien
MT	60	Valletta
MT	62	Xewkija
MT	64	Zabbar
MT	67	Zejtun
MT	68	Zurrieq
MT	13	Għajnsielem
MT	14	Għarb
MT	15	Għarghur
MT	16	Għasri
MT	17	Għaxaq
MT	18	Hamrun
MT	30	Mellieħa
MT	45	Rabat Għawdex
MT	46	Rabat Malta
MT	51	San Pawl il-Baħar
MT	58	Ta' Xbiex
MT	61	Xagħra
MT	63	Xgħajra
MT	65	Żebbuġ Għawdex
MT	66	Żebbuġ Malta
MH	ALL	Ailinglaplap
MH	ALK	Ailuk
MH	ARN	Arno
MH	AUR	Aur
MH	EBO	Ebon
MH	JAB	Jabat
MH	JAL	Jaluit
MH	KIL	Kili
MH	KWA	Kwajalein
MH	LAE	Lae
MH	LIB	Lib
MH	LIK	Likiep
MH	MAJ	Majuro
MH	MAL	Maloelap
MH	MEJ	Mejit
MH	MIL	Mili
MH	NMK	Namorik
MH	NMU	Namu
MH	RON	Rongelap
MH	UJA	Ujae

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MH	WTH	Wotho
MH	WTJ	Wotje
MH	ENI	Enewetak & Ujelang
MH	L	Ralik chain
MH	T	Ratak chain
MH	UTI	Utrök
MR	07	Adrar
MR	03	Assaba
MR	05	Brakna
MR	08	Dakhlet Nouadhibou
MR	04	Gorgol
MR	10	Guidimaka
MR	01	Hodh Ech Chargui
MR	02	Hodh El Gharbi
MR	12	Inchiri
MR	09	Tagant
MR	11	Tiris Zemmour
MR	06	Trarza
MR	13	Nuwākshūt al Gharbīyah
MR	14	Nuwākshūt ash Shamāliyah
MR	15	Nuwākshūt al Janūbīyah
MU	AG	Agalega Islands
MU	BL	Black River
MU	CC	Cargados Carajos Shoals (Saint Brandon Islands)
MU	FL	Flacq
MU	GP	Grand Port
MU	MO	Moka
MU	PA	Pamplemousses
MU	PW	Plaines Wilhems
MU	PL	Port Louis
MU	PL	Port Louis
MU	RR	Riviere du Rempart
MU	RO	Rodrigues
MU	SA	Savanne
MX	AGU	Aguascalientes
MX	BCS	Baja California Sur
MX	CAM	Campeche
MX	CHP	Chiapas
MX	CHH	Chihuahua
MX	COA	Coahuila de Zaragoza
MX	COL	Colima
MX	DUR	Durango
MX	GUA	Guanajuato
MX	GRO	Guerrero
MX	HID	Hidalgo
MX	JAL	Jalisco
MX	MEX	Mexico
MX	MIC	Michoacan de Ocampo

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MX	MOR	Morelos
MX	NAY	Nayarit
MX	NLE	Nuevo Leon
MX	OAX	Oaxaca
MX	PUE	Puebla
MX	ROO	Quintana Roo
MX	SLP	San Luis Potosi
MX	SIN	Sinaloa
MX	SON	Sonora
MX	TAB	Tabasco
MX	TAM	Tamaulipas
MX	TLA	Tlaxcala
MX	YUC	Yucatan
MX	ZAC	Zacatecas
MX	BCN	Baja California
MX	CMX	Ciudad de México
MX	QUE	Querétaro
MX	VER	Veracruz de Ignacio de la Llave
FM	TRK	Chuuk
FM	KSA	Kosrae
FM	PNI	Pohnpei
FM	YAP	Yap
MD	BA	Balti
MD	CA	Cahul
MD	CU	Chisinau
MD	ED	Edinet
MD	GA	Gagauzia
MD	OR	Orhei
MD	SO	Soroca
MD	SN	St.Âônga Nistrului
MD	UN	Ungheni
MD	AN	Anenii Noi
MD	BD	Bender [Tighina]
MD	BR	Briceni
MD	BS	Basarabeasca
MD	CL	Călărași
MD	CM	Cimișlia
MD	CR	Criuleni
MD	CS	Căușeni
MD	CT	Cantemir
MD	DO	Dondușeni
MD	DR	Drochia
MD	DU	Dubăsari
MD	FA	Fălești
MD	FL	Florești
MD	GL	Glodeni
MD	HI	Hîncești
MD	IA	Ialoveni

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MD	LE	Leova
MD	NI	Nisporeni
MD	OC	Ocnita
MD	RE	Rezina
MD	RI	Riscani
MD	SD	Soldanesti
MD	SI	Singerei
MD	ST	Straseni
MD	SV	Stefan Voda
MD	TA	Taracia
MD	TE	Telenesti
MC	FO	Fontvieille
MC	CO	La Condamine
MC	MO	Monaco-Ville
MC	MC	Monte-Carlo
MC	CL	La Colle
MC	GA	La Gare
MC	JE	Jardin Exotique
MC	LA	Larvotto
MC	MA	Malbousquet
MC	MG	Moneghetti
MC	MU	Moulins
MC	PH	Port-Hercule
MC	SD	Sainte-Dévote
MC	SO	La Source
MC	SP	Spélugues
MC	SR	Saint-Roman
MC	VR	Vallon de la Rousse
MN	073	Arhangay
MN	071	Bayan-Olgii
MN	069	Bayanhongor
MN	067	Bulgan
MN	037	Darhan uul
MN	061	Dornod
MN	057	Dzavhan
MN	065	Govi-Altay
MN	064	Govi-Sumber
MN	039	Hentiy
MN	043	Hovd
MN	041	Hovsgol
MN	053	Omnogovi
MN	035	Orhon
MN	055	Ovorhangay
MN	049	Selenge
MN	051	Suhbaatar
MN	047	Tov
MN	1	Ulanbaatar
MN	046	Uvs
MN	059	Dundgovi

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MN	063	Dornogovi
ME	01	Andrijevisa
ME	02	Bar
ME	03	Berane
ME	04	Bijelo Polje
ME	05	Budva
ME	06	Cetinje
ME	07	Danilovgrad
ME	08	Herceg-Novi
ME	09	Kolašin
ME	10	Kotor
ME	11	Mojkovac
ME	12	Nikšić
ME	13	Plav
ME	14	Pljevlja
ME	15	Plužine
ME	16	Podgorica
ME	17	Rožaje
ME	18	Šavnik
ME	19	Tivat
ME	20	Ulcinj
ME	21	Žabljak
ME	22	Gusinje
ME	23	Petnjica
ME	24	Tuzi
MA	AGD	Agadir
MA	HOC	Al Hoceima
MA	AZI	Azilal
MA	BEM	Beni Mellal
MA	BOM	Boulemane
MA	CAS	Casablanca
MA	JDI	El Jadida
MA	KES	El Kelaa des Sraghna
MA	ESM	Es Smara
MA	ESI	Essaouira
MA	FES	Fes
MA	FIG	Figuig
MA	GUE	Guelmim
MA	IFR	Ifrane
MA	KEN	Kenitra
MA	KHE	Khemisset
MA	KHN	Khenifra
MA	KHO	Khouribga
MA	LAR	Larache
MA	MAR	Marrakech
MA	MEK	Meknes
MA	NAD	Nador
MA	OUA	Ouarzazate
MA	OUI	Oujda

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MA	SAF	Safi
MA	SET	Settat
MA	SIK	Sidi Kacem
MA	TAO	Taounate
MA	TAR	Taroudannt
MA	TAT	Tata
MA	TAZ	Taza
MA	TET	Tetouan
MA	TIZ	Tiznit
MA	01	Tanger-Tétouan-Al Hoceïma
MA	02	L'Oriental
MA	03	Fès-Meknès
MA	04	Rabat-Salé-Kénitra
MA	05	Béni Mellal-Khénifra
MA	06	Casablanca-Settat
MA	07	Marrakech-Safi
MA	08	Drâa-Tafilalet
MA	09	Souss-Massa
MA	10	Guelmim-Oued Noun (EH-partial)
MA	11	Laâyoune-Sakia El Hamra (EH-partial)
MA	12	Dakhla-Oued Ed-Dahab (EH)
MA	AOU	Aousserd (EH)
MA	ASZ	Assa-Zag (EH-partial)
MA	BER	Berkane
MA	BES	Benslimane
MA	BOD	Boujdour (EH)
MA	BRR	Berrechid
MA	CHE	Chefchaouen
MA	CHI	Chichaoua
MA	CHT	Chtouka-Ait Baha
MA	DRI	Driouch
MA	ERR	Errachidia
MA	FAH	Fahs-Anjra
MA	FQH	Fquih Ben Salah
MA	GUF	Guercif
MA	HAI	El Hajeb
MA	HAO	Al Haouz
MA	INE	Inezgane-Ait Melloul
MA	JRA	Jerada
MA	LAA	Laâyoune (EH)
MA	MDF	M'diq-Fnideq
MA	MED	Médiouna
MA	MID	Midelt
MA	MOH	Mohammadia
MA	MOU	Moulay Yacoub
MA	NOU	Nouaceur
MA	ODU	Oued Ed-Dahab (EH)

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
MA	OUZ	Ouezzane
MA	RAB	Rabat
MA	REH	Rehamna
MA	SAL	Salé
MA	SEF	Sefrou
MA	SIB	Sidi Bennour
MA	SIF	Sidi Ifni
MA	SIL	Sidi Slimane
MA	SKH	Skhirate-Témara
MA	TAF	Tarfaya (EH-partial)
MA	TAI	Taurirt
MA	TIN	Tinghir
MA	TNG	Tanger-Assilah
MA	TNT	Tan-Tan (EH-partial)
MA	YUS	Yousseoufia
MA	ZAG	Zagora
MZ	P	Cabo Delgado
MZ	G	Gaza
MZ	I	Inhambane
MZ	B	Manica
MZ	MPM	Maputo
MZ	N	Nampula
MZ	A	Niassa
MZ	S	Sofala
MZ	T	Tete
MZ	Q	Zambezia
MM	07	Ayeyarwady
MM	02	Bago
MM	03	Magway
MM	04	Mandalay
MM	01	Sagaing
MM	05	Tanintharyi
MM	06	Yangon
MM	11	Kachin
MM	12	Kayah
MM	13	Kayin
MM	14	Chin
MM	15	Mon
MM	16	Rakhine
MM	17	Shan
MM	18	Nay Pyi Taw
NA	CA	Caprivi
NA	ER	Erongo
NA	HA	Hardap
NA	KH	Khomas
NA	KU	Kunene
NA	OW	Ohangwena
NA	OH	Omaheke
NA	OS	Omusati

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
NA	ON	Oshana
NA	OT	Oshikoto
NA	OD	Otjozondjupa
NA	KA	//Karas
NA	KE	Kavango East
NA	KW	Kavango West
NR	01	Aiwo
NR	02	Anabar
NR	03	Anetan
NR	04	Anibare
NR	06	Boe
NR	07	Buada
NR	08	Denigomodu
NR	09	Ewa
NR	10	Ijuw
NR	11	Meneng
NR	12	Nibok
NR	13	Uaboe
NR	14	Yaren
NR	05	Baitsi
NP	BA	Bagmati
NP	BH	Bheri
NP	DH	Dhawalagiri
NP	GA	Gandaki
NP	JA	Janakpur
NP	KA	Karnali
NP	KO	Kosi
NP	LU	Lumbini
NP	MA	Mahakali
NP	ME	Mechi
NP	NA	Narayani
NP	RA	Rapti
NP	SA	Sagarmatha
NP	SE	Seti
NP	1	Madhyamanchal
NP	2	Mid Western
NP	3	Western
NP	4	Purwanchal
NP	5	Sudur Pashchimanchal
NP	P1	Province 1
NP	P2	Province 2
NP	P5	Province 5
NP	P7	Sudūr Pashchim
NL	DR	Drenthe
NL	FL	Flevoland
NL	FR	Friesland
NL	GE	Gelderland
NL	GR	Groningen
NL	LI	Limburg

continues on next page

Table 13 - continued from previous page

Country	ISO 3166-2 Code	State Name
NL	NB	Noord Brabant
NL	NH	Noord Holland
NL	OV	Overijssel
NL	UT	Utrecht
NL	ZE	Zeeland
NL	ZH	Zuid Holland
NL	AW	Aruba (see also separate country code entry under AW)
NL	BQ1	Bonaire (see also separate country code entry under BQ)
NL	BQ2	Saba (see also separate country code entry under BQ)
NL	BQ3	Sint Eustatius (see also separate country code entry under BQ)
NL	CW	Curaçao (see also separate country code entry under CW)
NL	SX	Sint Maarten (see also separate country code entry under SX)
NZ	AUK	Auckland
NZ	BOP	Bay of Plenty
NZ	CAN	Canterbury
NZ	GIS	Gisborne
NZ	HKB	Hawke's Bay
NZ	MWT	Manawatu-Wanganui
NZ	MBH	Marlborough
NZ	NSN	Nelson
NZ	NTL	Northland
NZ	OTA	Otago
NZ	STL	Southland
NZ	TKI	Taranaki
NZ	WKO	Waikato
NZ	WGN	Wellington
NZ	WTC	West Coast
NZ	CIT	Wharekauri
NZ	TAS	Tasman
NI	AN	Atlantico Norte
NI	AS	Atlantico Sur
NI	BO	Boaco
NI	CA	Carazo
NI	CI	Chinandega
NI	CO	Chontales
NI	ES	Esteli
NI	GR	Granada
NI	JI	Jinotega

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
NI	LE	Leon
NI	MD	Madriz
NI	MN	Managua
NI	MS	Masaya
NI	MT	Matagalpa
NI	NS	Nuevo Segovia
NI	SJ	Rio San Juan
NI	RI	Rivas
NE	1	Agadez
NE	2	Diffa
NE	3	Dosso
NE	4	Maradi
NE	8	Niamey
NE	5	Tahoua
NE	6	Tillaberi
NE	7	Zinder
NG	AB	Abia
NG	FC	Abuja Federal Capital Territory
NG	AD	Adamawa
NG	AK	Akwa Ibom
NG	AN	Anambra
NG	BA	Bauchi
NG	BY	Bayelsa
NG	BE	Benue
NG	BO	Borno
NG	CR	Cross River
NG	DE	Delta
NG	EB	Ebonyi
NG	ED	Edo
NG	EK	Ekiti
NG	EN	Enugu
NG	GO	Gombe
NG	IM	Imo
NG	JI	Jigawa
NG	KD	Kaduna
NG	KN	Kano
NG	KT	Katsina
NG	KE	Kebbi
NG	KO	Kogi
NG	KW	Kwara
NG	LA	Lagos
NG	NA	Nassarawa
NG	NI	Niger
NG	OG	Ogun
NG	ON	Ondo
NG	OS	Osun
NG	OY	Oyo
NG	PL	Plateau

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
NG	RI	Rivers
NG	SO	Sokoto
NG	TA	Taraba
NG	YO	Yobe
NG	ZA	Zamfara
NO	18	Nordland
NO	03	Oslo
NO	11	Rogaland
NO	15	Møre og Romsdal
NO	21	Svalbard (Arctic Region) (see also separate country code entry under SJ)
NO	22	Jan Mayen (Arctic Region) (see also separate country code entry under SJ)
NO	30	Viken
NO	34	Innlandet
NO	38	Vestfold og Telemark
NO	42	Agder
NO	46	Vestland
NO	50	Trøndelag
NO	54	Tromssan ja Finmarkun
OM	DA	Ad Dakhiliyah
OM	WU	Al Wusta
OM	ZA	Az Zahirah
OM	MA	Masqat
OM	MU	Musandam
OM	ZU	Zufar
OM	BJ	Janūb al Bāṭinah
OM	BS	Shamāl al Bāṭinah
OM	BU	Al Buraymī
OM	SJ	Janūb ash Sharqīyah
OM	SS	Shamāl ash Sharqīyah
PK	BA	Balochistan
PK	PB	Punjab
PK	SD	Sindh
PK	GB	Gilgit-Baltistan
PK	IS	Islamabad
PK	JK	Azād Jammūñ o Kashmīr
PK	KP	Khyber Pakhtunkhwa
PW	002	Aimeliik
PW	004	Airai
PW	010	Angaur
PW	050	Hatohobei
PW	100	Kayangel
PW	150	Koror
PW	212	Melekeok
PW	214	Ngaraard
PW	218	Ngarchelong

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
PW	222	Ngardmau
PW	224	Ngatpang
PW	226	Ngchesar
PW	227	Ngeremlengui
PW	228	Ngiwal
PW	350	Peleliu
PW	370	Sonsorol
PS	BTH	Bethlehem
PS	DEB	Deir El Balah
PS	GZA	Ghazzah
PS	HBN	Hebron
PS	JEM	Jerusalem
PS	JEN	Jenin
PS	JRH	Jericho and Al Aghwar
PS	KYS	Khān Yūnis
PS	NBS	Nāblus
PS	NGZ	Shamāl Ghazzah
PS	QQA	Qalqīlyah
PS	RBH	Ramallah
PS	RFH	Rafaḥ
PS	SLT	Salfit
PS	TBS	Ṭūbās
PS	TKM	Ṭūlkarm
PA	1	Bocas del Toro
PA	4	Chiriqui
PA	2	Cocle
PA	3	Colon
PA	5	Darien
PA	6	Herrera
PA	7	Los Santos
PA	8	Panama
PA	9	Veraguas
PA	10	Panamá Oeste
PA	EM	Emberá
PA	KY	Guna Yala
PA	NB	Ngôbe-Buglé
PG	NSB	Bougainville
PG	CPM	Central
PG	CPK	Chimbu
PG	EBR	East New Britain
PG	ESW	East Sepik
PG	EHG	Eastern Highlands
PG	EPW	Enga
PG	GPK	Gulf
PG	MPM	Madang
PG	MRL	Manus
PG	MBA	Milne Bay
PG	MPL	Morobe
PG	NIK	New Ireland

continues on next page

Table 13 - continued from previous page

Country	ISO 3166-2 Code	State Name
PG	NPP	Northern
PG	SHM	Southern Highlands
PG	WBK	West New Britain
PG	WPD	Western
PG	WHM	Western Highlands
PG	HLA	Hela
PG	JWK	Jiwaka
PG	NCD	National Capital District (Port Moresby)
PG	SAN	West Sepik
PY	16	Alto Paraguay
PY	10	Alto Parana
PY	13	Amambay
PY	ASU	Asuncion
PY	19	Boqueron
PY	5	Caaguazu
PY	6	Caazapa
PY	14	Canindeyu
PY	11	Central
PY	1	Concepcion
PY	3	Cordillera
PY	4	Guaira
PY	7	Itapua
PY	8	Misiones
PY	12	Neembucu
PY	9	Paraguari
PY	15	Presidente Hayes
PY	2	San Pedro
PE	AMA	Amazonas
PE	ANC	Ancash
PE	APU	Apurimac
PE	ARE	Arequipa
PE	AYA	Ayacucho
PE	CAJ	Cajamarca
PE	CUS	Cusco
PE	HUV	Huancavelica
PE	HUC	Huanuco
PE	ICA	Ica
PE	JUN	Junin
PE	LAL	La Libertad
PE	LAM	Lambayeque
PE	LIM	Lima
PE	LOR	Loreto
PE	MDD	Madre de Dios
PE	MOQ	Moquegua
PE	PAS	Pasco
PE	PIU	Piura
PE	PUN	Puno
PE	SAM	San Martin

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
PE	TAC	Tacna
PE	TUM	Tumbes
PE	UCA	Ucayali
PE	CAL	Qallaw
PE	LMA	Municipalidad Metropolitana de Lima
PH	ABR	Abra
PH	AGN	Agusan del Norte
PH	AGS	Agusan del Sur
PH	AKL	Aklan
PH	ALB	Albay
PH	ANT	Antique
PH	APA	Apayao
PH	AUR	Aurora
PH	BAS	Basilan
PH	BAN	Bataan
PH	BTN	Batanes
PH	BTG	Batangas
PH	BEN	Benguet
PH	BIL	Biliran
PH	BOH	Bohol
PH	BUK	Bukidnon
PH	BUL	Bulacan
PH	CAG	Cagayan
PH	CAN	Camarines Norte
PH	CAS	Camarines Sur
PH	CAM	Camiguin
PH	CAP	Capiz
PH	CAT	Catanduanes
PH	CAV	Cavite
PH	CEB	Cebu
PH	DAO	Davao Oriental
PH	DAV	Davao del Norte
PH	DAS	Davao del Sur
PH	EAS	Eastern Samar
PH	GUI	Guimaras
PH	IFU	Ifugao
PH	ILN	Ilocos Norte
PH	ILS	Ilocos Sur
PH	ILI	Iloilo
PH	ISA	Isabela
PH	KAL	Kalinga
PH	LUN	La Union
PH	LAG	Laguna
PH	LAN	Lanao del Norte
PH	LAS	Lanao del Sur
PH	LEY	Leyte
PH	MAG	Maguindanao
PH	MAD	Marinduque

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
PH	MAS	Masbate
PH	MDC	Mindoro Occidental
PH	MDR	Mindoro Oriental
PH	MSC	Misamis Occidental
PH	MSR	Misamis Oriental
PH	NEC	Negros Occidental
PH	NER	Negros Oriental
PH	NSA	Northern Samar
PH	NUE	Nueva Ecija
PH	NUV	Nueva Vizcaya
PH	PLW	Palawan
PH	PAM	Pampanga
PH	PAN	Pangasinan
PH	QUE	Quezon
PH	QUI	Quirino
PH	RIZ	Rizal
PH	ROM	Romblon
PH	WSA	Samar
PH	SAR	Sarangani
PH	SIG	Siquijor
PH	SOR	Sorsogon
PH	SCO	South Cotabato
PH	SLE	Southern Leyte
PH	SUK	Sultan Kudarat
PH	SLU	Sulu
PH	SUN	Surigao del Norte
PH	SUR	Surigao del Sur
PH	TAR	Tarlac
PH	TAW	Tawi-Tawi
PH	ZMB	Zambales
PH	ZSI	Zamboanga Sibugay
PH	ZAN	Zamboanga del Norte
PH	ZAS	Zamboanga del Sur
PH	00	Pambansang Punong Rehiyon
PH	01	Rehiyon ng Iloko
PH	02	Rehiyon ng Lambak ng Kaganayan
PH	03	Rehiyon ng Gitnang Luzon
PH	05	Rehiyon ng Bikol
PH	06	Western Visayas (Region VI)
PH	07	Rehiyon ng Gitnang Bisaya
PH	08	Rehiyon ng Silangang Bisaya
PH	09	Zamboanga Peninsula (Region IX)
PH	10	Rehiyon ng Hilagang Mindanaw
PH	11	Rehiyon ng Dabaw

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
PH	12	Soccsksargen (Region XII)
PH	13	Rehiyon ng Karaga
PH	14	Nagsasariling Rehiyon ng Muslim sa Mindanaw
PH	15	Rehiyon ng Administratibo ng Kordilyera
PH	40	Rehiyon ng Calabarzon
PH	41	Rehiyon ng Mimaropa
PH	COM	Davao de Oro
PH	DIN	Pulo ng Dinagat
PH	DVO	Kanlurang Dabaw
PH	MOU	Mountain Province
PH	NCO	Kotabato
PL	02	Dolnoslaskie
PL	04	Kujawsko-Pomorskie
PL	06	Lubelskie
PL	08	Lubuskie
PL	14	Mazowieckie
PL	16	Opolskie
PL	18	Podkarpackie
PL	20	Podlaskie
PL	22	Pomorskie
PL	24	Slaskie
PL	26	Swietokrzyskie
PL	28	Warminsko-Mazurskie
PL	30	Wielkopolskie
PL	32	Zachodniopomorskie
PL	10	Łódzkie
PL	12	Małopolskie
PT	07	Évora
PT	01	Aveiro
PT	02	Beja
PT	03	Braga
PT	04	Bragança
PT	05	Castelo Branco
PT	06	Coimbra
PT	08	Faro
PT	09	Guarda
PT	10	Leiria
PT	11	Lisboa
PT	12	Portalegre
PT	13	Porto
PT	14	Santarém
PT	15	Setúbal
PT	16	Viana do Castelo
PT	17	Vila Real
PT	18	Viseu
PT	20	Região Autónoma dos Açores

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
PT	30	Região Autónoma da Madeira
QA	DA	Ad Dawhah
QA	WA	Al Wakrah
QA	RA	Ar Rayyan
QA	MS	Madinat ash Shamal
QA	US	Umm Salal
QA	KH	Al Khawr wa adh Dhakhīrah
QA	SH	Ash Shīhānīyah
QA	ZA	Az Ẓa‘āyin
RO	AB	Alba
RO	AR	Arad
RO	AG	Arges
RO	BC	Bacau
RO	BH	Bihor
RO	BN	Bistrita-Nasaud
RO	BT	Botosani
RO	BR	Braila
RO	BV	Brasov
RO	B	Bucuresti
RO	BZ	Buzau
RO	CL	Calarasi
RO	CS	Caras-Severin
RO	CJ	Cluj
RO	CT	Constanta
RO	CV	Covasna
RO	DB	Dimbovita
RO	DJ	Dolj
RO	GL	Galati
RO	GR	Giurgiu
RO	GJ	Gorj
RO	HR	Harghita
RO	HD	Hunedoara
RO	IL	Ialomita
RO	IS	Iasi
RO	IF	Ilfov
RO	MM	Maramures
RO	MH	Mehedinti
RO	MS	Mures
RO	NT	Neamt
RO	OT	Olt
RO	PH	Prahova
RO	SJ	Salaj
RO	SM	Satu-Mare
RO	SB	Sibiu
RO	SV	Suceava
RO	TR	Teleorman
RO	TM	Timis
RO	TL	Tulcea

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
RO	VL	Valcea
RO	VS	Vaslui
RO	VN	Vrancea
RU	BA	Barnaul
RU	KL	Kaluga
RU	KO	Kostroma
RU	KR	Krasnodar
RU	KK	Kursk
RU	MO	Moscow
RU	SA	Samara
RU	TA	Tambov
RU	TY	Tyumen
RU	AD	“Adygeya, Respublika”
RU	AL	“Altay, Respublika”
RU	ALT	Altayskiy kray
RU	AMU	Amurskaya oblast’
RU	ARK	Arkhangel’skaya oblast’
RU	AST	Astrakhanskaya oblast’
RU	BEL	Belgorodskaya oblast’
RU	BRY	Bryanskaya oblast’
RU	BU	“Buryatiya, Respublika”
RU	CE	Chechenskaya Respublika
RU	CHE	Chelyabinskaya oblast’
RU	CHU	Čukotskiy avtonomnyj okrug
RU	CU	Čuvaškaja Respublika
RU	DA	“Dagestan, Respublika”
RU	IN	“Ingushetiya, Respublika”
RU	IRK	Irkutskaya oblast’
RU	IVA	Ivanovskaya oblast’
RU	KAM	Kamchatskiy kray
RU	KB	Kabardino-Balkarskaya Respublika
RU	KC	Karachayevo-Cherkesskaya Respublika
RU	KDA	Krasnodarskiy kray
RU	KEM	Kemerovskaya oblast’
RU	KGD	Kaliningradskaya oblast’
RU	KGN	Kurganskaya oblast’
RU	KHA	Khabarovskiy kray
RU	KHM	Khanty-Mansiyskiy avtonomnyy okrug
RU	KIR	Kirovskaya oblast’
RU	KLU	Kalužskaja oblast’
RU	KOS	Kostromskaya oblast’
RU	KRS	Kurskaya oblast’
RU	KYA	Krasnoyarskiy kray
RU	LEN	Leningradskaya oblast’
RU	LIP	Lipetskaya oblast’
RU	MAG	Magadanskaya oblast’

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
RU	ME	“Mariy El, Respublika”
RU	MOS	Moskovskaya oblast’
RU	MOW	Moskva
RU	MUR	Murmanskaya oblast’
RU	NEN	Nenetskiy avtonomnyy okrug
RU	NGR	Novgorodskaya oblast’
RU	NIZ	Nizhegorodskaya oblast’
RU	NVS	Novosibirskaya oblast’
RU	OMS	Omskaya oblast’
RU	ORE	Orenburgskaya oblast’
RU	ORL	Orlovskaya oblast’
RU	PER	Permskiy kray
RU	PNZ	Penzenskaya oblast’
RU	PRI	Primorskiy kray
RU	PSK	Pskovskaya oblast’
RU	ROS	Rostovskaya oblast’
RU	RYA	Ryazanskaya oblast’
RU	SAK	Sakhalinskaya oblast’
RU	SAM	Samarskaya oblast’
RU	SAR	Saratovskaya oblast’
RU	SE	“Severnaya Osetiya, Respublika”
RU	SMO	Smolenskaya oblast’
RU	SPE	Sankt-Peterburg
RU	STA	Stavropol’skiy kray
RU	SVE	Sverdlovskaya oblast’
RU	TAM	Tambovskaya oblast’
RU	TOM	Tomskaya oblast’
RU	TUL	Tul’skaya oblast’
RU	TVE	Tverskaya oblast’
RU	TYU	Tyumenskaya oblast’
RU	UD	Udmurtskaya Respublika
RU	ULY	Ul’yanovskaya oblast’
RU	VGG	Volgogradskaya oblast’
RU	VLA	Vladimirskaya oblast’
RU	VLG	Vologodskaya oblast’
RU	VOR	Voronežskaja oblast’
RU	YAN	Yamalo-Nenetskiy avtonomnyy okrug
RU	YAR	Yaroslavskaya oblast’
RU	YEV	Yevreyskaya avtonomnaya oblast’
RU	ZAB	Zabaykal’skiy kray
RW	01	Ville de Kigali
RW	02	Iburasirazuba
RW	03	Northern
RW	04	Western
RW	05	Sud

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SH	AC	Ascension
SH	HL	Saint Helena
SH	TA	Tristan da Cunha
KN	01	Christ Church Nichola Town
KN	02	Saint Anne Sandy Point
KN	03	Saint George Basseterre
KN	04	Saint George Gingerland
KN	05	Saint James Windward
KN	07	Saint John Figtree
KN	08	Saint Mary Cayon
KN	10	Saint Paul Charlestown
KN	11	Saint Peter Basseterre
KN	12	Saint Thomas Lowland
KN	13	Saint Thomas Middle Island
KN	15	Trinity Palmetto Point
KN	06	Saint John Capisterre
KN	09	Saint Paul Capisterre
KN	K	Saint Kitts
KN	N	Nevis
LC	02	Castries
LC	03	Choiseul
LC	05	Dennerly
LC	07	Laborie
LC	08	Micoud
LC	10	Soufriere
LC	01	Anse la Raye
LC	06	Gros Islet
LC	11	Vieux Fort
LC	12	Canaries
VC	01	Charlotte
VC	06	Grenadines
VC	02	Saint Andrew
VC	03	Saint David
VC	04	Saint George
VC	05	Saint Patrick
WS	AA	A'ana
WS	AL	Aiga-i-le-Tai
WS	AT	Atua
WS	FA	Fa'asaleleaga
WS	GE	Gaga'emauga
WS	GI	Gagaifomauga
WS	PA	Palauli
WS	SA	Satupa'itea
WS	TU	Tuamasaga
WS	VF	Va'a-o-Fonoti
WS	VS	Vaisigano
SM	01	Acquaviva
SM	06	Borgo Maggiore
SM	02	Chiesanuova

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SM	07	Citta di San Marino
SM	03	Domagnano
SM	04	Faetano
SM	05	Fiorentino
SM	08	Montegiardino
SM	09	Serravalle
ST	P	Principe
ST	01	Água Grande
ST	02	Cantagalo
ST	03	Caué
ST	04	Lembá
ST	05	Lobata
ST	06	Mé-Zóchi
SA	14	‘Asir
SA	11	Al Bahah
SA	08	Al Hudud ash Shamaliyah
SA	12	Al Jawf
SA	05	Al Qasim
SA	01	Ar Riyad
SA	06	Ha’il
SA	10	Najran
SA	07	Tabuk
SA	02	Makkah al Mukarramah
SA	03	Al Madīnah al Munawwarah
SA	04	Ash Sharqīyah
SA	09	Jāzān
SN	DK	Dakar
SN	DB	Diourbel
SN	FK	Fatick
SN	KA	Kaolack
SN	KD	Kolda
SN	LG	Louga
SN	MT	Matam
SN	SL	Saint-Louis
SN	TC	Tambacounda
SN	TH	Thies
SN	ZG	Ziguinchor
SN	KE	Kédougou
SN	SE	Sédhiou
RS	00	Beograd
RS	01	Severnobački okrug
RS	02	Srednjobanatski okrug
RS	03	Severnobanatski okrug
RS	04	Južnobanatski okrug
RS	05	Zapadnobački okrug
RS	06	Južnobački okrug
RS	07	Sremski okrug
RS	08	Mačvanski okrug
RS	09	Kolubarski okrug

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
RS	10	Podunavski okrug
RS	11	Braničevski okrug
RS	12	Šumadijski okrug
RS	13	Pomoravski okrug
RS	14	Borski okrug
RS	15	Zaječarski okrug
RS	16	Zlatiborski okrug
RS	17	Moravički okrug
RS	18	Raški okrug
RS	19	Rasinski okrug
RS	20	Nišavski okrug
RS	21	Toplički okrug
RS	22	Pirotski okrug
RS	23	Jablanički okrug
RS	24	Pčinjski okrug
RS	25	Kosovski okrug
RS	26	Pečki okrug
RS	27	Prizrenski okrug
RS	28	Kosovsko-Mitrovački okrug
RS	29	Kosovsko-Pomoravski okrug
RS	KM	Kosovo-Metohija
RS	VO	Vojvodina
SC	02	Anse Boileau
SC	03	Anse Etoile
SC	05	Anse Royale
SC	01	Anse aux Pins
SC	06	Baie Lazare
SC	07	Baie Sainte Anne
SC	08	Beau Vallon
SC	09	Bel Air
SC	10	Bel Ombre
SC	11	Cascade
SC	12	Glacis
SC	15	La Digue
SC	16	La Riviere Anglaise
SC	17	Mont Buxton
SC	18	Mont Fleuri
SC	19	Plaisance
SC	20	Pointe La Rue
SC	21	Port Glaud
SC	22	Saint Louis
SC	23	Takamaka
SC	04	O Kap
SC	13	Grand'Anse Mahé
SC	14	Grand'Anse Praslin
SC	24	Les Mamelles
SC	25	Ros Kaiman
SC	26	Ile Perseverance I
SC	27	Ile Perseverance II

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SL	E	Eastern
SL	N	Northern
SL	S	Southern
SL	W	Western
SL	NW	North Western
SG	01	Central Singapore
SG	02	North East
SG	03	North West
SG	04	South East
SG	05	South West
SK	NI	Nitriansky
SK	TC	Trenčiansky
SK	ZI	Žilinský
SK	BC	Banskobystrický kraj
SK	BL	Bratislavský kraj
SK	KI	Košický kraj
SK	PV	Prešovský kraj
SK	TA	Trnavský kraj
SI	001	Ajdovščina
SI	002	Beltinci
SI	003	Bled
SI	004	Bohinj
SI	005	Borovnica
SI	006	Bovec
SI	007	Brda
SI	008	Brezovica
SI	009	Brežice
SI	010	Tišina
SI	011	Celje
SI	012	Cerklje na Gorenjskem
SI	013	Cerknica
SI	014	Cerkno
SI	015	Črenšovci
SI	016	Črna na Koroškem
SI	017	Črnomelj
SI	018	Destričnik
SI	019	Divača
SI	020	Dobrepolje
SI	021	Dobrova-Polhov Gradec
SI	022	Dol pri Ljubljani
SI	023	Domžale
SI	024	Dornava
SI	025	Dravograd
SI	026	Duplek
SI	027	Gorenja vas-Poljane
SI	028	Gorišnica
SI	029	Gornja Radgona
SI	030	Gornji Grad
SI	031	Gornji Petrovci

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SI	032	Grosuplje
SI	033	Šalovci
SI	034	Hrastnik
SI	035	Hrpolje-Kozina
SI	036	Idrija
SI	037	Ig
SI	038	Ilirska Bistrica
SI	039	Ivančna Gorica
SI	040	Izola
SI	041	Jesenice
SI	042	Juršinci
SI	043	Kamnik
SI	044	Kanal
SI	045	Kidričevo
SI	046	Kobarid
SI	047	Kobilje
SI	048	Kočevje
SI	049	Komen
SI	050	Koper
SI	051	Kozje
SI	052	Kranj
SI	053	Kranjska Gora
SI	054	Krško
SI	055	Kungota
SI	056	Kuzma
SI	057	Laško
SI	058	Lenart
SI	059	Lendava
SI	060	Litija
SI	061	Ljubljana
SI	062	Ljubno
SI	063	Ljutomer
SI	064	Logatec
SI	065	Loška dolina
SI	066	Loški Potok
SI	067	Luče
SI	068	Lukovica
SI	069	Majšperk
SI	070	Maribor
SI	071	Medvode
SI	072	Mengeš
SI	073	Metlika
SI	074	Mežica
SI	075	Miren-Kostanjevica
SI	076	Mislinja
SI	077	Moravče
SI	078	Moravske Toplice
SI	079	Mozirje
SI	080	Murska Sobota

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SI	081	Muta
SI	082	Naklo
SI	083	Nazarje
SI	084	Nova Gorica
SI	085	Novo Mesto
SI	086	Odranci
SI	087	Ormož
SI	088	Osilnica
SI	089	Pesnica
SI	090	Piran
SI	091	Pivka
SI	092	Podčetrtek
SI	093	Podvelka
SI	094	Postojna
SI	095	Preddvor
SI	096	Ptuj
SI	097	Puconci
SI	098	Rače-Fram
SI	099	Radeče
SI	100	Radenci
SI	101	Radlje ob Dravi
SI	102	Radovljica
SI	103	Ravne na Koroškem
SI	104	Ribnica
SI	105	Rogašovci
SI	106	Rogaška Slatina
SI	107	Rogatec
SI	108	Ruše
SI	109	Semič
SI	110	Sevnica
SI	111	Sežana
SI	112	Slovenj Gradec
SI	113	Slovenska Bistrica
SI	114	Slovenske Konjice
SI	115	Starše
SI	116	Sveti Jurij ob Ščavnici
SI	117	Šenčur
SI	118	Šentilj
SI	119	Šentjernej
SI	120	Šentjur
SI	121	Škocjan
SI	122	Škofja Loka
SI	123	Škofljica
SI	124	Šmarje pri Jelšah
SI	125	Šmartno ob Paki
SI	126	Šoštanj
SI	127	Štore
SI	128	Tolmin

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SI	129	Trbovlje
SI	130	Trebnje
SI	131	Tržič
SI	132	Turnišče
SI	133	Velenje
SI	134	Velike Lašče
SI	135	Videm
SI	136	Vipava
SI	137	Vitanje
SI	138	Vodice
SI	139	Vojnik
SI	140	Vrhnika
SI	141	Vuzenica
SI	142	Zagorje ob Savi
SI	143	Zavrč
SI	144	Zreče
SI	146	Železniki
SI	147	Žiri
SI	148	Benedikt
SI	149	Bistrica ob Sotli
SI	150	Bloke
SI	151	Braslovče
SI	152	Cankova
SI	153	Cerkvenjak
SI	154	Dobje
SI	155	Dobrna
SI	156	Dobrovnik
SI	157	Dolenjske Toplice
SI	158	Grad
SI	159	Hajdina
SI	160	Hoče-Slivnica
SI	161	Hodoš
SI	162	Horjul
SI	163	Jezersko
SI	164	Komenda
SI	165	Kostel
SI	166	Križevci
SI	167	Lovrenc na Pohorju
SI	168	Markovci
SI	169	Miklavž na Dravskem polju
SI	170	Mirna Peč
SI	171	Oplotnica
SI	172	Podlehnik
SI	173	Polzela
SI	174	Prebold
SI	175	Prevalje
SI	176	Razkrižje
SI	177	Ribnica na Pohorju
SI	178	Selnica ob Dravi

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SI	179	Sodražica
SI	180	Solčava
SI	181	Sveta Ana
SI	182	Sveti Andraž v Slovenskih goricah
SI	183	Šempeter-Vrtojba
SI	184	Tabor
SI	185	Trnovska Vas
SI	186	Trzin
SI	187	Velika Polana
SI	188	Veržej
SI	189	Vransko
SI	190	Žalec
SI	191	Žetale
SI	192	Žirovnica
SI	193	Žužemberk
SI	194	Šmartno pri Litiji
SI	195	Apače
SI	196	Cirkulane
SI	197	Kosanjevica na Krki
SI	198	Makole
SI	199	Mokronog-Trebelno
SI	200	Poljčane
SI	201	Renče-Vogrsko
SI	202	Središče ob Dravi
SI	203	Straža
SI	204	Sveta Trojica v Slovenskih goricah
SI	205	Sveti Tomaž
SI	206	Šmarješke Toplice
SI	207	Gorje
SI	208	Log-Dragomer
SI	209	Rečica ob Savinji
SI	210	Sveti Jurij v Slovenskih goricah
SI	211	Šentrupert
SI	212	Mirna
SI	213	Ankaran
SB	CE	Central
SB	CH	Choiseul
SB	GU	Guadalcanal
SB	IS	Isabel
SB	MK	Makira
SB	ML	Malaita
SB	RB	Rennell and Bellona
SB	TE	Temotu
SB	WE	Western
SB	CT	Capital Territory (Honiara)
SO	AW	Awdal

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SO	BK	Bakool
SO	BN	Banaadir
SO	BR	Bari
SO	BY	Bay
SO	GA	Galguduud
SO	GE	Gedo
SO	HI	Hiiraan
SO	JD	Jubbada Dhexe
SO	JH	Jubbada Hoose
SO	MU	Mudug
SO	NU	Nugaal
SO	SA	Sanaag
SO	SD	Shabeellaha Dhexe
SO	SH	Shabeellaha Hoose
SO	SO	Sool
SO	TO	Togdheer
SO	WO	Woqooyi Galbeed
ZA	EC	Eastern Cape
ZA	FS	Free State
ZA	GP	Gauteng
ZA	KZN	KwaZulu-Natal
ZA	LP	Limpopo
ZA	MP	Mpumalanga
ZA	NW	North West
ZA	NC	Northern Cape
ZA	WC	Western Cape
ES	AL	Álava
ES	AV	Ávila
ES	AB	Albacete
ES	A	Alicante
ES	AL	Almeria
ES	AS	Asturias
ES	BA	Badajoz
ES	IB	Baleares
ES	BA	Barcelona
ES	BU	Burgos
ES	CC	Cáceres
ES	CA	Cádiz
ES	CO	Córdoba
ES	CB	Cantabria
ES	CL	Castellón
ES	CE	Ceuta
ES	CR	Ciudad Real
ES	CU	Cuenca
ES	GI	Girona
ES	GR	Granada
ES	GU	Guadalajara
ES	H	Huelva
ES	HU	Huesca

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
ES	J	Jaén
ES	CA	La Coruña
ES	RI	La Rioja
ES	GC	Las Palmas
ES	LE	Leon
ES	MD	Madrid
ES	MA	Malaga
ES	ML	Melilla
ES	MU	Murcia
ES	NA	Navarra
ES	P	Palencia
ES	PO	Pontevedra
ES	SA	Salamanca
ES	TF	Santa Cruz de Tenerife
ES	SG	Segovia
ES	SE	Sevilla
ES	SO	Soria
ES	TE	Teruel
ES	TO	Toledo
ES	VC	Valencia
ES	VA	Valladolid
ES	ZA	Zamora
ES	Z	Zaragoza
ES	AN	Andalucía
ES	AR	Aragón
ES	B	Barcelona [Barcelona]
ES	BI	Bizkaia
ES	C	A Coruña [La Coruña]
ES	CM	Castilla-La Mancha
ES	CN	Canarias
ES	EX	Extremadura
ES	GA	Galicia [Galicia]
ES	L	Lleida [Lérida]
ES	LU	Lugo [Lugo]
ES	MC	“Murcia, Región de”
ES	NC	“Navarra, Comunidad Foral de”
ES	OR	Ourense [Orense]
ES	PV	País Vasco
ES	SS	Gipuzkoa
ES	T	Tarragona [Tarragona]
LK	1	Western Province
LK	11	Kōlumpu
LK	12	Kampahā
LK	13	Kaḷutturai
LK	2	Mattiya mākāṇam
LK	21	Mahanuvara
LK	22	Mātṭalai
LK	23	Nuwara Eliya

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
LK	3	Tēṇ mākāṇam
LK	31	Kāli
LK	32	Māttarai
LK	33	Hambantota
LK	4	Vaṭakku mākāṇam
LK	41	Yāpanaya
LK	42	Kilinochchi
LK	43	Mannārama
LK	44	Vavuniyāva
LK	45	Mullaittivu
LK	5	Nāḡēṇahira paḷāta
LK	51	Maṭṭakkaḷappu
LK	52	Ampārai
LK	53	Trincomalee
LK	6	Vayamba paḷāta
LK	61	Kuruṇāgala
LK	62	Puttalama
LK	7	Vaṭamattiya mākāṇam
LK	71	Anurātapuram
LK	72	Polonnaruwa
LK	8	Uva Province
LK	81	Patulaḷai
LK	82	Mōṇarākalai
LK	9	Sabaragamuwa Province
LK	91	Ratnapura
LK	92	Kāgalla
SD	RS	Al Bahr al Ahmar
SD	GZ	Al Jazirah
SD	KH	Al Khartum
SD	GD	Al Qadarif
SD	NW	An Nil al Abyad
SD	NB	An Nil al Azraq
SD	NO	Ash Shamaliyah
SD	DW	Gharb Darfur
SD	GK	Gharb Kurdufan
SD	DS	Janub Darfur
SD	KS	Janub Kurdufan
SD	KA	Kassala
SD	NR	Nahr an Nil
SD	DN	Shamal Darfur
SD	KN	Shamal Kurdufan
SD	SI	Sinnar
SD	DC	Wasat Dārfūr
SD	DE	Sharq Dārfūr
SR	BR	Brokopondo
SR	CM	Commewijne
SR	CR	Coronie
SR	MA	Marowijne
SR	NI	Nickerie

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
SR	PR	Para
SR	PM	Paramaribo
SR	SA	Saramacca
SR	SI	Sipaliwini
SR	WA	Wanica
SZ	HH	Hhohho
SZ	LU	Lubombo
SZ	MA	Manzini
SZ	SH	Shiselweni
SE	T	Örebro
SE	E	Östergötland
SE	K	Blekinge
SE	W	Dalarna
SE	X	Gävleborg
SE	I	Gotland
SE	N	Halland
SE	Z	Jämtland
SE	F	Jönköping
SE	H	Kalmar
SE	G	Kronoberg
SE	BD	Norrbottn
SE	D	Södermanland
SE	M	Skåne
SE	AB	Stockholm
SE	C	Uppsala
SE	S	Värmland
SE	AC	Västerbotten
SE	Y	Västernorrland
SE	U	Västmanland
SE	O	Västra Götaland
CH	AG	Aargau
CH	AR	Appenzell Ausserrhoden
CH	AI	Appenzell Innerrhoden
CH	BL	Basel-Landschaft
CH	BS	Basel-Stadt
CH	BE	Bern
CH	FR	Fribourg
CH	GE	Genève
CH	GL	Glarus
CH	GR	Graubünden
CH	JU	Jura
CH	LU	Luzern
CH	NE	Neuchâtel
CH	NW	Nidwald
CH	OW	Obwald
CH	SH	Schaffhausen
CH	SZ	Schwyz
CH	SO	Solothurn
CH	SG	St. Gallen

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
CH	TG	Thurgau
CH	TI	Ticino
CH	UR	Uri
CH	VS	Valais
CH	VD	Vaud
CH	ZH	Zürich
CH	ZG	Zug
SY	HA	Al Hasakah
SY	LA	Al Ladhikiyah
SY	QU	Al Qunaytirah
SY	RA	Ar Raqqah
SY	SU	As Suwayda
SY	DY	Dayr az Zawr
SY	DI	Dimashq
SY	HL	Halab
SY	HM	Hamah
SY	HI	Hims
SY	ID	Idlib
SY	RD	Rif Dimashq
SY	TA	Tartus
SY	DR	Dar'ā
TW	CHA	Changhua
TW	CYI	Chiayi
TW	CYQ	Chiayi
TW	HSQ	Hsinchu
TW	HSZ	Hsinchu
TW	HUA	Hualien
TW	ILA	Yilan
TW	KEE	Keelung
TW	KHH	Kaohsiung
TW	KIN	Kinmen
TW	LIE	Lienchiang
TW	MIA	Miaoli
TW	NAN	Nantou
TW	NWT	New Taipei
TW	PEN	Penghu
TW	PIF	Pingtung
TW	TAO	Taoyuan
TW	TNN	Tainan
TW	TPE	Taipei
TW	TTT	Taitung
TW	TXG	Taichung
TW	YUN	Yunlin
TJ	GB	Gorno-Badakhstan
TJ	KT	Khatlon
TJ	SU	Sughd
TJ	DU	Dushanbe
TJ	RA	nohiyahoī tobei jumhuri
TZ	01	Arusha

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
TZ	02	Dar es Salaam
TZ	03	Dodoma
TZ	04	Iringa
TZ	05	Kagera
TZ	08	Kigoma
TZ	09	Kilimanjaro
TZ	12	Lindi
TZ	26	Manyara
TZ	13	Mara
TZ	14	Mbeya
TZ	16	Morogoro
TZ	17	Mtwara
TZ	18	Mwanza
TZ	06	Pemba North
TZ	10	Pemba South
TZ	19	Pwani
TZ	20	Rukwa
TZ	21	Ruvuma
TZ	22	Shinyanga
TZ	23	Singida
TZ	24	Tabora
TZ	25	Tanga
TZ	07	Zanzibar North
TZ	11	Zanzibar South
TZ	15	Zanzibar West
TZ	27	Geita
TZ	28	Katavi
TZ	29	Njombe
TZ	30	Simiyu
TZ	31	Songwe
TH	37	Amnat Charoen
TH	15	Ang Thong
TH	14	Ayutthaya
TH	10	Bangkok
TH	31	Buriram
TH	24	Chachoengsao
TH	18	Chai Nat
TH	36	Chaiyaphum
TH	22	Chanthaburi
TH	50	Chiang Mai
TH	57	Chiang Rai
TH	20	Chon Buri
TH	86	Chumphon
TH	46	Kalasin
TH	62	Kamphaeng Phet
TH	71	Kanchanaburi
TH	40	Khon Kaen
TH	81	Krabi
TH	52	Lampang

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
TH	51	Lamphun
TH	42	Loei
TH	16	Lop Buri
TH	58	Mae Hong Son
TH	44	Maha Sarakham
TH	49	Mukdahan
TH	26	Nakhon Nayok
TH	73	Nakhon Pathom
TH	48	Nakhon Phanom
TH	30	Nakhon Ratchasima
TH	60	Nakhon Sawan
TH	80	Nakhon Si Thammarat
TH	55	Nan
TH	96	Narathiwat
TH	39	Nong Bua Lamphu
TH	43	Nong Khai
TH	12	Nonthaburi
TH	13	Pathum Thani
TH	94	Pattani
TH	82	Phangnga
TH	93	Phatthalung
TH	56	Phayao
TH	67	Phetchabun
TH	76	Phetchaburi
TH	66	Phichit
TH	65	Phitsanulok
TH	54	Phrae
TH	83	Phuket
TH	25	Prachin Buri
TH	77	Prachuap Khiri Khan
TH	85	Ranong
TH	70	Ratchaburi
TH	21	Rayong
TH	45	Roi Et
TH	27	Sa Kaeo
TH	47	Sakon Nakhon
TH	11	Samut Prakan
TH	74	Samut Sakhon
TH	75	Samut Songkhram
TH	19	Sara Buri
TH	91	Satun
TH	17	Sing Buri
TH	33	Sisaket
TH	90	Songkhla
TH	64	Sukhothai
TH	72	Suphan Buri
TH	84	Surat Thani
TH	32	Surin
TH	63	Tak

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
TH	92	Trang
TH	23	Trat
TH	34	Ubon Ratchathani
TH	41	Udon Thani
TH	61	Uthai Thani
TH	53	Uttaradit
TH	95	Yala
TH	35	Yasothon
TH	38	Bueng Kan
TH	S	Phatthaya
TL	AL	Aileu
TL	AN	Ainaro
TL	BA	Baucau
TL	BO	Bobonaro
TL	CO	Cova Lima
TL	DI	Dili
TL	ER	Ermera
TL	LA	Lautem
TL	LI	Liquica
TL	MT	Manatuto
TL	MF	Manufahi
TL	OE	Oecussi
TL	VI	Viqueque
TG	C	Centrale
TG	K	Kara
TG	M	Maritime
TG	P	Plateaux
TG	S	Savanes
TO	02	Ha'apai
TO	04	Tongatapu
TO	05	Vava'u
TO	01	'Eua
TO	03	Niuas
TT	ARI	Arima
TT	CHA	Chaguanas
TT	DMN	Diego Martin
TT	PTF	Point Fortin
TT	POS	Port of Spain
TT	PRT	Princes Town
TT	SFO	San Fernando
TT	SGE	Sangre Grande
TT	SIP	Siparia
TT	TOB	Tobago
TT	CTT	Couva-Tabaquite-Talparo
TT	MRC	Mayaro-Rio Claro
TT	PED	Penal-Debe
TT	SJL	San Juan-Laventille
TT	TUP	Tunapuna-Piarco
TN	31	Beja

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
TN	13	Ben Arous
TN	23	Bizerte
TN	81	Gabes
TN	71	Gafsa
TN	32	Jendouba
TN	41	Kairouan
TN	42	Kasserine
TN	73	Kebili
TN	53	Mahdia
TN	82	Medenine
TN	52	Monastir
TN	21	Nabeul
TN	61	Sfax
TN	34	Siliana
TN	51	Sousse
TN	83	Tataouine
TN	72	Tozeur
TN	11	Tunis
TN	22	Zaghouan
TN	12	L'Ariana
TN	14	La Manouba
TN	33	Le Kef
TN	43	Sidi Bouzid
TR	01	Adana
TR	02	Adiyaman
TR	03	Afyonkarahisar
TR	04	Agri
TR	68	Aksaray
TR	05	Amasya
TR	06	Ankara
TR	07	Antalya
TR	75	Ardahan
TR	08	Artvin
TR	09	Aydin
TR	10	Balikesir
TR	74	Bartin
TR	72	Batman
TR	69	Bayburt
TR	11	Bilecik
TR	12	Bingol
TR	13	Bitlis
TR	14	Bolu
TR	15	Burdur
TR	16	Bursa
TR	17	Canakkale
TR	18	Cankiri
TR	19	Corum
TR	20	Denizli
TR	21	Diyarbakir

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
TR	81	Duzce
TR	22	Edirne
TR	23	Elazig
TR	24	Erzincan
TR	25	Erzurum
TR	26	Eskisehir
TR	27	Gaziantep
TR	28	Giresun
TR	29	Gumushane
TR	30	Hakkari
TR	31	Hatay
TR	76	Igdir
TR	32	Isparta
TR	34	Istanbul
TR	35	Izmir
TR	46	Kahramanmaras
TR	78	Karabuk
TR	70	Karaman
TR	36	Kars
TR	37	Kastamonu
TR	38	Kayseri
TR	79	Kilis
TR	71	Kirikkale
TR	39	Kirklareli
TR	40	Kirsehir
TR	41	Kocaeli
TR	42	Konya
TR	43	Kutahya
TR	44	Malatya
TR	45	Manisa
TR	47	Mardin
TR	33	Mersin
TR	48	Mugla
TR	49	Mus
TR	50	Nevsehir
TR	51	Nigde
TR	52	Ordu
TR	80	Osmaniye
TR	53	Rize
TR	54	Sakarya
TR	55	Samsun
TR	63	Sanliurfa
TR	56	Siirt
TR	57	Sinop
TR	73	Sirnak
TR	58	Sivas
TR	59	Tekirdag
TR	60	Tokat
TR	61	Trabzon

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
TR	62	Tunceli
TR	64	Usak
TR	65	Van
TR	77	Yalova
TR	66	Yozgat
TR	67	Zonguldak
TM	A	Ahal Welayaty
TM	B	Balkan Welayaty
TM	D	Dashhowuz Welayaty
TM	L	Lebap Welayaty
TM	M	Mary Welayaty
TM	S	Aşgabat
TV	FUN	Funafuti
TV	NMG	Nanumanga
TV	NMA	Nanumea
TV	NIT	Niutao
TV	NUI	Nui
TV	NKF	Nukufetau
TV	NKL	Nukulaelae
TV	VAI	Vaitupu
UG	301	Adjumani
UG	302	Apac
UG	303	Arua
UG	201	Bugiri
UG	401	Bundibugyo
UG	402	Bushenyi
UG	202	Busia
UG	304	Gulu
UG	403	Hoima
UG	203	Iganga
UG	204	Jinja
UG	404	Kabale
UG	405	Kabarole
UG	213	Kaberamaido
UG	101	Kalangala
UG	102	Kampala
UG	205	Kamuli
UG	413	Kamwenge
UG	414	Kanungu
UG	206	Kapchorwa
UG	406	Kasese
UG	207	Katakwi
UG	112	Kayunga
UG	407	Kibaale
UG	103	Kiboga
UG	408	Kisoro
UG	305	Kitgum
UG	306	Kotido
UG	208	Kumi

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
UG	415	Kyenjojo
UG	307	Lira
UG	104	Luwero
UG	105	Masaka
UG	409	Masindi
UG	214	Mayuge
UG	209	Mbale
UG	410	Mbarara
UG	308	Moroto
UG	309	Moyo
UG	106	Mpigi
UG	107	Mubende
UG	108	Mukono
UG	311	Nakapiripirit
UG	109	Nakasongola
UG	310	Nebbi
UG	411	Ntungamo
UG	312	Pader
UG	210	Pallisa
UG	110	Rakai
UG	412	Rukungiri
UG	111	Sembabule
UG	215	Sironko
UG	211	Soroti
UG	212	Tororo
UG	113	Wakiso
UG	313	Yumbe
UG	114	Lyantonde
UG	115	Mityana
UG	116	Nakaseke
UG	117	Buikwe
UG	118	Bukomansibi
UG	119	Butambala
UG	120	Buvuma
UG	121	Gomba
UG	122	Kalungu
UG	123	Kyankwanzi
UG	124	Lwengo
UG	125	Kyotera
UG	126	Kasanda
UG	216	Amuria
UG	217	Budaka
UG	218	Bududa
UG	219	Bukedea
UG	220	Bukwo
UG	221	Butaleja
UG	222	Kaliro
UG	223	Manafwa
UG	224	Namutumba

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
UG	225	Bulambuli
UG	226	Buyende
UG	227	Kibuku
UG	228	Kween
UG	229	Luuka
UG	230	Namayingo
UG	231	Ngora
UG	232	Serere
UG	233	Butebo
UG	234	Namisindwa
UG	235	Bugweri
UG	236	Kapelebyong
UG	237	Kalaki
UG	314	Abim
UG	315	Amolatar
UG	316	Amuru
UG	317	Dokolo
UG	318	Kaabong
UG	319	Koboko
UG	320	Maracha
UG	321	Oyam
UG	322	Agago
UG	323	Alebtong
UG	324	Amudat
UG	325	Kole
UG	326	Lamwo
UG	327	Napak
UG	328	Nwoya
UG	329	Otuke
UG	330	Zombo
UG	331	Omoro
UG	332	Pakwach
UG	333	Kwania
UG	334	Nabiatuk
UG	335	Karenga
UG	336	Madi-Okollo
UG	337	Obongi
UG	416	Buliisa
UG	417	Ibanda
UG	418	Isingiro
UG	419	Kiruhura
UG	420	Buhweju
UG	421	Kiryandongo
UG	422	Kyegegwa
UG	423	Mitooma
UG	424	Ntoroko
UG	425	Rubirizi
UG	426	Sheema
UG	427	Kagadi

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
UG	428	Kakumiro
UG	429	Rubanda
UG	430	Bunyangabu
UG	431	Rukiga
UG	432	Kikuube
UG	433	Kazo
UG	434	Kitagwenda
UG	435	Rwampara
UG	C	Central
UG	E	Eastern
UG	N	Northern
UG	W	Western
UA	40	Sevastopol
UA	05	Vinnytska oblast
UA	07	Volynska oblast
UA	09	Luhanska oblast
UA	12	Dnipropetrovska oblast
UA	14	Donetska oblast
UA	18	Zhytomyrska oblast
UA	21	Zakarpatska oblast
UA	23	Zaporizka oblast
UA	26	Ivano-Frankivska oblast
UA	30	Kyiv
UA	32	Kyivska oblast
UA	35	Kirovohradska oblast
UA	43	Avtonomna Respublika Krym
UA	46	Lvivska oblast
UA	48	Mykolaivska oblast
UA	51	Odeska oblast
UA	53	Poltavska oblast
UA	56	Rivnenska oblast
UA	59	Sumska oblast
UA	61	Ternopil'ska oblast
UA	63	Kharkivska oblast
UA	65	Khersonska oblast
UA	68	Khmelnitska oblast
UA	71	Cherkaska oblast
UA	74	Chernihivska oblast
UA	77	Chernivetska oblast
AE	AJ	'Ajman
AE	AZ	Abu Zaby
AE	FU	Al Fujayrah
AE	SH	Ash Shariqah
AE	DU	Dubayy
AE	RK	R'as al Khaymah
AE	UQ	Umm al Qaywayn
GB	ABE	Aberdeen City
GB	ABD	Aberdeenshire

continues on next page

Table 13 - continued from previous page

Country	ISO 3166-2 Code	State Name
GB	ANS	Angus
GB	AGB	Argyll and Bute
GB	BDG	Barking and Dagenham
GB	BNE	Barnet
GB	BNS	Barnsley
GB	BAS	Bath and North East Somerset
GB	BDF	Bedfordshire
GB	BFS	Belfast
GB	BEX	Bexley
GB	BIR	Birmingham
GB	BBD	Blackburn with Darwen
GB	BPL	Blackpool
GB	BGW	Blaenau Gwent
GB	BOL	Bolton
GB	BRC	Bracknell Forest
GB	BRD	Bradford
GB	BEN	Brent
GB	BGE	Bridgend [Pen-y-bont ar Ogwr GB-POG]
GB	BNH	Brighton and Hove
GB	BST	"Bristol, City of"
GB	BRY	Bromley
GB	BKM	Buckinghamshire
GB	BUR	Bury
GB	CAY	Caerphilly [Caerffili GB-CAF]
GB	CLD	Calderdale
GB	CAM	Cambridgeshire
GB	CMD	Camden
GB	CRF	Cardiff [Caerdydd GB-CRD]
GB	CMN	Carmarthenshire [Sir Gaerfyrddin GB-GFY]
GB	CBF	Central Bedfordshire
GB	CGN	Ceredigion [Sir Ceredigion]
GB	CHE	Cheshire East
GB	CHW	Cheshire West and Chester
GB	CLK	Clackmannanshire
GB	CWY	Conwy
GB	CON	Cornwall
GB	COV	Coventry
GB	CRY	Croydon
GB	CMA	Cumbria
GB	DAL	Darlington
GB	DEN	Denbighshire [Sir Ddinbych GB-DDB]
GB	DBY	Derbyshire
GB	DER	Derby
GB	DEV	Devon

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
GB	DNC	Doncaster
GB	DOR	Dorset
GB	DUD	Dudley
GB	DGY	Dumfries and Galloway
GB	DND	Dundee City
GB	DUR	Durham
GB	EAL	Ealing
GB	EAY	East Ayrshire
GB	EDU	East Dunbartonshire
GB	ELN	East Lothian
GB	ERW	East Renfrewshire
GB	ERY	East Riding of Yorkshire
GB	ESX	East Sussex
GB	EDH	“Edinburgh, City of”
GB	ELS	Eilean Siar
GB	ENF	Enfield
GB	ESS	Essex
GB	FAL	Falkirk
GB	FIF	Fife
GB	FLN	Flintshire [Sir y Fflint GB-FFL]
GB	GAT	Gateshead
GB	GLG	Glasgow City
GB	GLS	Gloucestershire
GB	GRE	Greenwich
GB	GWN	Gwynedd
GB	HCK	Hackney
GB	HAL	Halton
GB	HMF	Hammersmith and Fulham
GB	HAM	Hampshire
GB	HRY	Haringey
GB	HRW	Harrow
GB	HPL	Hartlepool
GB	HAV	Havering
GB	HEF	“Herefordshire, County of”
GB	HRT	Hertfordshire
GB	HLD	Highland
GB	HIL	Hillingdon
GB	HNS	Hounslow
GB	IVC	Inverclyde
GB	AGY	Isle of Anglesey [Sir Ynys Môn GB-YNM]
GB	IOW	Isle of Wight
GB	ISL	Islington
GB	KEC	Kensington and Chelsea
GB	KEN	Kent
GB	KHL	“Kingston upon Hull, City of”
GB	KTT	Kingston upon Thames

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
GB	KIR	Kirklees
GB	KWL	Knowsley
GB	LBH	Lambeth
GB	LAN	Lancashire
GB	LDS	Leeds
GB	LEC	Leicestershire
GB	LCE	Leicester
GB	LEW	Lewisham
GB	LIN	Lincolnshire
GB	LIV	Liverpool
GB	LND	“London, City of”
GB	LUT	Luton
GB	MAN	Manchester
GB	MDW	Medway
GB	MTY	Merthyr Tydfil [Merthyr Tudful GB-MTU]
GB	MRT	Merton
GB	MDB	Middlesbrough
GB	MLN	Midlothian
GB	MIK	Milton Keynes
GB	MON	Monmouthshire [Sir Fynwy GB-FYN]
GB	MRY	Moray
GB	NTL	Neath Port Talbot [Castell-nedd Port Talbot GB-CTL]
GB	NET	Newcastle upon Tyne
GB	NWM	Newham
GB	NWP	Newport [Casnewydd GB-CNW]
GB	NFK	Norfolk
GB	NAY	North Ayrshire
GB	NEL	North East Lincolnshire
GB	NLK	North Lanarkshire
GB	NLN	North Lincolnshire
GB	NSM	North Somerset
GB	NTY	North Tyneside
GB	NYK	North Yorkshire
GB	NTH	Northamptonshire
GB	NBL	Northumberland
GB	NTT	Nottinghamshire
GB	NGM	Nottingham
GB	OLD	Oldham
GB	ORK	Orkney Islands
GB	OXF	Oxfordshire
GB	PEM	Pembrokeshire [Sir Benfro GB-BNF]
GB	PKN	Perth and Kinross
GB	PTE	Peterborough
GB	PLY	Plymouth

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
GB	POR	Portsmouth
GB	POW	Powys
GB	RDG	Reading
GB	RDB	Redbridge
GB	RCC	Redcar and Cleveland
GB	RFW	Renfrewshire
GB	RCT	“Rhonddda, Cynon, Taff [Rhonddda, Cynon,Taf]”
GB	RIC	Richmond upon Thames
GB	RCH	Rochdale
GB	ROT	Rotherham
GB	RUT	Rutland
GB	SLF	Salford
GB	SAW	Sandwell
GB	SCB	“Scottish Borders, The”
GB	SFT	Sefton
GB	SHF	Sheffield
GB	ZET	Shetland Islands
GB	SHR	Shropshire
GB	SLG	Slough
GB	SOL	Solihull
GB	SOM	Somerset
GB	SAY	South Ayrshire
GB	SGC	South Gloucestershire
GB	SLK	South Lanarkshire
GB	STY	South Tyneside
GB	STH	Southampton
GB	SOS	Southend-on-Sea
GB	SWK	Southwark
GB	SHN	St. Helens
GB	STS	Staffordshire
GB	STG	Stirling
GB	SKP	Stockport
GB	STT	Stockton-on-Tees
GB	STE	Stoke-on-Trent
GB	SFK	Suffolk
GB	SND	Sunderland
GB	SRY	Surrey
GB	STN	Sutton
GB	SWA	Swansea [Abertawe GB-ATA]
GB	SWD	Swindon
GB	TAM	Tameside
GB	TFW	Telford and Wrekin
GB	THR	Thurrock
GB	TOB	Torbay
GB	TOF	Torfaen [Tor-faen]
GB	TWH	Tower Hamlets
GB	TRF	Trafford

continues on next page

Table 13 - continued from previous page

Country	ISO 3166-2 Code	State Name
GB	VGL	"Vale of Glamorgan, The [Bro Morgannwg GB-BMG]"
GB	WKF	Wakefield
GB	WLL	Walsall
GB	WFT	Waltham Forest
GB	WND	Wandsworth
GB	WRT	Warrington
GB	WAR	Warwickshire
GB	WBK	West Berkshire
GB	WDU	West Dunbartonshire
GB	WLN	West Lothian
GB	WSX	West Sussex
GB	WSM	Westminster
GB	WGN	Wigan
GB	WIL	Wiltshire
GB	WNM	Windsor and Maidenhead
GB	WRL	Wirral
GB	WOK	Wokingham
GB	WLV	Wolverhampton
GB	WOR	Worcestershire
GB	WRX	Wrexham [Wreccsam GB-WRC]
GB	YOR	York
GB	ABC	"Armagh City, Banbridge and Craigavon"
GB	AND	Ards and North Down
GB	ANN	Antrim and Newtownabbey
GB	BCP	"Bournemouth, Christchurch and Poole"
GB	CCG	Causeway Coast and Glens
GB	DRS	Derry and Strabane
GB	FMO	Fermanagh and Omagh
GB	IOS	Isles of Scilly
GB	LBC	Lisburn and Castlereagh
GB	MEA	Mid and East Antrim
GB	MUL	Mid-Ulster
GB	NMD	"Newry, Mourne and Down"
UM	81	Baker Island
UM	84	Howland Island
UM	86	Jarvis Island
UM	67	Johnston Atoll
UM	89	Kingman Reef
UM	76	Navassa Island
UM	95	Palmyra Atoll
UM	79	Wake Island
UM	71	Midway Islands
UY	AR	Artigas
UY	CA	Canelones
UY	CL	Cerro Largo

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
UY	CO	Colonia
UY	DU	Durazno
UY	FS	Flores
UY	FD	Florida
UY	LA	Lavalleja
UY	MA	Maldonado
UY	MO	Montevideo
UY	PA	Paysandu
UY	RN	Rio Negro
UY	RV	Rivera
UY	RO	Rocha
UY	SA	Salto
UY	SJ	San Jose
UY	SO	Soriano
UY	TA	Tacuarembó
UY	TT	Treinta y Tres
UZ	AN	Andijon
UZ	BU	Buxoro
UZ	FA	Farg'ona
UZ	JI	Jizzax
UZ	NG	Namangan
UZ	NW	Navoiy
UZ	QA	Qashqadaryo
UZ	QR	Qoraqalpog'iston Repub-likasi
UZ	SA	Samarqand
UZ	SI	Sirdaryo
UZ	SU	Surxondaryo
UZ	TK	Toshkent City
UZ	TO	Toshkent Region
UZ	XO	Xorazm
VU	MAP	Malampa
VU	PAM	Penama
VU	SAM	Sanma
VU	SEE	Shefa
VU	TAE	Tafea
VU	TOB	Torba
VE	Z	Amazonas
VE	B	Anzoategui
VE	C	Apure
VE	D	Aragua
VE	E	Barinas
VE	F	Bolivar
VE	G	Carabobo
VE	H	Cojedes
VE	Y	Delta Amacuro
VE	W	Dependencias Federales
VE	I	Falcon
VE	J	Guarico

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
VE	K	Lara
VE	L	Merida
VE	M	Miranda
VE	N	Monagas
VE	O	Nueva Esparta
VE	P	Portuguesa
VE	R	Sucre
VE	S	Tachira
VE	T	Trujillo
VE	U	Yaracuy
VE	V	Zulia
VE	A	Distrito Capital
VE	X	La Guaira
VN	44	An Giang
VN	54	Bac Giang
VN	53	Bac Kan
VN	55	Bac Lieu
VN	56	Bac Ninh
VN	50	Ben Tre
VN	57	Binh Duong
VN	58	Binh Phuoc
VN	40	Binh Thuan
VN	59	Ca Mau
VN	CT	Can Tho
VN	04	Cao Bang
VN	DN	Da Nang
VN	30	Gia Lai
VN	03	Ha Giang
VN	63	Ha Nam
VN	HN	Ha Noi
VN	23	Ha Tinh
VN	61	Hai Duong
VN	HP	Hai Phong
VN	73	Hau Giang
VN	14	Hoa Binh
VN	66	Hung Yen
VN	01	Lai Chau
VN	02	Lào Cai
VN	05	Sơn La
VN	06	Yên Bái
VN	07	Tuyên Quang
VN	09	Lạng Sơn
VN	13	Quảng Ninh
VN	18	Ninh Bình
VN	20	Thái Bình
VN	21	Thanh Hóa
VN	22	Nghệ An
VN	24	Quảng Bình
VN	25	Quảng Trị

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
VN	26	Thừa Thiên-Huế
VN	27	Quảng Nam
VN	28	Kon Tum
VN	29	Quảng Ngãi
VN	31	Bình Định
VN	32	Phú Yên
VN	33	Đắk Lắk
VN	34	Khánh Hòa
VN	35	Lâm Đồng
VN	36	Ninh Thuận
VN	37	Tây Ninh
VN	39	Đồng Nai
VN	41	Long An
VN	43	Bà Rịa - Vũng Tàu
VN	45	Đồng Tháp
VN	46	Tiền Giang
VN	47	Kiến Giang
VN	49	Vĩnh Long
VN	51	Trà Vinh
VN	52	Sóc Trăng
VN	67	Nam Định
VN	68	Phú Thọ
VN	69	Thái Nguyên
VN	70	Vĩnh Phúc
VN	71	Điện Biên
VN	72	Đắk Nông
VN	SG	Hồ Chí Minh
WF	AL	Alo
WF	SG	Sigave
WF	UV	Uvea
YE	AB	Abyan
YE	DA	Ad Dali
YE	AD	Adan
YE	BA	Al Bayda
YE	HU	Al Hodaydah
YE	JA	Al Jawf
YE	MR	Al Mahrah
YE	MW	Al Mahwit
YE	AM	Amran
YE	DH	Dhamar
YE	HD	Hadramawt
YE	HJ	Hajjah
YE	IB	Ibb
YE	LA	Lahij
YE	MA	Ma'rib
YE	SD	Sa'dah
YE	SN	San'a
YE	SH	Shabwah
YE	TA	Ta'izz

continues on next page

Table 13 – continued from previous page

Country	ISO 3166-2 Code	State Name
YE	RA	Raymah
YE	SA	Amānat al ‘Aṣimah [city]
YE	SU	Arkhabīl Suquṭrá
ZM	02	Central
ZM	08	Copperbelt
ZM	03	Eastern
ZM	04	Luapula
ZM	09	Lusaka
ZM	06	North-Western
ZM	05	Northern
ZM	07	Southern
ZM	01	Western
ZM	10	Muchinga
ZW	BU	Bulawayo
ZW	HA	Harare
ZW	MA	Manicaland
ZW	MC	Mashonaland Central
ZW	ME	Mashonaland East
ZW	MW	Mashonaland West
ZW	MV	Masvingo
ZW	MN	Matabeleland North
ZW	MS	Matabeleland South
ZW	MI	Midlands
BQ	BO	Boneiru
BQ	SA	Saba
BQ	SE	Sint Eustatius
SS	BN	Northern Bahr el Ghazal
SS	BW	Western Bahr el Ghazal
SS	EC	Central Equatoria
SS	EE	Eastern Equatoria
SS	EW	Western Equatoria
SS	JG	Jonglei
SS	LK	Lakes
SS	NU	Upper Nile
SS	UY	Unity
SS	WR	Warrap

10.5 Currency Codes

Currency code	Description
AED	United Arab Emirates dirham
ALL	Albanian lek
AMD	Armenian dram
ARS	Argentine peso
AUD	Australian dollar
AZN	Azerbaijani manat
BAM	Bosnia and Herzegovina convertible mark

continues on next page

Table 14 - continued from previous page

Currency code	Description
BDT	Bangladeshi taka
BGN	Bulgarian lev
BHD	Bahraini dinar
BIF	Burundian franc
BOB	Bolivian boliviano
BRL	Brazilian real
BWP	Botswana pula
BYN	Belarusian ruble
CAD	Canadian dollar
CDF	Congolese franc
CHF	Swiss franc
CLP	Chilean peso
CNY	Chinese yuan
COP	Colombian peso
CRC	Costa Rican colón
CVE	Cape Verdean escudo
CZK	Czech koruna
DKK	Danish krone
DJF	Djiboutian franc
DLS	DLS Loyalty program
DOP	Dominican peso
DZD	Algerian dinar
EGP	Egyptian pound
ERN	Eritrean nakfa
ETB	Ethiopian birr
EUR	Euro
FIM	Finnish markka (obsolete)
GBP	Pound sterling
GEL	Georgian lari
GHS	Ghanaian cedi
GMD	Gambian dalasi
GNF	Guinean franc
GTQ	Guatemalan quetzal
HKD	Hong Kong dollar
HNL	Honduran lempira
HRK	Croatian kuna
HUF	Hungarian forint
IDR	Indonesian rupiah
ILS	Israeli new shekel
INR	Indian rupee
IRR	Iranian rial
ISK	Icelandic króna
JOD	Jordanian dinar
JPY	Japanese yen
KES	Kenyan shilling
KGS	Kyrgyzstani som
KHR	Cambodian riel
KMF	Comoro franc
KRW	Korean Republic Won

continues on next page

Table 14 - continued from previous page

Currency code	Description
KWD	Kuwaiti dinar
KZT	Kazakhstani tenge
LAK	Lao kip
LKR	Sri Lankan rupee
LSL	Lesotho loti
MAD	Moroccan dirham
MDL	Moldovan leu
MGA	Malagasy ariary
MKD	Macedonian denar
MMK	Myanmar kyat
MNT	Mongolian tögrög
MRU	Mauritanian ouguiya
MUR	Mauritian rupee
MWK	Malawian kwacha
MXN	Mexican peso
MYR	Malaysian ringgit
MZN	Mozambican metical
NAD	Namibian dollar
NGN	Nigerian naira
NIO	Nicaraguan córdoba
NOK	Norwegian krone
NZD	New Zealand dollar
OMR	Omani rial
PAB	Panamanian balboa
PEN	Peruvian sol
PHP	Philippine peso
PKR	Pakistani rupee
PLN	Polish złoty
PYG	Paraguayan guaraní
QAR	Qatari riyal
RON	Romanian Lion
RSD	Serbian dinar
RUB	Russian ruble
RWF	Rwandan franc
SAR	Saudi riyal
SCR	Seychelles rupee
SEK	Swedish krona
SGD	Singapore dollar
SHP	Saint Helena pound
SLL	Sierra Leonean leone
SRD	Surinamese dollar
STD	São Tomé and Príncipe dobra
SZL	Eswatini lilangeni
THB	Thailand Baht
TJS	Tajikistani somoni
TMT	Turkmenistani manat
TND	Tunisian dinar
TRY	Turkish lira
TWD	Taiwan dollar

continues on next page

Table 14 - continued from previous page

Currency code	Description
TZS	Tanzanian shilling
UAH	Ukrainian hryvnia
UGX	Ugandan Shilling
USD	United States Dollar
UYU	Uruguayan peso
UZS	Uzbekistani som
VES	Venezuelan bolívar soberano
VND	Vietnamese dong
WMZ	WebMoney WMZ (USD)
XAF	Central African CFA franc
XOF	West African CFA franc
ZAR	South African rand
ZMW	Zambian kwacha
BTC	Bitcoin
µBTC	Micro Bitcoin
USDT	Tether (stablecoin)
UNS	Unspecified currency (XXX)

10.6 Payment Gateway Internal Errors

Error id	Error message	Error type	Filter error code
1	MALFORMED_REQUEST	INTERNAL	
2	INVALID_CONTROL_CODE	INTERNAL	
3	MERCHANT_CONTROL_KEY_NOT_FOUND	INTERNAL	
4	ENDPOINT_ID_NOT_SPECIFIED	INTERNAL	
5	ERROR_PARSING_ENDPOINT_ID	INTERNAL	
6	ILLEGAL_STATE_FOR_RETURN	INTERNAL	
7	MERCHANT_NOT_FOUND	INTERNAL	
8	BAD_CREDIT_CARD_NUMBER	INTERNAL	
9	BAD_CREDIT_CARD_CVV	INTERNAL	
10	NO_CARD_PRINTED_NAME	INTERNAL	
17	LANGUAGE_NOT_FOUND_BY_CODE	INTERNAL	
101	ORDER_NOT_FOUND	INTERNAL	
102	INTERNAL_ERROR	INTERNAL	
103	CREATE_ORDER_ERROR	INTERNAL	
104	SESSION_VALIDATION_ERROR	INTERNAL	
105	CARD_NUMBER_ERROR	INTERNAL	
106	SESSION_STATUS_ERROR	INTERNAL	
107	WRONG_REVERSAL_AMOUNT	INTERNAL	
108	RETURN_AMOUNT_REQUIRES_CURRENCY	INTERNAL	
109	PARTIAL_REVERSAL_NOT_SUPPORTED	INTERNAL	
111	ALREADY_CAPTURED	INTERNAL	
112	CAPTURE_ALREADY_STARTED	INTERNAL	
114	WRONG_CAPTURE_AMOUNT	INTERNAL	
115	CAPTURE_AMOUNT_REQUIRES_CURRENCY	INTERNAL	
116	PARTIAL_CAPTURE_NOT_SUPPORTED	INTERNAL	

continues on next page

Table 15 - continued from previous page

Error id	Error message	Error type	Filter error code
117	ACKNOWLEDGEMENT_CALL_NETWORK_ERROR	INTERNAL	
118	ACKNOWLEDGEMENT_CALL_ERROR	INTERNAL	
119	ACKNOWLEDGEMENT_INVALID_RESPONSE	INTERNAL	
120	ACKNOWLEDGEMENT_VALIDATION_ERROR	INTERNAL	
121	CARD_DATA_NOT_FOUND	INTERNAL	
122	CARD_DATA_RETRIEVAL_ERROR	INTERNAL	
123	INVALID_CALLBACK_URL	INTERNAL	
501	PAYMENT_FORM_CANCELED	INTERNAL	
1001	ERROR_CREATING_ORDER_INFO	INTERNAL	
1002	ERROR_CREATING_TRANSACTION	INTERNAL	
1003	CREATE_TRANSACTION_ERROR	INTERNAL	
1004	CREATE_SESSION_ERROR	INTERNAL	
1005	PROCESSING_EMPTY_GATES_LIST	INTERNAL	
1006	FILTERING_IS_FAILED	INTERNAL	
1007	CHECKING_IS_FAILED	INTERNAL	
1008	SECURE_3D_FAILED	INTERNAL	1050
1009	RANDOM_SUM_CHECK_FAILED	INTERNAL	1051
1010	PHONE_CHECK_FAILED	INTERNAL	1057
1011	RANDOM_SUM_CHECK_CANCELLED	INTERNAL	
1012	PHONE_CHECK_CANCELLED	INTERNAL	
1013	SYSTEM_UNAVAILABLE	INTERNAL	
1014	CARDNO_ENROLLED_CHECK_FAILED	INTERNAL	1058
1015	SECURE_3D_TIMEOUT	INTERNAL	
1016	MPI_STATUS_CHECK_FAILED	INTERNAL	1059
1017	ECI_STATUS_CHECK_FAILED	INTERNAL	1060
1018	CAVV_STATUS_CHECK_FAILED	INTERNAL	1061
1019	GATE_DESCRIPTOR_CHECK_CANCELLED	INTERNAL	
1020	GATE_DESCRIPTOR_CHECK_FAILED	INTERNAL	1062
1021	RAPIDA_BALANCE_CHECK_FAILED	INTERNAL	
2001	PROCESSOR_BAD_MESSAGE	INTERNAL	
2002	PROCESSOR_CONNECT_TIME_OUT	INTERNAL	
2003	PROCESSOR_GATE_IO_ERROR_ON_RESPONSE	INTERNAL	
2004	PROCESSOR_CAN_NOT_GET_EXTERNAL_ID	INTERNAL	
2005	PROCESSOR_OR_GATE_NOT_CONFIGURED	INTERNAL	
2006	PROCESSOR_ERROR_ON_GATE	INTERNAL	
2007	PROCESSOR_NETWORK_PROBLEM	INTERNAL	
2008	PROCESSOR_INTERNAL_ERROR	INTERNAL	
3001	PROCESS_CALLBACK_UNKNOWN_RESULT	INTERNAL	
3002	PROCESS_CALLBACK_NOT_VERIFIED	INTERNAL	
10000	Transaction was declined	INTERNAL	
N/A	project disabled	INTERNAL	
N/A	traffic disabled	INTERNAL	
N/A	end point disabled	INTERNAL	
N/A	merchant disabled	INTERNAL	
N/A	gate not found	INTERNAL	
N/A	zero weights	INTERNAL	
N/A	native rebill support check failed	INTERNAL	
N/A	Gate recurrent processor mismatch	INTERNAL	

continues on next page

Table 15 - continued from previous page

Error id	Error message	Error type	Filter error code
N/A	Gate recurrent mid mismatch	INTERNAL	
10002	Credit card blacklisted	FILTER	1022
10003	End point limit is reached. Transaction can be processed only tomorrow.	FILTER	1014
10004	End point limit is reached. Transaction can be processed only next month.	FILTER	1015
10010	Client has bought this item already.	FILTER	1018
10011	Client (card) has bought this item already during period.	FILTER	1020
10012	Client (bin and e-mail) has bought this item already during period.	FILTER	1021
10024	To many card numbers from one IP and e-mail	FILTER	1034
10025	Customer e-mail, IP and holder filtered	FILTER	1035
10029	Credit card issuer blacklisted	FILTER	1039
10030	IP address blacklisted	FILTER	1040
10031	Email blacklisted	FILTER	1041
10067	Destination credit card blacklisted	FILTER	1077
10068	Email blacklisted	FILTER	1078
10069	Destination blacklisted	FILTER	1079
10094	Credit card blacklisted	FILTER	1104
10095	Destination credit card blacklisted	FILTER	1105
10096	IP address blacklisted	FILTER	1106
10097	Email blacklisted	FILTER	1107
10098	Destination blacklisted	FILTER	1108
10106	Merchant daily total limit reached	FILTER	1116
10107	Merchant monthly total limit reached	FILTER	1117
10108	Merchant daily 3D-Secure limit reached	FILTER	1118
10109	Merchant monthly 3D-Secure limit reached	FILTER	1119
10110	Merchant daily non 3D-Secure limit reached	FILTER	1120
10111	Merchant monthly non 3D-Secure limit reached	FILTER	1121
10116	Merchant daily total limit for processor reached	FILTER	1126
10117	Merchant monthly total limit for processor reached	FILTER	1127
10118	Merchant daily 3D-Secure limit for processor reached	FILTER	1128
10119	Merchant monthly 3D-Secure limit for processor reached	FILTER	1129
10120	Merchant daily non 3D-Secure limit for processor reached	FILTER	1130
10121	Merchant monthly non 3D-Secure limit for processor reached	FILTER	1131
10125	Destination credit card issuer blacklisted	FILTER	1135
10128	Total filter score reached threshold (100)	FILTER	1138
10130	Merchant daily payout limit reached	FILTER	1140
10131	Merchant monthly payout limit reached	FILTER	1141

continues on next page

Table 15 - continued from previous page

Error id	Error message	Error type	Filter error code
10132	Merchant daily payout limit for processor reached	FILTER	1142
10133	Merchant monthly payout limit for processor reached	FILTER	1143
10134	Limit reached	FILTER	1144
15027	Customer e-mail blacklisted for gate	FILTER	
15028	Customer fingerprint blacklisted for gate	FILTER	
15029	Customer ip-address blacklisted for gate	FILTER	
15030	Customer purpose blacklisted for gate	FILTER	
15031	Destination card number blacklisted for gate	FILTER	
15032	E-mail domain blacklisted for gate	FILTER	
15033	Source card number blacklisted for gate	FILTER	
15034	Gate loyal customer e-mail check failed	FILTER	
15035	Gate loyal customer purpose check failed	FILTER	
15036	Gate loyal destination card number check failed	FILTER	
15037	Gate loyal source card number check failed	FILTER	
15038	Destination card bin blacklisted for processor	FILTER	
15039	Destination card country blacklisted for processor	FILTER	
15040	Source card bin blacklisted for processor	FILTER	
15041	Source card country blacklisted for processor	FILTER	
10001	Duplicate request	FILTER	1007
10090	Visa rules violation for preauthorized transaction(DELAY)	FILTER	1100
10135	PSP rules violation for no CVV transaction(CANCEL)	FILTER	1145
10136	PSP rules violation for no CVV transaction(PICKUP)	FILTER	1146
15060	Transaction amount is less than gate min allowed amount	FILTER	
15019	Transaction amount is greater than gate max allowed amount	FILTER	
15062	Billing country blacklisted for gate	FILTER	
15063	IP-address country blacklisted for gate	FILTER	
15064	Destination card country blacklisted for gate	FILTER	
N/A	end point balance reached	FILTER	
N/A	all mid limits have been reached	FILTER	
N/A	gate skipped in chain for card and error message	FILTER	
N/A	gate skipped in chain for card and error	FILTER	
N/A	Source card country blacklisted for processor	FILTER	
N/A	Reached processing limits capacity please contact your account manager	FILTER	

continues on next page

Table 15 - continued from previous page

Error id	Error message	Error type	Filter error code
N/A	Reached balance limits capacity please contact your account manager	FILTER	
N/A	all gates were filtered out	FILTER	

Besides the mentioned errors, handling of system filter codes and text messages should be implemented. Full list of such filters can be found on `transaction_filters`

* each acquirer has its own error codes list, ask your manager to provide you error codes of the exact acquirer that will be used in Live mode

10.7 Language Codes

10.7.1 Two-Letter Language Codes

Country ID	Country	Code
1	Afar	aa
2	Abkhazian	ab
3	Afrikaans	af
4	Akan	ak
5	Albanian	sq
6	Amharic	am
7	Arabic	ar
8	Aragonese	an
9	Assamese	as
10	Avaric	av
11	Avestan	ae
12	Aymara	ay
13	Azerbaijani	az
14	Bashkir	ba
15	Bambara	bm
16	Basque	eu
17	Belarusian	be
18	Bengali	bn
19	Bihari languages	bh
20	Bislama	bi
21	Tibetan	bo
22	Bosnian	bs
23	Breton	br
24	Bulgarian	bg
25	Burmese	my
26	Catalan Valencian	ca
27	Czech	cs
28	Chamorro	ch
29	Chechen	ce
30	Church Slavic	cu
31	Chuvash	cv

continues on next page

Table 16 - continued from previous page

Country ID	Country	Code
32	Cornish	kw
33	Corsican	co
34	Cree	cr
35	Welsh	cy
36	Danish	da
37	German	de
38	Divehi Maldivian	dv
39	Dutch Flemish	nl
40	Dzongkha	dz
41	Greek, Modern	el
42	English	en
43	Esperanto	eo
44	Estonian	et
45	Ewe	ee
46	Faroese	fo
47	Persian	fa
48	Fijian	fj
49	Finnish	fi
50	French	fr
51	Western Frisian	fy
52	Fulah	ff
53	Georgian	ka
54	Gaelic Scottish Gaelic	gd
55	Irish	ga
56	Galician	gl
57	Manx	gv
58	Guarani	gn
59	Gujarati	gu
60	Haitian Haitian Creole	ht
61	Hausa	ha
62	Hebrew	he
63	Herero	hz
64	Hindi	hi
65	Hiri Motu	ho
66	Croatian	hr
67	Hungarian	hu
68	Armenian	hy
69	Igbo	ig
70	Icelandic	is
71	Ido	io
72	Sichuan Yi Nuosu	ii
73	Inuktitut	iu
74	Interlingue Occidental	ie
75	Interlingua	ia
76	Indonesian	id
77	Inupiaq	ik
78	Italian	it
79	Javanese	jv
80	Japanese	ja

continues on next page

Table 16 - continued from previous page

Country ID	Country	Code
81	Kalaallisut Greenlandic	kl
82	Kannada	kn
83	Kashmiri	ks
84	Kanuri	kr
85	Kazakh	kk
86	Central Khmer	km
87	Kikuyu Gikuyu	ki
88	Kinyarwanda	rw
89	Kirghiz Kyrgyz	ky
90	Komi	kv
91	Kongo	kg
92	Korean	ko
93	Kuanyama Kwanyama	kj
94	Kurdish	ku
95	Lao	lo
96	Latin	la
97	Latvian	lv
98	Limburgan Limburger Lim- burgish	li
99	Lingala	ln
100	Lithuanian	lt
101	Luxembourgish Letze- burgesch	lb
102	Luba-Katanga	lu
103	Ganda	lg
104	Marshallese	mh
105	Malayalam	ml
106	Marathi	mr
107	Macedonian	mk
108	Malagasy	mg
109	Maltese	mt
110	Mongolian	mn
111	Maori	mi
112	Malay	ms
113	Nauru	na
114	Navajo Navaho	nv
115	South Ndebele	nr
116	North Ndebele	nd
117	Ndonga	ng
118	Nepali	ne
119	Norwegian Nynorsk	nn
120	Bokmaal, Norwegian	nb
121	Norwegian	no
122	Chichewa Chewa Nyanja	ny
123	Occitan (post 1500)	oc
124	Ojibwa	oj
125	Oriya	or
126	Oromo	om
127	Ossetian Ossetic	os

continues on next page

Table 16 - continued from previous page

Country ID	Country	Code
128	Panjabi Punjabi	pa
129	Pali	pi
130	Polish	pl
131	Portuguese	pt
132	Pushto Pashto	ps
133	Quechua	qu
134	Romansh	rm
135	Romanian Moldavian Moldovan	ro
136	Rundi	rn
137	Russian	ru
138	Sango	sg
139	Sanskrit	sa
140	Sinhala Sinhalese	si
141	Slovak	sk
142	Slovenian	sl
143	Northern Sami	se
144	Samoan	sm
145	Shona	sn
146	Sindhi	sd
147	Somali	so
148	Sotho, Southern	st
149	Spanish Castilian	es
150	Sardinian	sc
151	Serbian	sr
152	Swati	ss
153	Sundanese	su
154	Swahili	sw
155	Swedish	sv
156	Tahitian	ty
157	Tamil	ta
158	Tatar	tt
159	Telugu	te
160	Tajik	tg
161	Tagalog	tl
162	Thai	th
163	Tigrinya	ti
164	Tonga (Tonga Islands)	to
165	Tswana	tn
166	Tsonga	ts
167	Turkmen	tk
168	Turkish	tr
169	Twi	tw
170	Uighur Uyghur	ug
171	Ukrainian	uk
172	Urdu	ur
173	Uzbek	uz
174	Venda	ve
175	Vietnamese	vi

continues on next page

Table 16 - continued from previous page

Country ID	Country	Code
176	Volapuk	vo
177	Walloon	wa
178	Wolof	wo
179	Xhosa	xh
180	Yiddish	yi
181	Yoruba	yo
182	Zhuang Chuang	za
183	Chinese	zh
184	Zulu	zu

10.8 Order Stage

Some APIs provide additional transaction stages, if you've receive unlisted transaction status please inform your account manager.

Stage Name	Description	Final Stage
AUTH_3D_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AUTH_3D_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_APPROVED	Transaction has been approved, final status	Yes
AUTH_CANCELLED	Transaction has been declined, final status	Yes
AUTH_CARD_CHIP_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AUTH_CARD_CHIP_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_CHAIN_DECLINED	Transaction has been declined, but processing is still in progress, non-final status	No
AUTH_CUST_EMAIL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AUTH_CUST_EMAIL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
AUTH_CUST_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AUTH_CUST_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_CUST_TWITTER_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AUTH_CUST_TWITTER_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_DECLINED	Transaction has been declined, final status	Yes
AUTH_DESCRIPTOR_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AUTH_DESCRIPTOR_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
AUTH_FAILED	Transaction has been declined, final status	Yes
AUTH_FILTERED	Transaction has been declined by fraud internal or external control systems, final status	Yes
AUTH_FILTERING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_LOAN_REGISTERING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_LOAN_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_LOYALTY_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AUTH_LOYALTY_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
AUTH_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AUTH_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_PHONE_VERIFICATED	Verification has been finished, you should continue polling, non-final status	No
AUTH_PHONE_VERIFICATING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_RND_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AUTH_RND_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_STARTING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_TEST_CHECKING	Transaction is being processed, you should continue polling, non-final status	No
AUTH_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
AV_3D_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AV_3D_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AV_APPROVED	Transaction has been approved, final status	Yes
AV_CANCELLED	Transaction has been declined, final status	Yes
AV_CHAIN_DECLINED	Transaction has been declined, but processing is still in progress, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
AV_CUST_EMAIL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AV_CUST_EMAIL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AV_CUST_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AV_CUST_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AV_CUST_TWITTER_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AV_CUST_TWITTER_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AV_DECLINED	Transaction has been declined, final status	Yes
AV_DESCRIPTOR_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AV_DESCRIPTOR_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AV_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
AV_FAILED	Transaction has been declined, final status	Yes
AV_FILTERED	Transaction has been declined by fraud internal or external control systems, final status	Yes
AV_FILTERING	Transaction is being processed, you should continue polling, non-final status	No
AV_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AV_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
AV_PHONE_VERIFICATED	Verification has been finished, you should continue polling, non-final status	No
AV_PHONE_VERIFICATING	Transaction is being processed, you should continue polling, non-final status	No
AV_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
AV_RND_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
AV_RND_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
AV_STARTING	Transaction is being processed, you should continue polling, non-final status	No
AV_TEST_CHECKING	Transaction is being processed, you should continue polling, non-final status	No
AV_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
CANCEL_APPROVED	Transaction has been approved, final status	Yes
CANCEL_DECLINED	Transaction has been declined, final status	Yes
CANCEL_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
CANCEL_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
CANCEL_REJECTED	Transaction has been declined, final status	Yes
CANCEL_STARTING	Transaction is being processed, you should continue polling, non-final status	No
CANCEL_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
CAPTURE_APPROVED	Transaction has been approved, final status	Yes
CAPTURE_DECLINED	Transaction has been declined, final status	Yes

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
CAPTURE_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
CAPTURE_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
CAPTURE_REJECTED	Transaction has been declined, final status	Yes
CAPTURE_STARTING	Transaction is being processed, you should continue polling, non-final status	No
CAPTURE_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
CHARGEBACK_APPROVED	Transaction has been approved, final status	Yes
CREATE_CM_APPROVED	Transaction has been approved, final status	Yes
CREATE_CM_CANCELLED	Transaction has been declined, final status	Yes
CREATE_CM_DECLINED	Transaction has been declined, final status	Yes
CREATE_CM_EMAIL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
CREATE_CM_EMAIL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
CREATE_CM_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
CREATE_CM_EXTERNAL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
CREATE_CM_EXTERNAL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
CREATE_CM_FAILED	Transaction has been declined, final status	Yes
CREATE_CM_FILTERED	Transaction has been declined by fraud internal or external control systems, final status	Yes
CREATE_CM_FILTERING	Transaction is being processed, you should continue polling, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
CREATE_CM_PE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
CREATE_CM_PE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
CREATE_CM_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
CREATE_CM_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
CREATE_CM_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
CREATE_CM_STARTING	Transaction is being processed, you should continue polling, non-final status	No
CREATE_CM_TWITTER_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
CREATE_CM_TWITTER_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
CREATE_CM_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
DELETE_CM_APPROVED	Transaction has been approved, final status	Yes
DELETE_CM_CANCELLED	Transaction has been declined, final status	Yes
DELETE_CM_EMAIL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
DELETE_CM_EMAIL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
DELETE_CM_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
DELETE_CM_FAILED	Transaction has been declined, final status	Yes
DELETE_CM_FILTERED	Transaction has been declined by fraud internal or external control systems, final status	Yes

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
DELETE_CM_FILTERING	Transaction is being processed, you should continue polling, non-final status	No
DELETE_CM_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
DELETE_CM_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
DELETE_CM_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
DELETE_CM_STARTING	Transaction is being processed, you should continue polling, non-final status	No
DELETE_CM_TWITTER_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
DELETE_CM_TWITTER_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
DELETE_CM_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
DISPUTE_APPROVED	Transaction has been approved, final status	Yes
FRAUD_APPROVED	Transaction has been approved, final status	Yes
INQUIRE_CM_APPROVED	Transaction has been approved, final status	Yes
INQUIRE_CM_CANCELLED	Transaction has been declined, final status	Yes
INQUIRE_CM_EMAIL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
INQUIRE_CM_EMAIL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
INQUIRE_CM_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
INQUIRE_CM_FAILED	Transaction has been declined, final status	Yes
INQUIRE_CM_FILTERED	Transaction has been declined by fraud internal or external control systems, final status	Yes

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
INQUIRE_CM_FILTERING	Transaction is being processed, you should continue polling, non-final status	No
INQUIRE_CM_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
INQUIRE_CM_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
INQUIRE_CM_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
INQUIRE_CM_STARTING	Transaction is being processed, you should continue polling, non-final status	No
INQUIRE_CM_TWITTER_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
INQUIRE_CM_TWITTER_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
INQUIRE_CM_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
MFO_SCORING_APPROVED	Transaction has been approved, final status	Yes
MFO_SCORING_DECLINED	Transaction has been declined, final status	Yes
MFO_SCORING_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
MFO_SCORING_STARTING	Transaction is being processed, you should continue polling, non-final status	No
PAN_ELIGIBILITY_APPROVED	Transaction has been approved, final status	Yes
PAN_ELIGIBILITY_CANCELLED	Transaction has been declined, final status	Yes
PAN_ELIGIBILITY_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
PAN_ELIGIBILITY_FAILED	Transaction has been declined, final status	Yes
PAN_ELIGIBILITY_FILTERED	Transaction has been declined by fraud internal or external control systems, final status	Yes

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
PAN_ELIGIBILITY_FILTERING	Transaction is being processed, you should continue polling, non-final status	No
PAN_ELIGIBILITY_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
PAN_ELIGIBILITY_REJECTED	Transaction has been declined, final status	Yes
PAN_ELIGIBILITY_STARTING	Transaction is being processed, you should continue polling, non-final status	No
PAN_ELIGIBILITY_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
PAYOUT_CHECK_VALIDATING	The availability for payout is still not confirmed, non-final status	No
PAYOUT_CHECK_VALIDATED	The availability for payout is confirmed, non-final status	No
RETRIEVAL_APPROVED	Transaction has been approved, final status	Yes
REVERSAL_APPROVED	Transaction has been approved, final status	Yes
REVERSAL_DECLINED	Transaction has been declined, final status	Yes
REVERSAL_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
REVERSAL_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
REVERSAL_REJECTED	Transaction has been declined, final status	Yes
REVERSAL_STARTING	Transaction is being processed, you should continue polling, non-final status	No
REVERSAL_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
SALE3D_END_APPROVED	Transaction has been approved, final status	Yes
SALE3D_END_DECLINED	Transaction has been declined, final status	Yes
SALE3D_END_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
SALE3D_END_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
SALE3D_END_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
SALE_3D_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
SALE_3D_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
SALE_APPROVED	Transaction has been approved, final status	Yes
SALE_AUTO_REVERSAL_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
SALE_CANCELLED	Transaction has been declined, final status	Yes
SALE_CARD_CHIP_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
SALE_CARD_CHIP_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
SALE_CHAIN_DECLINED	Transaction has been declined, but processing is still in progress, non-final status	No
SALE_CUST_EMAIL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
SALE_CUST_EMAIL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
SALE_CUST_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
SALE_CUST_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
SALE_CUST_TWITTER_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
SALE_CUST_TWITTER_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
SALE_DECLINED	Transaction has been declined, final status	Yes
SALE_DESCRIPTOR_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
SALE_DESCRIPTOR_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
SALE_EMV_PROCESSING	Transaction is being processed, sending 0200 to an Acquirer, you should continue polling, non-final status	No
SALE_EMV_VALIDATING	Transaction is being processed, waiting for 2nd Gen AC from a chip, you should send EMV Final Advice to a host	No
SALE_EMV_VALIDATED	Transaction is being processed, EMV Final Advice is received, you should continue polling, non-final status	No
SALE_EMV_ADVICE_PROCESSING	Transaction is being processed, sending 0220 to an Acquirer, you should continue polling, non-final status	No
SALE_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
SALE_FAILED	Transaction has been declined, final status	Yes
SALE_FILTERED	Transaction has been declined by fraud internal or external control systems, final status	Yes
SALE_FILTERING	Transaction is being processed, you should continue polling, non-final status	No
SALE_LOYALTY_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
SALE_LOYALTY_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
SALE_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
SALE_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
SALE_PHONE_VERIFICATED	Verification has been finished, you should continue polling, non-final status	No
SALE_PHONE_VERIFICATING	Transaction is being processed, you should continue polling, non-final status	No
SALE_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
SALE_RND_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
SALE_RND_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
SALE_STARTING	Transaction is being processed, you should continue polling, non-final status	No
SALE_TEST_CHECKING	Transaction is being processed, you should continue polling, non-final status	No
SALE_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
TRANSFER_3D_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_3D_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_APPROVED	Transaction has been approved, final status	Yes
TRANSFER_BALANCE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_BALANCE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_CANCELLED	Transaction has been declined, final status	Yes
TRANSFER_CARD_CHIP_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
TRANSFER_CARD_CHIP_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_CHAIN_DECLINED	Transaction has been declined, but processing is still in progress, non-final status	No
TRANSFER_CHECK_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_CHECK_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_CUST_EMAIL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_CUST_EMAIL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_CUST_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_CUST_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_CUST_TWITTER_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_CUST_TWITTER_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_DECLINED	Transaction has been declined, final status	Yes
TRANSFER_DESCRIPTOR_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_DESCRIPTOR_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
TRANSFER_FAILED	Transaction has been declined, final status	Yes

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
TRANSFER_FILTERED	Transaction has been declined by fraud internal or external control systems, final status	Yes
TRANSFER_FILTERING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_LOYALTY_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_LOYALTY_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_PHONE_VERIFICATED	Verification has been finished, you should continue polling, non-final status	No
TRANSFER_PHONE_VERIFICATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_RND_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
TRANSFER_RND_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_TEST_CHECKING	Transaction is being processed, you should continue polling, non-final status	No
TRANSFER_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
UPDATE_CM_APPROVED	Transaction has been approved, final status	Yes
UPDATE_CM_CANCELLED	Transaction has been declined, final status	Yes
UPDATE_CM_DECLINED	Transaction has been declined, final status	Yes

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
UPDATE_CM_EMAIL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
UPDATE_CM_EMAIL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
UPDATE_CM_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
UPDATE_CM_EXTERNAL_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
UPDATE_CM_EXTERNAL_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
UPDATE_CM_FAILED	Transaction has been declined, final status	Yes
UPDATE_CM_FILTERED	Transaction has been declined by fraud internal or external control systems, final status	Yes
UPDATE_CM_FILTERING	Transaction is being processed, you should continue polling, non-final status	No
UPDATE_CM_PE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
UPDATE_CM_PE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
UPDATE_CM_PHONE_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No
UPDATE_CM_PHONE_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
UPDATE_CM_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
UPDATE_CM_STARTING	Transaction is being processed, you should continue polling, non-final status	No
UPDATE_CM_TWITTER_VALIDATED	Transaction is being processed, internal stage have been finished, you should continue polling, non-final status	No

continues on next page

Table 17 - continued from previous page

Stage Name	Description	Final Stage
UPDATE_CM_TWITTER_VALIDATING	Transaction is being processed, you should continue polling, non-final status	No
UPDATE_CM_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No
VOID_APPROVED	Transaction has been approved, final status	Yes
VOID_ERROR	The status of transaction is unknown, please inform your account manager, final status	Yes
VOID_PROCESSING	Transaction is being processed, you should continue polling, non-final status	No
VOID_REJECTED	Transaction has been declined, final status	Yes
VOID_STARTING	Transaction is being processed, you should continue polling, non-final status	No
VOID_UNKNOWN	Transaction has been declined but something went wrong, please inform your account manager, non-final status	No

10.9 Statuses

- [Status List](#)
- [3D Secure Status List](#)
- [Random Sum Check Status List](#)

10.9.1 Status List

Status	Description
approved	Transaction is approved, final status.
declined	Transaction is declined, final status.
error	Transaction is declined but something went wrong, please inform your account manager, final status.
filtered	Transaction is declined by fraud internal or external control systems, final status.
processing	Transaction is being processed, you should continue polling, non final status.
unknown	The status of transaction is unknown, please inform your account manager, non final status.

continues on next page

Table 18 - continued from previous page

Status	Description
chain_declined	Transaction is declined in a cascading chain, non final status.

* after reconciliation transaction status could be changed even if transaction is in final status.

** if transaction status haven't been changed for one hour something went wrong, please stop polling and inform your account manager.

*** some APIs provide additional transaction statuses, if you've receive unlisted transaction status please inform your account manager.

10.9.2 3D Secure Status List

Value	Description
AUTHENTICATED	Card MPI Status Y or A.
NOT_AUTHENTICATED	Card MPI Status N.
UNSUPPORTED	Card MPI Status U.
UNKNOWN	Verification have not been performed.

10.9.3 Random Sum Check Status List

Value	Description
AUTHENTICATED	The customer has entered correct random sum.
NOT_AUTHENTICATED	The customer has entered incorrect random sum.
UNSUPPORTED	The customer has not entered random sum.
UNKNOWN	Verification have not been performed.

10.10 Transaction Filters

- [General information](#)
- [Fraud protection filters](#)
- [Technical filters](#)
 - [Detecting and preventing duplicate requests](#)
 - [Detecting and preventing duplicate invoices](#)
 - [Detecting and preventing accidental duplicate credit card number usage](#)
 - [Declined transactions](#)
 - [Preventing new transactions with source card which has previous attempt in unknown status](#)
- [Referral filters](#)

- Blacklist check (BL)
- Predefined emails whitelist check (ABL)
- Ban untrusted networks
- Credit Card Whitelist check
- Customer fingerprint
- Check reader entry mode
- Loyal Customer check
- Payment Method check
- Email Domain blacklist
- IP address Country check
- Issuer Country check
- Billing Country check
- IP address Country blacklist
- Issuer Country blacklist
- Billing Country blacklist
- Issuer Country blacklist by Payment method
- Transaction amount check
- Source Credit Card type check
- Destination Credit Card type check
- Velocity filters
 - Source Credit Card Number decline frequency for last 24 hours (daily decline limit)
 - Source Credit Card Number increasing sequence of approved transaction amounts for last 24 hours (daily rising limit)
 - Source Credit Card Number decreasing sequence of declined transaction amounts for last 24 hours (daily lessening limit)
 - Source Credit Card Number usage frequency for last 24 hours (daily limit)
 - Source Credit Card Number usage frequency for last 7 days (weekly limit)
 - Source Credit Card Number usage frequency for last month (monthly limit)
 - Destination Credit Card Number usage frequency for last 24 hours (daily limit)
 - Destination Credit Card Number usage frequency for last 7 days (weekly limit)
 - Destination Credit Card Number usage frequency for last month (monthly limit)
 - Total Credit Card Number usage frequency for last 24 hours (daily limit)
 - Total Credit Card Number usage frequency for last 7 days (weekly limit)
 - Total Credit Card Number usage frequency for last month (monthly limit)
 - Purpose usage frequency for last 24 hours (daily limit)

- Purpose usage frequency for last 7 days (weekly limit)
- Purpose usage frequency for for last month (monthly limit)
- Email usage frequency for last 24 hours (daily limit)
- Email usage frequency for last 7 days (weekly limit)
- Email usage frequency for last month (monthly limit)
- IP address usage frequency for last 24 hours (daily limit)
- IP address usage frequency for last 7 days (weekly limit)
- IP address usage frequency for last month (monthly limit)
- Source Credit Card Number usage frequency for Purpose
- Source Credit Card Number usage frequency for Email address
- Source Credit Card Number usage frequency for First and Last name
- Source Credit Card Number usage frequency for Destination Credit Card number
- Source Credit Card Number usage frequency for Email or IP address
- Source Credit Card Number usage frequency
- Customer IP address usage frequency
- Credit Card number already used from another IP address
- Credit Card number already used from another Country
- Credit Card number already used with another Email
- Credit Card number already used with another Purpose
- Credit Card number already used with another Cardholder name
- Customer IP address already used by another Cardholder
- Customer Email address already used by another Cardholder
- Source Credit Card Number approved transaction interval
- Source Credit Card Number declined transaction interval
- Source Credit Card Number Issuer Country change frequency for current Purpose
- Reversal frequency
- Fingerprint usage frequency for last 24 hours (daily limit)
- Fingerprint usage frequency for last 7 days (weekly limit)
- Fingerprint usage frequency for last month (monthly limit)
- Source Credit Card Number usage frequency for Fingerprint
- Source Credit Card number Issuer Country change frequency for current Device Fingerprint
- Destination Credit Card Number usage frequency for Device fingerprint
- Destination Credit Card number Issuer Country change frequency for current Device fingerprint

- Email usage frequency for Device fingerprint
- Purpose usage frequency for Device fingerprint
- Account Number usage frequency for last 24 hours (daily limit)
- Account Number usage frequency for last 7 days (weekly limit)
- Account Number usage frequency for last month (monthly limit)
- Source Credit Card Number decline frequency for last week (weekly decline limit)
- Source Credit Card Number usage frequency per Email address for last 24 hours (daily limit)
- Source Credit Card Number usage frequency per Email address for last 7 days (weekly limit)
- Source Credit Card Number usage frequency per Email address for last month (monthly limit)
- Source Credit Card Number usage frequency for last N days
- Destination Credit Card Number usage frequency for last N days
- Total Credit Card Number usage frequency for last N days
- Purpose usage frequency for last N days
- Email usage frequency for last N days
- IP address usage frequency for last N days
- Fingerprint usage frequency for last N days
- Account Number usage frequency for last N days
- Source Credit Card Number decline frequency for last month (monthly decline limit)
- Destination Credit Card Number decline frequency for last 24 hours (daily decline limit)
- Destination Credit Card Number decline frequency for last week (weekly decline limit)
- Destination Credit Card Number decline frequency for last month (monthly decline limit)
- Total Credit Card Number decline frequency for last 24 hours (daily decline limit)
- Total Credit Card Number decline frequency for last week (weekly decline limit)
- Total Credit Card Number decline frequency for last month (monthly decline limit)
- Customer IP address anonymous VPN
- Customer IP address anonymous
- Customer IP Hosting Provider
- Customer IP Public Proxy
- Customer IP Residential Proxy

- Customer IP Tor Exit Node
- Customer static IP score
- Customer IP user count
- Customer IP user type
- Consistency filters
 - Customer IP address Country differs from Issuing Country
 - Customer name differs from Cardholder name
 - Customer IP address differs from IP address used for 3-D Secure validation
 - Customer birthday check
 - Source Credit Card number expiration date check
 - The 6+4 customer cards differs from the 6+4 passed in purpose
- Custom business validations
 - Transaction amount changing for Purpose
 - Authorization reattempts requirements EMEA
 - Visa Preauthorized Transaction Decline Response requirements
 - MCC 6211 restrictions
 - MCC 7995 restrictions
 - CDB processing restrictions
- Customer Level filters
 - Customer id usage frequency for last 24 hours (daily limit)
 - Customer id usage frequency for last 7 days (weekly limit)
 - Customer id usage frequency for last month (monthly limit)

10.10.1 General information

Transaction filters in System are intended for rejection of certain transactions on various reasons. For example, there are filters to check fraud transactions, or to check if the issuer-bank or card number is included in the White List, etc.

The Transaction filters are managed on a Project-level of the System. To set filters for the Project navigate to Fraud protection filters in the Project menu.

The icon to the left from the filter name displays it's status. To turn a filter ON/OFF click on this icon.

Filter's details are available by clicking on the Configure button.

The number (or multiple numbers) in "**Error codes**" is called decline-code, or the code of transaction's rejection reason. This code will be displayed on the Orders screen if the

transaction will get the Filtered status.

Some filters have Scoring. Scoring allows more flexible approach in filtering system and fraud control. Each filter's score can be set from 0 to 100. Transaction goes through different filters and each triggered filter adds it's own score to transaction. If transaction reaches 100 score - it gets Filtered; if not - it passes through. Turned OFF filters have 0 score.

Filters are also available on Endpoint-level of the System. Filter settings on Endpoint-level override Project-level settings.

Some Filter parameters can be added only from the Order information screen.

10.10.2 Fraud protection filters

Rule name	Rule description
Technical filters	Technical filters compare two or more merchant requests to detect and prevent duplicate invoice payments. All filters in this category could be applied to requests in a short period of time since the moment of the transaction received by our system.
Referral filters	Referral checks allow to establish block and trust lists of both good and bad transaction attributes, affecting the risk score based on a known trend on many different customer attributes.
Velocity filters	Velocity checks allow merchants to set velocity thresholds on various customer attributes, controlling how often a customer can attempt transactions. These checks are intended to identify high-speed fraud attacks. Velocity Rules are calculated at the merchant account level. If a merchant has several merchant accounts under their company account, velocity counts do not aggregate across the entire company if not specified additionally. For example: A single credit card is used for 2 transactions in merchant account A and 3 transactions in Merchant Account B. The Velocity Rule counts 2 in Account A and 3 in Account B. An abandonment of the shopper after redirecting to a payment method or 3-D Secure is counted as an attempt and adds to the count for velocity rules as declined transaction. But not all of these abandoned attempts can be found in the payment list, it depends on the exact integration with PSP.
Consistency filters	Consistency checks compare two or more transaction attributes with each other.
Custom business validations	Additional validations and risk profiles defined by the manager.

10.10.3 Technical filters

Detecting and preventing duplicate requests

Merchant request with the same request parameters will be filtered out

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Checking interval in seconds	max interval in seconds to check duplicate request	Type: int Default: 15

Error codes

#	Code	Name
10001	1007	Duplicate request

Detecting and preventing duplicate invoices

Merchant request with the same client order ID will be filtered out

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Checking interval in seconds	max interval in seconds to check duplicate invoices, set to 0 for infinity	Type: int Default: 30
Skip unapproved transactions	Y - to skip sessions in Error, Failed, Limited or Rejected status status, N - otherwise	Type: String Default: Y
Skip error transactions	Y - to skip to skip sessions in Filtered, Declined or Cancelled status status, N - otherwise	Type: String Default: Y

Error codes

#	Code	Name
10048	1058	Duplicate invoice

Detecting and preventing accidental duplicate credit card number usage

Customer request for the current merchant with the same credit card number will be filtered out

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Checking interval in seconds	max interval in seconds to check duplicate credit card number usage, values more then 60 seconds are ignored	Type: int Default: 30
Skip unapproved transactions	Y - to skip sessions in Filtered or Declined status, N - otherwise	Type: String Default: Y

Error codes

#	Code	Name
10087	1097	Duplicate credit card

Declined transactions

This check fires when the number of declined transactions associated with exact Credit Card number and Invoice number exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction, all transactions dates are truncated to minutes during window calculation. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 declines in 60 minutes, it fires on the 11th decline in 60 minutes. Counts transactions for Account verification, Sale, Preauth or Transfer transactions in the Filtered or Declined status. The limit is calculated for the current End point if parameter "for all merchant projects" set to N, or for all Merchant Projects if parameter set to Y

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Checking interval in minutes	time frame to calculate declines count in minutes, values more then 24 hours are ignored	Type: int Default: 30

continues on next page

Table 28 - continued from previous page

Name	Description	Value
Maximum declines count	maximum number of declined or filtered transactions allowed	Type: int Default: 2
For all Merchant projects	Y - to check transactions for all projects of the current Merchant, otherwise check transactions for current end point only	Type: String Default: N

Error codes

#	Code	Name
10009	1013	Too many declines for the same credit card number and invoice

Preventing new transactions with source card which has previous attempt in unknown status

This check fires when customer tries to perform new card transaction while last transaction with the same source card still has non-final unknown status for past N minutes (max 1 hour). Counts Sale, Preauth or Transfer transactions in unknown status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Checking interval in minutes	Max interval in minutes to check existing unknown operation	Type: Int Default: 60
For all merchant projects	Y - to check transactions for all projects of the current merchant; N - check transactions for current end point only	Type: String Default: N

Error codes

#	Code	Name
10259	1269	Customer card has previous transactions in unknown status

10.10.4 Referral filters

When transaction is filtered by merchant blacklist, the API response message will have the following structure: "Transaction declined - please contact support with the following code: {error code}:{error #}" This message is relevant only for merchant black lists and can be displayed to the customer instead of actual filtering reason.

Blacklist check (BL)

Allows blacklisting of specified clients based on various criteria such as email, IP address, etc

Score: No

Enabled by default: Y

Parameters

Name	Description	Value
For all merchant projects	Y - to check blacklists for all projects of the current merchant, otherwise check blacklists for current project only	Type: String Default: Y

Error codes

#	Code	Name
10002	1022	Credit card blacklisted
10094	1104	Credit card blacklisted
10029	1039	Credit card issuer blacklisted
10067	1077	Destination credit card blacklisted
10095	1105	Destination credit card blacklisted
10125	1135	Destination credit card issuer blacklisted
10030	1040	IP address blacklisted
10096	1106	IP address blacklisted
10031	1041	Email blacklisted
10097	1107	Email blacklisted
10069	1079	Destination blacklisted
10098	1108	Destination blacklisted
10137	1147	Billing country blacklisted for merchant
10138	1148	IP-address country blacklisted for merchant
10139	1149	Customer e-mail blacklisted for merchant
10140	1150	Customer fingerprint blacklisted for merchant
10141	1151	Customer ip-address blacklisted for merchant
10142	1152	Customer purpose blacklisted for merchant
10143	1153	Destination card bin blacklisted for merchant

continues on next page

Table 33 - continued from previous page

#	Code	Name
10144	1154	Destination card country blacklisted for merchant
10145	1155	Destination card number blacklisted for merchant
10146	1156	Destination card type blacklisted for merchant
10147	1157	E-mail domain blacklisted for merchant
10148	1158	Source card bin blacklisted for merchant
10149	1159	Source card country blacklisted for merchant
10150	1160	Source card number blacklisted for merchant
10151	1161	Source card type blacklisted for merchant

Predefined emails whitelist check (ABL)

Allows processing for trusted customers only. Different merchants have different definitions of a trusted customer; this filter allows processing for customers with emails uploaded by ABL batch only. Transactions for customers which emails are not listed in ABL will be filtered out. Filter will be applied for chosen countries only, for all other countries check will be ignored

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
Issuer country identifier	apply filter for selected countries only, country defined by credit card issuer	Type: String Default: *
IP country identifier	apply filter for selected countries only, country defined by customer IP	Type: String Default: *
For all merchant projects	Y - to check customer email in email anti-blacklists for all projects of the current merchant, otherwise check in email list for current project only	Type: String Default: Y

Error codes

#	Code	Name
10068	1078	Email blacklisted

Ban untrusted networks

Allows to make block lists based on the specific IP address ranges of the customer. Merchants are able to submit IP address ranges in either IPv4 or IPv6 format via CSV upload.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Customer ip ranges	file format example: 192.168.0.0,192.168.255.255 fd00:0000:0000:0000:0000:0000:0000,fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff	Type: File

Error codes

#	Code	Name
10034	1044	Untrusted network

Credit Card Whitelist check

Allows ignoring all other fraud filters for selected credit cards. Sometimes customer's behavior can lead to the unfortunate situation where a shopper is completely unable to process transactions. You can whitelist a customer's credit card so they can successfully process their transaction. White list will be applied only before total transactions amount for the last month for this credit will not reach the limit specified by filter parameters. White list could be specified for: the exact source card number by manager and merchant, the exact destination card number by merchant or the whole source card number issuer BIN range.

Score: N/A

Enabled by default: Y

Parameters

Name	Description	Value
Up to amount	maximum total transactions amount for the last month for this credit card to allow credit card to be whitelisted. If this limit reached, whitelist will be ignored	Type: Decimal Default: 99999999

continues on next page

Table 38 - continued from previous page

Name	Description	Value
For all merchant projects	current total transactions amount for the last month for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with "up to amount" value	Type: String Default: Y

Customer fingerprint

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
Add to black list threshold		Type: Decimal Default: 10.0
Block transaction threshold		Type: Decimal Default: 3.5

Error codes

#	Code	Name
10035	1045	Fraud suspicious activity
10039	1049	Fraud suspicious activity

Check reader entry mode

If reader entry mode not in allowed list and this list is configured - filter declines transaction

Score: No

Enabled by default: N

Error codes

#	Code	Name
10088	1098	Incorrect reader entry mode

Loyal Customer check

System allows to manage following three types of predefined clients lists:

- the lists of the cards managed by Manager
- the lists of the clients' names and email addresses managed by Merchant
- the lists of the clients' names and phone numbers managed by Merchant

In the typical flow the filter distinguishes the credit cards transactions and the uncontested transactions. The credit cards transactions are initially disabled in the system (in case the list of predefined clients was not loaded). If the uncontested transaction was made, e.g. e-check transaction the client's attributes like family name, email and phone number are automatically added to anti black lists. If the system gets the credit card transaction with the same attributes (at least two of them) the transaction is allowed and the card number is put under inspection. The inspection period is defined by the Manager and is 90 days by default. If there are no refunds or chargebacks for this card within the period of inspection all the attributes of the credit card transactions are added to anti black lists. If there are negatives for the credit card the client is marked unreliable and all subsequent transactions with similar attributes will be denied irregardless of what payment system was used. The elements of this list can have four statuses:

- data not available
- the attribute is in the anti black list: negative activity not present
- the attribute is in the anti black list: returns are present
- the attribute is in the anti black list: fraud is detected

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Strategy	GLL - accept only credit cards for loyal clients for whom period of disloyalty have passed MLL+GLL - accept both credit cards and APM's for loyal clients	Type: String Default: MLL+GLL
Credit card period of disloyalty	the period to consider credit card transaction as suspicious one, during this period transaction from this credit card would be accepted only if other attributes of it could be found in merchant loyalty lists by name and phone or email, after this period transaction would be accepted for any manager project by card number only, no matter if its other attributes could be found in merchant loyalty lists or not	Type: int Default: 90

continues on next page

Table 42 - continued from previous page

Name	Description	Value
Deny type	customer attributes will be removed from both merchant and manager loyalty lists in case of getting F - fraud transaction only R - reversal transaction only F+R - both fraud and reversal transactions	Type: String Default: F+R

Error codes

#	Code	Name
Disloyal customer	10046	1056

Payment Method check

This check can be used to apply higher or lower risk scores to transactions based on the payment method used. Some payment methods have higher or lower correlations with fraud. This check allows merchants to affect the risk score based on the payment method behavior that have noted with their fraudsters and good users

Score: Ind

Enabled by default: N

Error codes

#	Code	Name
10047	1057	Payment method is restricted

Email Domain blacklist

This referral list allows the merchant to make block lists based on the domain of the provided customer email address. By default this list includes one time email domains collected on regular basis.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
Customer email domains	file format example: google.com hot-mail.com	Type: File

Error codes

#	Code	Name
10033	1043	Email server blacklisted

IP address Country check

This referral list allows the merchant to process transactions only for selected countries based on country of the customer IP address. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Country identifier	comma separated country identifiers list	Type: String

Error codes

#	Code	Name
10014	1024	Country not in trust list

Issuer Country check

This referral list allows the merchant to process transactions only for selected countries based on issuing country of the card. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check. Check applied for both Source and Destination card numbers.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Country identifier	comma separated country identifiers list	Type: String

Error codes

#	Code	Name
10015	1025	Issuer country not in trust list

Billing Country check

This referral list allows the merchant to process transactions only for selected countries based on billing country of the customer. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Country identifier	comma separated country identifiers list	Type: String

Error codes

#	Code	Name
10112	1122	Billing country not in trust list

IP address Country blacklist

This referral list allows the merchant to make block lists based on country of the customer IP address. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Country identifier	comma separated country identifiers list	Type: String

Error codes

#	Code	Name
10028	1038	Country in blacklist

Issuer Country blacklist

This referral list allows the merchant to make block lists based on issuing country of the card. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check. Check applied for both Source and Destination card numbers.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Country identifier	comma separated country identifiers list	Type: String

Error codes

#	Code	Name
10027	1037	Issuer country in blacklist

Billing Country blacklist

This referral list allows the merchant to make block lists based on billing country of the customer. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
Country identifier	comma separated country identifiers list	Type: String

Error codes

#	Code	Name
10113	1123	Billing country in blacklist

Issuer Country blacklist by Payment method

This referral list allows the merchant to make block lists based on issuing country of the card for selected payment method. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check. Check applied for both Source and Destination card numbers.

Score: Yes

Enabled by default: N

Error codes

#	Code	Name
10123	1133	Issuer country in blacklist for selected payment method

Transaction amount check

This check can be used to apply higher risk scores to transactions based on the transaction amount.

Score: Ind

Enabled by default: N

Error codes

#	Code	Name
10032	1042	Incorrect transaction amount

Source Credit Card type check

This referral list allows the merchant to process transactions only for selected Source Credit Card types. Counts Sale and Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
forbidden card level types	comma separated card level type list. Typical card types: PREPAID, REWARDS, VIRTUAL, CASH, ATM, STANDARD, CLASSIC, GOLD, SIGNATURE, PLATINUM, ELECTRON, CORPORATE, BUSINESS, WORLD, DEBIT and variations like GOLD REWARDS, WORLD CORPORATE, etc.	Type: String Default: PREPAID, REWARD

Error codes

#	Code	Name
10127	1137	Unsupported product type

Destination Credit Card type check

This referral list allows the merchant to process transactions only for selected Destination Credit Card types. Counts Sale and Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
forbidden card level types	comma separated card level type list. Typical card types: PREPAID, REWARDS, VIRTUAL, CASH, ATM, STANDARD, CLASSIC, GOLD, SIGNATURE, PLATINUM, ELECTRON, CORPORATE, BUSINESS, WORLD, DEBIT and variations like GOLD REWARDS, WORLD CORPORATE, etc.	Type: String Default: PREPAID, REWARD

Error codes

#	Code	Name
10129	1139	Unsupported destination product type

10.10.5 Velocity filters

Source Credit Card Number decline frequency for last 24 hours (daily decline limit)

This check fires when the number or amount of declined transactions associated with exact Source credit card number exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Account verification, Sale, Preauth or Transfer transactions in the Declined or Filtered status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
For all merchant projects	current total transactions amount or count for the last 24 hours for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
Amount limit	maximum total transactions amount for the last 24 hours for this credit card used as Source card	Type: Decimal Default: 99999999
Quantity limit	maximum total transactions count for the last 24 hours for this credit card used as Source card	Type: Int Default: 99999
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10083	1093	Daily decline amount limit exceeded for sender
10084	1094	Daily decline quantity limit exceeded for sender

Source Credit Card Number increasing sequence of approved transaction amounts for last 24 hours (daily rising limit)

This check fires when the number of approved transactions with increasing amount associated with exact Source credit card number exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 increasing transactions, it fires on the 11th increasing transaction in 24 hours. Counts Sale, Preauth or Transfer transactions in the Approved status. This check only cross-checks transactions within the same merchant account in the same project.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
Quantity limit	maximum transactions count with increasing amount for the last 24 hours for this credit card number	Type: Int Default: 99999

Error codes

#	Code	Name
10022	1032	Too many transactions with increasing amounts for sender

Source Credit Card Number decreasing sequence of declined transaction amounts for last 24 hours (daily lessening limit)

This check fires when the number of declined transactions with strictly decreasing amount associated with exact Source credit card number exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 strictly decreasing transactions, it fires on the 11th strictly decreasing transaction in 24 hours. Counts Account verification, Sale, Preauth or Transfer transactions in the Filtered and Declined status. This check only cross-checks transactions within the same merchant account in the same project.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
Quantity limit	maximum transactions count with decreasing amount for the last 24 hours for this credit card number	Type: Int Default: 99999

Error codes

#	Code	Name
10082	1092	Too many transactions with decreasing amounts for sender

Source Credit Card Number usage frequency for last 24 hours (daily limit)

This check fires when the number or amount of transactions associated with exact Source credit card number exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
For all merchant projects	current total transactions amount or count for the last 24 hours for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
Amount limit	maximum total transactions amount for the last 24 hours for this credit card used as Source card	Type: Decimal Default: 99999999
Quantity limit	maximum total transactions count for the last 24 hours for this credit card used as Source card	Type: Int Default: 99999
Skip payouts	allows to process Payout transactions even when the count or amount exceeds the thresholds	Type: String Default: Y

continues on next page

Table 71 – continued from previous page

Name	Description	Value
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	“Y” For calculation using calendar days instead of calculation from moment when filter was enabled “N” for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10016	1026	Daily amount limit exceeded for sender
10017	1027	Daily quantity limit exceeded for sender

Source Credit Card Number usage frequency for last 7 days (weekly limit)

This check fires when the number or amount of transactions associated with exact Source credit card number exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
For all merchant projects	current total transactions amount or count for the last 7 days for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
Amount limit	maximum total transactions amount for the last 7 days for this credit card used as Source card	Type: Decimal Default: 99999999
Quantity limit	maximum total transactions count for the last 7 days for this credit card used as Source card	Type: Int Default: 99999

continues on next page

Table 73 - continued from previous page

Name	Description	Value
Skip payouts	allows to process Payout transactions even when the count or amount exceeds the thresholds	Type: String Default: Y
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10018	1028	Weekly amount limit exceeded for sender
10019	1029	Weekly quantity limit exceeded for sender

Source Credit Card Number usage frequency for last month (monthly limit)

This check fires when the number or amount of transactions associated with exact Source credit card number exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
For all merchant projects	current total transactions amount or count for the last one month for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y

continues on next page

Table 75 - continued from previous page

Name	Description	Value
Amount limit	maximum total transactions amount for the last one month for this credit card used as Source card	Type: Decimal Default: 99999999
Quantity limit	maximum total transactions count for the last one month for this credit card used as Source card	Type: Int Default: 99999
Skip payouts	allows to process Payout transactions even when the count or amount exceeds the thresholds	Type: String Default: Y
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10020	1030	Monthly amount limit exceeded for sender
10021	1031	Monthly quantity limit exceeded for sender

Destination Credit Card Number usage frequency for last 24 hours (daily limit)

This check fires when the number or amount of transactions associated with exact Destination credit card number exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 24 hours for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 24 hours for this credit card used as Destination card	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 24 hours for this credit card used as Destination card	Type: Int Default: 99999
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10049	1059	Daily amount limit exceeded for recipient
10050	1060	Daily quantity limit exceeded for recipient

Destination Credit Card Number usage frequency for last 7 days (weekly limit)

This check fires when the number or amount of transactions associated with exact Destination credit card number exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 7 days for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 7 days for this credit card used as Destination card	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 7 days for this credit card used as Destination card	Type: Int Default: 99999
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10051	1061	Weekly amount limit exceeded for recipient
10052	1062	Weekly quantity limit exceeded for recipient

Destination Credit Card Number usage frequency for last month (monthly limit)

This check fires when the number or amount of transactions associated with exact Destination credit card number exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last one month for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last one month for this credit card used as Destination card	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last one month for this credit card used as Destination card	Type: Int Default: 99999
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10053	1063	Monthly amount limit exceeded for recipient
10054	1064	Monthly quantity limit exceeded for recipient

Total Credit Card Number usage frequency for last 24 hours (daily limit)

This check fires when the number or amount of transactions associated with exact credit card number used as Source or Destination exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Sale, Preauth or Transfer transactions in the approved status

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 24 hours for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 24 hours for this credit card used as Source or Destination	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 24 hours for this credit card used as Source or Destination	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10055	1065	Daily total amount limit exceeded for sender
10056	1066	Daily total quantity limit exceeded for sender
10057	1067	Daily total amount limit exceeded for recipient
10058	1068	Daily total quantity limit exceeded for recipient

Total Credit Card Number usage frequency for last 7 days (weekly limit)

This check fires when the number or amount of transactions associated with exact credit card number used as Source or Destination exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Sale, Preauth or Transfer transactions in the approved status

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 7 days for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 7 days for this credit card used as Source or Destination	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 7 days for this credit card used as Source or Destination	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10059	1069	Weekly total amount limit exceeded for sender
10060	1070	Weekly total quantity limit exceeded for sender
10061	1071	Weekly total amount limit exceeded for recipient
10062	1072	Weekly total quantity limit exceeded for recipient

Total Credit Card Number usage frequency for last month (monthly limit)

This check fires when the number or amount of transactions associated with exact credit card number used as Source or Destination exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last one month for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last one month for this credit card used as Source or Destination	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last one month for this credit card used as Source or Destination	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10063	1073	Monthly total amount limit exceeded for sender
10064	1074	Monthly total quantity limit exceeded for sender
10065	1075	Monthly total amount limit exceeded for recipient
10066	1076	Monthly total quantity limit exceeded for recipient

Purpose usage frequency for last 24 hours (daily limit)

This check fires when the number or amount of transactions associated with exact Purpose exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 24 hours for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 24 hours for this Purpose	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 24 hours for this Purpose	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10040	1050	Daily amount limit exceeded for purpose
10041	1051	Daily quantity limit exceeded for purpose

Purpose usage frequency for last 7 days (weekly limit)

This check fires when the number or amount of transactions associated with exact Purpose exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 7 days for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 7 days for this Purpose	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 7 days for this Purpose	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10042	1052	Weekly amount limit exceeded for purpose
10043	1053	Weekly quantity limit exceeded for purpose

Purpose usage frequency for for last month (monthly limit)

This check fires when the number or amount of transactions associated with exact Purpose exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last one month for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last one month for this Purpose	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last one month for this Purpose	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10044	1054	Monthly amount limit exceeded for purpose
10045	1055	Monthly quantity limit exceeded for purpose

Email usage frequency for last 24 hours (daily limit)

This check fires when the number or amount of transactions associated with exact Email address exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 24 hours for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 24 hours for this Email address	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 24 hours for this Email address	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10073	1083	Daily amount limit exceeded for email address
10074	1084	Daily quantity limit exceeded for email address

Email usage frequency for last 7 days (weekly limit)

This check fires when the number or amount of transactions associated with exact Email address exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 7 days for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 7 days for this Email address	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 7 days for this Email address	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10075	1085	Weekly amount limit exceeded for email address
10076	1086	Weekly quantity limit exceeded for email address

Email usage frequency for last month (monthly limit)

This check fires when the number or amount of transactions associated with exact Email address exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last one month for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last one month for this Email address	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last one month for this Email address	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10077	1087	Monthly amount limit exceeded for email address
10078	1088	Monthly quantity limit exceeded for email address

IP address usage frequency for last 24 hours (daily limit)

This check fires when the number or amount of transactions associated with exact IP address exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 24 hours for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 24 hours for this IP address	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 24 hours for this IP address	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10100	1110	Daily amount limit exceeded for IP address
10101	1111	Daily quantity limit exceeded for IP address

IP address usage frequency for last 7 days (weekly limit)

This check fires when the number or amount of transactions associated with exact IP address exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 7 days for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last 7 days for this IP address	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 7 days for this IP address	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10102	1112	Weekly amount limit exceeded for IP address
10103	1113	Weekly quantity limit exceeded for IP address

IP address usage frequency for last month (monthly limit)

This check fires when the number or amount of transactions associated with exact IP address exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last one month for this credit card value would be calculated: - Y: for all projects - 3DS: for 3DS gates only - Non-3DS: for non-3DS gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
amount limit	maximum total transactions amount for the last one month for this IP address	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last one month for this IP address	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10104	1114	Monthly amount limit exceeded for IP address
10105	1115	Monthly quantity limit exceeded for IP address

Source Credit Card Number usage frequency for Purpose

This check fires when the number of Source Credit Cards associated with exact Purpose exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards in 6 hours, it fires on the 11th unique Credit Card in 6 hours. Counts unique Source Credit Card numbers for Account verification, Sale, Preauth or Transfer transactions in any status for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique credit card numbers count	Type: Int Default: 12
maximum card number count	maximum number of unique credit cards	Type: Int Default: 5

Error codes

#	Code	Name
10071	1081	Too many credit cards used for the same account

Source Credit Card Number usage frequency for Email address

This check fires when the number of Source Credit Cards associated with exact Email address exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards in 6 hours, it fires on the 11th unique Credit Card in 6 hours. Counts unique Source Credit Card numbers for Sale or Preauth transactions in any status for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique credit card numbers count	Type: Int Default: 12

continues on next page

Table 109 – continued from previous page

Name	Description	Value
maximum card number count	maximum number of unique credit cards	Type: Int Default: 5

Error codes

#	Code	Name
10091	1101	Too many credit cards used for the same Email address

Source Credit Card Number usage frequency for First and Last name

This check fires when the number of Source Credit Cards associated with exact Customer First and Last names exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards in 6 hours, it fires on the 11th unique Credit Card in 6 hours. Counts unique Source Credit Card numbers for Sale or Preauth transactions in any status for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique credit card numbers count	Type: Int Default: 12
maximum card number count	maximum number of unique credit cards	Type: Int Default: 5

Error codes

#	Code	Name
10092	1102	Too many credit cards used for the same customer

Source Credit Card Number usage frequency for Destination Credit Card number

This check fires when the number of Source Credit Cards associated with exact Destination Credit Card number exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards in 6 hours, it fires on the 11th unique Credit Card in 6 hours. Counts unique Source Credit Card numbers for Sale or Preauth transactions in any status for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique credit card numbers count	Type: Decimal Default: 12.00
maximum card number count	maximum number of unique credit cards	Type: Int Default: 5

Error codes

#	Code	Name
10093	1103	Too many credit cards used for the same destination card

Source Credit Card Number usage frequency for Email or IP address

This check fires when the number of Source Credit Cards associated with exact Email or IP address exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards in 6 hours, it fires on the 11th unique Credit Card in 6 hours. Counts unique Source Credit Card numbers for Sale, Preauth or Transfer transactions in Approved status for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique credit card numbers count	Type: Int Default: 24
maximum card number count	maximum number of unique credit cards	Type: Int Default: 3

Error codes

#	Code	Name
10026	1036	Too many credit cards used for the same email or IP address

Source Credit Card Number usage frequency

This check fires when the number of requests associated with exact Source Credit Card exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 requests in 6 hours, it fires on the 11th request in 6 hours. Counts requests count for Account verification, Sale, Preauth or Transfer transactions in the Approved or Declined status for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate requests count	Type: Int Default: 24
maximum number of requests	maximum number of requests allowed	Type: Int Default: 5

Error codes

#	Code	Name
10072	1082	Too many requests for the same credit card

Customer IP address usage frequency

This check fires when the number of requests associated with exact Customer IP address exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 requests in 60 minutes, it fires on the 11th request in 60 minutes. Counts requests count for Account verification, Sale, Preauth or Transfer transactions in the Approved or Declined status for the current Merchant. Local IP addresses are ignored. Requests from IP address listed in “Merchant API IP address” list are ignored, i.e. if merchant initiates the request from IP address X.X.X.X and knowingly sets customer_ip_address to Y.Y.Y.Y for each transaction, but address X.X.X.X classified as “API IP address” for this merchant, this check will be ignored.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in minutes	time frame to calculate requests count	Type: Int Default: 10
maximum number of requests	maximum number of requests allowed	Type: Int Default: 5

Error codes

#	Code	Name
10115	1125	Too many requests for the same IP address

Credit Card number already used from another IP address

This check fires when the Credit Card number has already been successfully used from a different IP address. The time threshold is a moving window calculated backwards from the moment of the transaction. This check only cross-checks transactions within the same merchant account. Analyses Sale, Preauth or Transfer transactions in the Approved status. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in minutes	time frame to analyse approved transactions in minutes, set to 0 for infinity (not recommended)	Type: Int Default: 30

Error codes

#	Code	Name
10008	1006	Too many IP addresses for the same credit card number

Credit Card number already used from another Country

This check fires when the Credit Card number has already been successfully used from another Country. IP address is used to calculate customers country code. This check only cross-checks transactions within the same merchant account. Analyses Sale, Preauth or Transfer transactions in the Approved status. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check.

Score: Yes

Enabled by default: N

Error codes

#	Code	Name
10079	1089	Too many countries for the same credit card number

Credit Card number already used with another Email

This check fires when the Credit Card number has already been successfully used with a different Email address. The time threshold is a moving window calculated backwards from the moment of the transaction. This check only cross-checks transactions within the same merchant account. Analyses Sale, Preauth or Transfer transactions in the Approved status. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in minutes	time frame to analyse approved transactions in minutes, set to 0 for infinity (not recommended)	Type: Int Default: 30

Error codes

#	Code	Name
10007	1005	Too many Emails for the same credit card number

Credit Card number already used with another Purpose

This check fires when the Credit Card number has already been successfully used with a different Purpose. The time threshold is a moving window calculated backwards from the moment of the transaction. This check only cross-checks transactions within the same merchant account. Analyses Sale, Preauth or Transfer transactions in the Approved status. Requests from IP addresses listed in "Merchant API IP address" are ignoring this check.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in minutes	time frame to analyse approved transactions in minutes, set to 0 for infinity (not recommended)	Type: Int Default: 30

Error codes

#	Code	Name
10081	1091	Too many purposes for the same credit card number

Credit Card number already used with another Cardholder name

This check fires when the Credit Card number has already been successfully used with another Cardholder name. Distances between current Cardholder name and existing ones are calculated using Levenshtein algorithm. This check only cross-checks transactions within the same merchant account. Analyses Sale, Preauth or Transfer transactions in the Approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to analyse approved transactions in hours	Type: Int Default: 30

continues on next page

Table 128 – continued from previous page

Name	Description	Value
greatest levenshtein distance	greatest levenshtein distance to consider both cardholder names equal	Type: Int Default: 3

Error codes

#	Code	Name
10089	1099	Too many cardholder names for the same credit card number

Customer IP address already used by another Cardholder

This check fires when the Customer IP address has already been successfully used with a different Cardholder name. The time threshold is a moving window calculated backwards from the moment of the transaction. This check only cross-checks transactions within the same merchant account. Analyses Sale, Preauth or Transfer transactions in the Approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in minutes	time frame to analyse approved transactions in minutes	Type: Int Default: 30

Error codes

#	Code	Name
10006	1004	Too many card holders from the same IP address

Customer Email address already used by another Cardholder

This check fires when the Customer Email has already been successfully used with a different Cardholder name. The time threshold is a moving window calculated backwards from the moment of the transaction. This check only cross-checks transactions within the same merchant account. Analyses Sale, Preauth or Transfer transactions in the Approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in minutes	time frame to analyse approved transactions in minutes	Type: Int Default: 30

Error codes

#	Code	Name
10005	1003	Too many card holders for the same Email

Source Credit Card Number approved transaction interval

This check fires when the interval for the last approved transaction associated with exact Source credit card number lesser the configured thresholds. The time threshold is time window calculated backwards from the moment of the transaction. The risk fires on the transaction below the set threshold. So, you set a threshold of 10 minutes and the last approved transaction time is 10:00:00, it fires until 10:10:01. Counts Sale, Preauth or Transfer transactions in the Approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in minutes	time frame to analyse approved transactions in minutes	Type: Int Default: 30
for all merchant projects	Y - to check transactions for all projects of the current merchant, otherwise check transactions for current project only	Type: String Default: N

Error codes

#	Code	Name
10023	1033	Too many approved transactions for the same credit card number

Source Credit Card Number declined transaction interval

This check fires when the interval for the last declined transaction associated with exact Source credit card number lesser the configured thresholds. The time threshold is time window calculated backwards from the moment of the transaction. The risk fires on the transaction below the set threshold. So, you set a threshold of 10 minutes and the last approved transaction time is 10:00:00, it fires until 10:10:01. Counts Sale, Preauth or Transfer transactions in the Declined or Filtered status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in minutes	time frame to analyse approved transactions in minutes	Type: Int Default: 30
for all merchant projects	Y - to check transactions for all projects of the current merchant, otherwise check transactions for current project only	Type: String Default: N

Error codes

#	Code	Name
10085	1095	Too many declined transactions for the same credit card number

Source Credit Card Number Issuer Country change frequency for current Purpose

This check fires when the number of Countries, calculated for Source Credit Card number issuer, associated with exact Purpose exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 2 Countries in 24 hours, it fires on the 3rd unique Country in 24 hours for the same Purpose. Counts unique Source Credit Card number issuer Countries for Sale, Preauth or Transfer transactions in Approved status for the current Merchant. Requests from IP addresses listed in "Merchant API IP address" are ignoring this check.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate countries count	Type: Int Default: 24
maximum countries count	maximum number of countries per one purpose allowed	Type: Int Default: 5
ignore BINs	requests for listed BINs are ignoring this check	Type: String

Error codes

#	Code	Name
10122	1132	Too many countries per account

Reversal frequency

This check fires when the number or ratio of reversal transactions calculated for the whole merchant or for the exact merchant project exceeds the configured thresholds. Ratio is calculated for the full lifetime, absolute number could be limited for the whole lifetime or on daily basis from 00:00:00 till 23:59:59. The risk fires on the transaction after the set threshold. So, if you set a threshold of 5 reversals, it fires on the 6th one. Ratio calculation is based on transactions count, i.e. total reversal transactions count divided by total sale transactions count. Counts Sale and Capture transactions in Approved status and Reversal and Void transactions.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
percentage ratio	maximum allowable ratio of reversal transactions calculated by transaction count in percent (from 0 to 100)	Type: Decimal Default: 101.00
absolute number	maximum allowable number of reversal transactions	Type: Int Default: 99999
daily absolute number	maximum allowable number of reversal transactions per day	Type: Int Default: 9999
for all merchant projects	Y - to check transactions for all projects of the current merchant, otherwise check transactions for current project only	Type: String Default: N

Error codes

#	Code	Name
10036	1046	Too high reversal ratio
10037	1047	Too many reversals
10038	1048	Too many reversals today

Fingerprint usage frequency for last 24 hours (daily limit)

This check fires when the number or amount of transactions associated with exact Fingerprint exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last 24 hours for this Fingerprint	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 24 hours for this Fingerprint	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: N
for all merchant projects	Y - to check transactions for all projects of the current merchant, otherwise check transactions for current project only	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10170	1180	Daily fingerprint amount limit exceeded for fingerprint
10171	1181	Daily fingerprint quantity limit exceeded for fingerprint

Fingerprint usage frequency for last 7 days (weekly limit)

This check fires when the number or amount of transactions associated with exact Fingerprint exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last 7 days for this Fingerprint	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last 7 days for this credit Fingerprint	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: N
for all merchant projects	Y - to check transactions for all projects of the current merchant, otherwise check transactions for current project only	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10172	1182	Weekly fingerprint amount limit exceeded for fingerprint
10173	1183	Weekly fingerprint quantity limit exceeded for fingerprint

Fingerprint usage frequency for last month (monthly limit)

This check fires when the number or amount of transactions associated with exact Fingerprint exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last one month for this Fingerprint	Type: Decimal Default: 99999999
quantity limit	maximum total transactions count for the last one month for this credit Fingerprint	Type: Int Default: 99999
Subtract Cancel transactions	subtracts Cancelled transactions from the calculated count and amount thresholds	Type: String Default: N
for all merchant projects	Y - to check transactions for all projects of the current merchant, otherwise check transactions for current project only	Type: String Default: Y
Use calendar days	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10174	1184	Monthly fingerprint amount limit exceeded for fingerprint
10175	1185	Monthly fingerprint quantity limit exceeded for fingerprint

Source Credit Card Number usage frequency for Fingerprint

This check fires when the number of Source Credit Cards associated with exact Fingerprint exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards in 6 hours, it fires on the 11th unique Credit Card in 6 hours. Counts unique Source Credit Card numbers for Account verification, Sale, Preauth or Transfer transactions in any status for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique card number count	Type: Int Default: 12
maximum card count	maximum number of unique card numbers	Type: Int Default: 5

Error codes

#	Code	Name
10176	1186	Too many source credit cards used for the same fingerprint

Source Credit Card number Issuer Country change frequency for current Device Fingerprint

This check fires when the number of Countries, calculated for Source Credit Card number issuer, associated with exact Fingerprint exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 2 Countries in 24 hours, it fires on the 3rd unique Country in 24 hours for the same Purpose. Counts unique Source Credit Card number issuer Countries for Sale, Preauth or Transfer transactions in Approved status for the current Merchant. Requests from IP addresses listed in "Merchant API IP address" are ignoring this check.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique card number count	Type: Int Default: 12
maximum countries count	maximum number of countries for one fingerprint	Type: Int Default: 5

Error codes

#	Code	Name
10177	1187	Too many countries for the same fingerprint

Destination Credit Card Number usage frequency for Device fingerprint

This check fires when the number of Destination Credit Cards associated with exact Device Fingerprint exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards in 6 hours, it fires on the 11th unique Credit Card in 6 hours. Counts unique Destination Credit Card numbers for Transfer transactions in any status for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique destination credit card numbers count	Type: Int Default: 12
maximum card count	maximum number of unique destination credit cards	Type: Int Default: 5

Error codes

#	Code	Name
10178	1188	Too many destination credit cards used for the same fingerprint

Destination Credit Card number Issuer Country change frequency for current Device fingerprint

This check fires when the number of Countries, calculated for Destination Credit Card number issuer, associated with exact Device Fingerprint exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 2 Countries in 24 hours, it fires on the 3rd unique Country in 24 hours for the same Fingerprint. Counts unique Destination Credit Card number issuer Countries for Sale, Preauth or Transfer transactions in Approved status for the current Merchant. Requests from IP addresses listed in "Merchant API IP address" are ignoring this check.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique card number count	Type: Int Default: 12
maximum countries count	maximum number of destination countries for one fingerprint	Type: Int Default: 5

Error codes

#	Code	Name
10179	1189	Too many destination credit cards countries used for the same fingerprint

Email usage frequency for Device fingerprint

This check fires when the number of Purpose associated with exact Device Fingerprint exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 2 Purpose in 24 hours, it fires on the 3rd unique Purpose in 24 hours for the same Fingerprint. Counts unique Purposes for all types of transactions for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique emails count	Type: Int Default: 24
maximum emails count	maximum number of emails for one fingerprint	Type: Int Default: 5

Error codes

#	Code	Name
10180	1190	Too many emails used for the same fingerprint

Purpose usage frequency for Device fingerprint

This check fires when the number of Purpose associated with exact Device Fingerprint exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 2 Purpose in 24 hours, it fires on the 3rd unique Purpose in 24 hours for the same Fingerprint. Counts unique Purposes for all types of transactions for the current Merchant.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
checking interval in hours	time frame to calculate unique purposes count	Type: Int Default: 24
maximum purposes count	maximum number of purposes for one fingerprint	Type: Int Default: 5

Error codes

#	Code	Name
10181	1191	Too many purposes used for the same fingerprint

Account Number usage frequency for last 24 hours (daily limit)

This check fires when the number or amount of transactions associated with exact Account number exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last 24 hours for this account number	Type: Decimal Default: 99999999
for all merchant projects	current total transactions amount or count for the last 24 hours for this credit card value would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
quantity limit	maximum total transactions count for the last 24 hours for this account number	Type: Int Default: 99999999
skip payouts	ignore payouts	Type: String Default: N
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N
use calendar day	"Y" For calculation using calendar days instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10187	1197	Daily amount limit exceeded for account number address
10188	1198	Daily quantity limit exceeded for account number address

Account Number usage frequency for last 7 days (weekly limit)

This check fires when the number or amount of transactions associated with exact Account number exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last 7 days for this account number	Type: Decimal Default: 99999999
calendar week starts from Sunday	- Y: calendar week starts from Sunday - - N: calendar week starts from Monday	Type: String Default: N
for all merchant projects	current total transactions amount or count for the last 7 days for this credit card value would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
quantity limit	maximum total transactions count for the last 7 days for this account number calculated	Type: Int Default: 99999999
skip payouts	ignore payouts	Type: String Default: N
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N

continues on next page

Table 162 – continued from previous page

Name	Description	Value
use calendar week	“Y” For calculation using calendar week instead of calculation from moment when filter was enabled “N” for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10189	1199	Weekly amount limit exceeded for account number address
10190	1200	Weekly quantity limit exceeded for account number address

Account Number usage frequency for last month (monthly limit)

This check fires when the number or amount of transactions associated with exact Account number exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last one month for this account number	Type: Decimal Default: 99999999
for all merchant projects	current total transactions amount or count for the last one month for this credit card value would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount or quantity limit values	Type: String Default: Y
quantity limit	maximum total transactions count for the last one month for this account number	Type: Int Default: 99999999

continues on next page

Table 164 – continued from previous page

Name	Description	Value
skip payouts	ignore payouts	Type: String Default: N
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N
use calendar month	“Y” For calculation using calendar month instead of calculation from moment when filter was enabled “N” for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10191	1201	Monthly amount limit exceeded for account number address
10192	1202	Monthly quantity limit exceeded for account number address

Source Credit Card Number decline frequency for last week (weekly decline limit)

This check fires when the number or amount of declined transactions associated with exact Source credit card number exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Account verification, Sale, Preauth or Transfer transactions in the Declined status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last 7 days for this credit card used as Source card	Type: Decimal Default: 999999999
calendar week starts from Sunday	Y: calendar week starts from Sunday; N: calendar week starts from Monday	Type: String Default: N

continues on next page

Table 166 – continued from previous page

Name	Description	Value
for all merchant projects	current total transactions amount or count for the last 7 days for this credit card would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
quantity limit	maximum total transactions count for the last 7 days for this credit card used as Source card	Type: Int Default: 99999
use calendar week	“Y” For calculation using calendar week instead of calculation from moment when filter was enabled “N” for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10206	1216	Weekly decline amount limit exceeded for sender
10207	1217	Weekly decline quantity limit exceeded for sender

Source Credit Card Number usage frequency per Email address for last 24 hours (daily limit)

This check fires when the number of Source Credit Cards associated with exact Email address exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards, it fires on the 11th unique Credit Card in 24 hours. Counts unique Source Credit Card numbers for Sale or Preauth transactions in any status for the current Merchant.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
approved only	counts unique Source Credit Card numbers for Sale or Preauth transactions in approved status	Type: String Default: N

continues on next page

Table 168 – continued from previous page

Name	Description	Value
maximum card number count	maximum total card number count for the last 24 hours for this Email address	Type: Int Default: 99999
use calendar day	“Y” For calculation using calendar days instead of calculation from moment when filter was enabled “N” for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10208	1218	Daily card number count limit exceeded for email address

Source Credit Card Number usage frequency per Email address for last 7 days (weekly limit)

This check fires when the number of Source Credit Cards associated with exact Email address exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards, it fires on the 11th unique Credit Card in 7 days. Counts unique Source Credit Card numbers for Sale or Preauth transactions in any status for the current Merchant.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
approved only	counts unique Source Credit Card numbers for Sale or Preauth transactions in approved status	Type: String Default: N
calendar week starts from Sunday	Y: calendar week starts from Sunday - N: calendar week starts from Monday	Type: String Default: N
maximum card number count	maximum total card number count for the last 7 days for this Email address	Type: Int Default: 99999
use calendar week	“Y” For calculation using calendar week instead of calculation from moment when filter was enabled “N” for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10209	1219	Weekly card number count limit exceeded for email address

Source Credit Card Number usage frequency per Email address for last month (monthly limit)

This check fires when the number of Source Credit Cards associated with exact Email address exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 Credit Cards, it fires on the 11th unique Credit Card in 1 month. Counts unique Source Credit Card numbers for Sale or Preauth transactions in any status for the current Merchant.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
approved only	counts unique Source Credit Card numbers for Sale or Preauth transactions in approved status	Type: String Default: N
maximum card number count	maximum total card number count for the last month for this Email address	Type: Int Default: 99999
use calendar month	"Y" For calculation using calendar month instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10210	1220	Monthly card number count limit exceeded for email address

Source Credit Card Number usage frequency for last N days

This check fires when the number or amount of transactions associated with exact Source credit card number exceeds the configured thresholds. The time threshold is a N days window calculated backwards from the moment of the transaction. The N parameter (date period) can be set from 1 to 30. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in N days. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last N days for this credit card used as Source card	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last N days for this credit card value would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
for last N days	date period can be set from 1 to 30 days	Type: Int Default: 1
quantity limit	maximum total transactions count for the last N days for this credit card used as Source card	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N

Error codes

#	Code	Name
10211	1221	Specified period amount limit exceeded for sender
10212	1222	Specified period quantity limit exceeded for sender

Destination Credit Card Number usage frequency for last N days

This check fires when the number or amount of transactions associated with exact Destination credit card number exceeds the configured thresholds. The time threshold is a N days window calculated backwards from the moment of the transaction. The N parameter (date period) can be set from 1 to 30. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in N days. Counts Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last N days for this credit card used as Destination card	Type: Decimal Default: 9999999999
for all merchant projects	current total transactions amount or count for the last N days for this Destination card value would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
for last N days	date period can be set from 1 to 30 days	Type: Int Default: 1
quantity limit	maximum total transactions count for the last N days for this credit card used as Destination card	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N

Error codes

#	Code	Name
10213	1223	Specified period amount limit exceeded for recipient
10214	1224	Specified period quantity limit exceeded for recipient

Total Credit Card Number usage frequency for last N days

This check fires when the number or amount of transactions associated with exact credit card number used as Source or Destination exceeds the configured thresholds. The time threshold is a N days window calculated backwards from the moment of the transaction. The N parameter (date period) can be set from 1 to 30. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in N days. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last N days for this credit card used as Source or Destination card	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last N days for this credit card value would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
for last N days	date period can be set from 1 to 30 days	Type: Int Default: 1
quantity limit	maximum total transactions count for the last N days for this credit card used as Source or Destination card	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N

Error codes

#	Code	Name
10215	1225	Specified period total amount limit exceeded for sender
10216	1226	Specified period total quantity limit exceeded for sender
10217	1227	Specified period total amount limit exceeded for recipient
10218	1228	Specified period total quantity limit exceeded for recipient

Purpose usage frequency for last N days

This check fires when the number or amount of transactions associated with exact Purpose exceeds the configured thresholds. The time threshold is a N days window calculated backwards from the moment of the transaction. The N parameter (date period) can be set from 1 to 30. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in N days. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last N days for this Purpose	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last N days for this Purpose would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
for last N days	date period can be set from 1 to 30 days	Type: Int Default: 1
quantity limit	maximum total transactions count for the last N days for this Purpose	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N

Error codes

#	Code	Name
10219	1229	Specified period amount limit exceeded for purpose
10220	1230	Specified period quantity limit exceeded for purpose

Email usage frequency for last N days

This check fires when the number or amount of transactions associated with exact Email exceeds the configured thresholds. The time threshold is a N days window calculated backwards from the moment of the transaction. The N parameter (date period) can be set from 1 to 30. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in N days. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last N days for this Email	Type: Decimal Default: 9999999999
for all merchant projects	current total transactions amount or count for the last N days for this Email would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
for last N days	date period can be set from 1 to 30 days	Type: Int Default: 1
quantity limit	maximum total transactions count for the last N days for this Email	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N

Error codes

#	Code	Name
10221	1231	Specified period amount limit exceeded for email address
10222	1232	Specified period quantity limit exceeded for email address

IP address usage frequency for last N days

This check fires when the number or amount of transactions associated with exact IP address exceeds the configured thresholds. The time threshold is a N days window calculated backwards from the moment of the transaction. The N parameter (date period) can be set from 1 to 30. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in N days. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last N days for this IP address	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last N days for this IP address would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
for last N days	date period can be set from 1 to 30 days	Type: Int Default: 1
quantity limit	maximum total transactions count for the last N days for this IP address	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N

Error codes

#	Code	Name
10223	1233	Specified period amount limit exceeded for IP address
10224	1234	Specified period quantity limit exceeded for IP address

Fingerprint usage frequency for last N days

This check fires when the number or amount of transactions associated with exact Fingerprint exceeds the configured thresholds. The time threshold is a N days window calculated backwards from the moment of the transaction. The N parameter (date period) can be set from 1 to 30. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in N days. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last N days for this Fingerprint	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last N days for this Fingerprint would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
for last N days	date period can be set from 1 to 30 days	Type: Int Default: 1
quantity limit	maximum total transactions count for the last N days for this Fingerprint	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N

Error codes

#	Code	Name
10225	1235	Specified period amount limit exceeded for fingerprint
10226	1236	Specified period quantity limit exceeded for fingerprint

Account Number usage frequency for last N days

This check fires when the number or amount of transactions associated with exact Account Number exceeds the configured thresholds. The time threshold is a N days window calculated backwards from the moment of the transaction. The N parameter (date period) can be set from 1 to 30. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in N days. Counts Sale, Preauth or Transfer transactions in the approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last N days for this Account Number	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last N days for this Account Number would be calculated Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
for last N days	date period can be set from 1 to 30 days	Type: Int Default: 1
quantity limit	maximum total transactions count for the last N days for this Account Number	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: N

Error codes

#	Code	Name
10227	1237	Specified period amount limit exceeded for account number
10228	1238	Specified period quantity limit exceeded for account number

Source Credit Card Number decline frequency for last month (monthly decline limit)

This check fires when the number or amount of declined transactions associated with exact Source credit card number exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Account verification, Sale, Preauth or Transfer transactions in the Declined status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last month for this credit card used as Source card	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last month for this credit card would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
quantity limit	maximum total transactions count for the last month for this credit card used as Source card	Type: Int Default: 99999
use calendar month	"Y" For calculation using calendar month instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10229	1239	Monthly decline amount limit exceeded for sender
10230	1240	Monthly decline quantity limit exceeded for sender

Destination Credit Card Number decline frequency for last 24 hours (daily decline limit)

This check fires when the number or amount of declined transactions associated with exact Destination credit card number exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Transfer transactions in the Declined status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last 24 hours for this credit card used as Destination card	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last day for this credit card would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
quantity limit	maximum total transactions count for the last day for this credit card used as Destination card	Type: Int Default: 99999
use calendar day	"Y" For calculation using calendar day instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10231	1241	Daily decline amount limit exceeded for recipient
10232	1242	Daily decline quantity limit exceeded for recipient

Destination Credit Card Number decline frequency for last week (weekly decline limit)

This check fires when the number or amount of declined transactions associated with exact Destination credit card number exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Transfer transactions in the Declined status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last week for this credit card used as Destination card	Type: Decimal Default: 999999999
calendar week starts from Sunday	Y: calendar week starts from Sunday; N: calendar week starts from Monday	Type: String Default: N
for all merchant projects	current total transactions amount or count for the last week for this credit card would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
quantity limit	maximum total transactions count for the last week for this credit card used as Destination card	Type: Int Default: 99999
use calendar week	"Y" For calculation using calendar week instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10233	1243	Weekly decline amount limit exceeded for recipient
10234	1244	Weekly decline quantity limit exceeded for recipient

Destination Credit Card Number decline frequency for last month (monthly decline limit)

This check fires when the number or amount of declined transactions associated with exact Destination credit card number exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Transfer transactions in the Declined status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last month for this credit card used as Destination card	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last month for this credit card would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
quantity limit	maximum total transactions count for the last month for this credit card used as Destination card	Type: Int Default: 99999
use calendar month	"Y" For calculation using calendar month instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10237	1247	Daily decline total amount limit exceeded for sender
10238	1248	Daily decline total quantity limit exceeded for sender
10239	1249	Daily decline total amount limit exceeded for recipient
10240	1250	Daily decline total quantity limit exceeded for recipient

Total Credit Card Number decline frequency for last 24 hours (daily decline limit)

This check fires when the number or amount of declined transactions associated with exact Source or Destination credit card number exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Account verification, Sale, Preauth or Transfer transactions in the Declined status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last 24 hours for this credit card	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last day for this credit card would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
quantity limit	maximum total transactions count for the last day for this credit card	Type: Int Default: 99999
use calendar day	"Y" For calculation using calendar day instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10241	1251	Weekly decline total amount limit exceeded for sender
10242	1252	Weekly decline total quantity limit exceeded for sender
10243	1253	Weekly decline total amount limit exceeded for recipient
10244	1254	Weekly decline total quantity limit exceeded for recipient

Total Credit Card Number decline frequency for last week (weekly decline limit)

This check fires when the number or amount of declined transactions associated with exact Source or Destination credit card number exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Account verification, Sale, Preauth or Transfer transactions in the Declined status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last week for this credit card	Type: Decimal Default: 999999999
calendar week starts from Sunday	Y: calendar week starts from Sunday; N: calendar week starts from Monday	Type: String Default: N
for all merchant projects	current total transactions amount or count for the last week for this credit card would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
quantity limit	maximum total transactions count for the last week for this credit card	Type: Int Default: 99999
use calendar week	"Y" For calculation using calendar week instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10241	1251	Weekly decline total amount limit exceeded for sender
10242	1252	Weekly decline total quantity limit exceeded for sender
10243	1253	Weekly decline total amount limit exceeded for recipient
10244	1254	Weekly decline total quantity limit exceeded for recipient

Total Credit Card Number decline frequency for last month (monthly decline limit)

This check fires when the number or amount of declined transactions associated with exact Source or Destination credit card number exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Account verification, Sale, Preauth or Transfer transactions in the Declined status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last month for this credit card	Type: Decimal Default: 999999999
for all merchant projects	current total transactions amount or count for the last month for this credit card would be calculated - Y: for all projects - 3D: for 3D gates only - Non3D: for non 3D gates only - N: for current project only of the current merchant and converted to current project currency to compare with amount limit values	Type: String Default: N
quantity limit	maximum total transactions count for the last month for this credit card	Type: Int Default: 99999
use calendar month	"Y" For calculation using calendar month instead of calculation from moment when filter was enabled "N" for calculation from moment when filter was enabled	Type: String Default: N

Error codes

#	Code	Name
10245	1255	Monthly decline total amount limit exceeded for sender
10246	1256	Monthly decline total quantity limit exceeded for sender
10247	1257	Monthly decline total amount limit exceeded for recipient
10248	1258	Monthly decline total quantity limit exceeded for recipient

Customer IP address anonymous VPN

This check fires when customer IP address is considered as anonymous VPN by MaxMind service. Counts Sale, Preauth or Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
filter parameter value	Y - to filter if condition is true, N - to filter if condition is false	Type: String Default: Y

Error codes

#	Code	Name
10250	1260	Customer IP address is anonymous VPN

Customer IP address anonymous

This check fires when customer IP address is considered as anonymous by MaxMind service. Counts Sale, Preauth or Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
filter parameter value	Y - to filter if condition is true, N - to filter if condition is false	Type: String Default: Y

Error codes

#	Code	Name
10251	1261	Customer IP address is anonymous

Customer IP Hosting Provider

This check fires when customer IP address belongs to a hosting or VPN provider considered by MaxMind service. Counts Sale, Preauth or Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
filter parameter value	Y - to filter if condition is true, N - to filter if condition is false	Type: String Default: Y

Error codes

#	Code	Name
10252	1262	Customer ip address is hosting provider

Customer IP Public Proxy

This check fires when customer IP address belongs to a public proxy considered by MaxMind service. Counts Sale, Preauth or Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
filter parameter value	Y - to filter if condition is true, N - to filter if condition is false	Type: String Default: Y

Error codes

#	Code	Name
10253	1263	Customer IP address is public proxy

Customer IP Residential Proxy

This check fires when customer IP address belongs to a hosting or VPN provider considered by MaxMind service. Counts Sale, Preauth or Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
filter parameter value	Y - to filter if condition is true, N - to filter if condition is false	Type: String Default: Y

Error codes

#	Code	Name
10254	1264	Customer IP address is residential proxy

Customer IP Tor Exit Node

This check fires when customer IP address is a Tor exit node considered by MaxMind service. Counts Sale, Preauth or Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
filter parameter value	Y - to filter if condition is true, N - to filter if condition is false	Type: String Default: Y

Error codes

#	Code	Name
10255	1265	Customer IP address is tor exit node

Customer static IP score

This check fires when customer IP address static IP score which is considered by MaxMind service is lower or equal to the settled threshold value. Higher values meaning a greater static association. For example, many IP addresses with a user type of cellular have a score under one. Broadband IPs that don't change very often typically have a score above thirty. This indicator can be useful for deciding whether an IP address represents the same user over time. The value ranges from 0 to 99.99. Counts Sale, Preauth or Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
filter threshold value	Lower or equal values will be filtered out	Type: Decimal Default: 20.000

Error codes

#	Code	Name
10256	1266	Customer IP address static IP score is less or equal to configured threshold

Customer IP user count

This check fires when customer IP address user count considered by MaxMind service is higher or equal to the settled threshold value. The estimated number of users sharing the IP/network during the past 24 hours. For IPv4, the count is for the individual IP. For IPv6, the count is for the /64 network. Counts Sale, Preauth or Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
filter threshold value	Higher or equal values will be filtered out	Type: Decimal Default: 3

Error codes

#	Code	Name
10257	1267	Customer IP address user count is higher or equal to configured threshold

Customer IP user type

This check fires when customer IP address user type considered by MaxMind service is in blocked user types list. Possible values: business, cafe, cellular, college, consumer_privacy_network, content_delivery_network, government, hosting, library, military, residential, router, school, search_engine_spider, traveler. Counts Sale, Preauth or Transfer transactions.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
blocked user types	Blocked user types will be filtered out, split values with comma	Type: String Default: None

Error codes

#	Code	Name
10258	1268	Customer IP address user type is in blocked user types list

10.10.6 Consistency filters

Customer IP address Country differs from Issuing Country

This risk check is triggered when a transaction has the customer IP country different from the issuing country of the card. Requests from IP addresses listed in “Merchant API IP address” are ignoring this check. If parameter “apply for countries” is empty, filter will require strict customer country to issuer country matching for all the countries, otherwise this check will force country matching for listed countries only. For example if you setup “apply for countries” to US - check will be triggered for following country combinations US-anyNonUS or anyNonUS-US, but for combinations anyNonUS-anyNonUS and US-US the check will not fire. For card2card transactions issuer country of the source card should be equal to issuer country of the destination card, i.e. this check will be triggered for any cross-border transaction.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
ignore undefined countries	ignore check if country of the customer or issuer could not be defined	Type: String Default: Y
skip country identifier	ignore check for specific customer country, for example check could be skipped if customer uses mobile network with Opera browser proxy to process the transaction	Type: String Default: 250
apply for countries	if current parameters is not empty, check will be applied for listed countries only	Type: String

Error codes

#	Code	Name
10013	1023	Country of the customer does not correspond to the country of the issuer

Customer name differs from Cardholder name

This check fires when the provided customer name does not match cardholder name.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
greatest levenshtein distance	greatest levenshtein distance to consider customer and cardholder names equal	Type: Int Default: 3

Error codes

#	Code	Name
10114	1124	Customer name does not correspond to the cardholder name

Customer IP address differs from IP address used for 3-D Secure validation

This check fires when the provided Customer IP address does not match IP address used for 3-D Secure validation. Sometimes fraudsters are changing the destination of the payment converting sale operations (revocable operation) to card2card To exclude the automation of such fraud cases this filter could be used. Some providers are using dynamic IP addresses for their clients and during transaction processing customer IP address might be changed slightly. To avoid false positives in such cases IP address change in /24 subnet is allowed.

Score: Yes

Enabled by default: N

Error codes

#	Code	Name
10070	1080	Customer IP address have been changed during transaction processing

Customer birthday check

This check fires when the provided customer birthday is in incorrect format or customer is too young to perform requested operation.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
maximum age	maximum client age to process the transaction	Type: Int Default: 100
minimum age	minimum client age to process the transaction	Type: Int Default: 16

continues on next page

Table 227 – continued from previous page

Name	Description	Value
date format	input format for customers birthday, following macros are allowed - %Y Year, numeric, four digits - %m Month, numeric (00..12) - %d Day of the month, numeric (00..31)	Type: String Default: %Y%m%d

Error codes

#	Code	Name
10086	1096	Invalid customer birthday

Source Credit Card number expiration date check

This check fires when the provided Source Credit Card expiration date will expire soon. The time threshold is a moving window calculated backwards from the moment of the transaction. Usually card expires in the last day of the expiration month printed on card. Check could be used to avoid acceptance of the Credit Card for future preauthorized payments if it expires before the the last recurring payment planned.

Score: Yes

Enabled by default: N

Parameters

Name	Description	Value
minimum days before card expiration	minimum days before card expiration date	Type: Int Default: 0

Error codes

#	Code	Name
10126	1136	Card expires too soon

The 6+4 customer cards differs from the 6+4 passed in purpose

This check fires when the provided Customer 6+4 card does not match 6+4 passed in purpose. Counts Sale and Preauth transactions in any status.

Score: Yes

Enabled by default: N

Error codes

#	Code	Name
10249	1259	Customer card 6+4 does not match the purpose

10.10.7 Custom business validations

Transaction amount changing for Purpose

This check fires when the Transaction amount changing associated with exact Purpose exceeds the configured thresholds. The time threshold is a moving window calculated backwards from the moment of the transaction. So, if you set a quantity threshold of 10 transactions in 30 days, only 10 transactions with amount fitting parameter regexp below will be allowed, starting from 11th transactions its amount should fit regexp after parameter value. Counts Account verification, Sale, Preauth or Transfer transactions in Approved status.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
regexp below	if transaction amount fits this parameter value, "quantity threshold" current value increased by one	Type: String Default: ^([0-9]{1,2} 1[0-9]{2} 2[0-4][0-9])([.][0-9]{0,3})?\$\$
regexp after	regular expression value to fit all transaction amounts, starting from "quantity threshold" parameter value	Type: String ^(2[5-9][0-9][3-9][0-9]{2} [1-9][0-9]{3,10})([.][0-9]{0,3})?\$\$
quantity threshold	since specified threshold all transaction amounts should fit "regexp after" parameter value	Type: int Default: 99999
lookup period in days	lookup period to analyse transaction amount velocity in days	Type: int Default: 30
for all merchant projects	Y - to check transactions for all projects of the current merchant, otherwise check transactions for current project only	Type: String Default: N

Error codes

#	Code	Name
10080	1090	Invalid transaction amount

Authorization reattempts requirements EMEA

This check fires for all purchase transactions. Merchants that receive a decline response for a transaction will only be allowed to resubmit it for authorization up to fifteen times within 30 calendar days from the date of the original decline response if the response code is one of the following:

- Response Code 100-Denial (general, no comments)
- Response Code 101-Denial, expired card
- Response Code 102-Denial, suspected fraud
- Response Code 104-Denial, restricted card
- Response Code 106-Denial, allowable PIN tries exceeded
- Response Code 107-Denial, refer to card issuer
- Response Code 108-Denial, refer to card issuer's special conditions
- Response Code 109-Denial, invalid merchant
- Response Code 110-Denial, invalid amount
- Response Code 112-Denial, PIN data required
- Response Code 114-Denial, no account of type requested
- Response Code 115-Denial, requested function not supported
- Response Code 116-Denial, not sufficient funds
- Response Code 117-Denial, incorrect PIN
- Response Code 120-Denial, transaction not permitted to terminal
- Response Code 121-Denial, exceeds withdrawal amount limit
- Response Code 122-Denial, security violation
- Response Code 123-Denial, exceeds withdrawal frequency limit
- Response Code 124-Denial, violation of law
- Response Code 160-Denial, additional customer authentication required
- Response Code 197-Denial, call AmEx
- Response Code 198-Denial, call Card Processing Centre
- Response Code 903-Status message: re-enter transaction
- Response Code 904-Denial reason message: format error
- Response Code 907-Denial reason message: card issuer or switch inoperative
- Response Code 909-Denial reason message: system malfunction
- Response Code 913-Denial reason message: duplicate transmission
- Response Code 914-Denial reason message: not able to trace back to original transaction

- Response Code 921-Denial reason message: security software/hardware error - no action
- Response Code 950-Denial reason message: violation of business arrangement

International card systems Rules to prohibit acquirers and their recurring services merchants from resubmitting a declined transaction for authorization if it receives a response:

- Response Code 111-Denial, invalid card number
- Response Code 180-Denial, by cardholders wish
- Response Code 200-Pick-up (general, no comments)
- Response Code 207-Pick-up, special conditions
- Response Code 208-Pick-up, lost card
- Response Code 209-Pick-up, stolen card
- Response Code 908-Denial reason message: transaction destination cannot be found for routing

Visa only:

- Response Code 119-Denial, transaction not permitted to cardholder
- Response Code 902-Denial reason message: invalid transaction

Mastercard only:

- Response Code 201-Pick-up, expired card
- Response Code 202-Pick-up, suspected fraud
- Response Code 203-Pick-up, card acceptor contact card acquirer
- Response Code 204-Pick-up, restricted card
- Response Code 205-Pick-up, card acceptor call acquirer's security department
- Response Code 206-Pick-up, allowable PIN tries exceeded
- Response Code 210-Pick-up, suspected counterfeit card

The time threshold is a moving window calculated backwards from the moment of the transaction. Counts Account verification, Sale, Preauth transactions in Declined status for listed decline reasons. Correct decline reasons should be supported by a connected PSP. Limits are calculated separately for each merchant id, PAN and transaction amount.

Score: No

Enabled by default: N

Error codes

#	Code	Name
10182	1192	Payment system authorisation reattempts limit reached

Visa Preauthorized Transaction Decline Response requirements

This check fires for recurring transactions only. Merchants that receive a decline response for a preauthorized transaction will only be allowed to resubmit it for authorization up to four times within 16 calendar days from the date of the original decline response if the response code is one of the following:

- Response Code 05-Authorization Declined
- Response Code 51-Insufficient Funds
- Response Code 61-Exceeds Approval Amount Limit
- Response Code 65-Exceeds Withdrawal Frequency Limit

If an approval response is not received within this time frame, merchants must not resubmit the transaction or their acquirers may be subject to non-compliance actions, as outlined in the Visa Rules, and may be subject to chargebacks. Visa Rules to prohibit acquirers and their recurring services merchants from resubmitting a declined transaction for authorization if it receives a pickup response:

- Response Code 04-Pick Up Card
- Response Code 07-Pick Up Card, Special
- Response Code 33-Expired Card, Capture
- Response Code 34-Suspected Fraud, Retain Card
- Response Code 35-Card Acceptor, Contact Acquirer, Retain Card
- Response Code 36-Restricted Card, Retain Card
- Response Code 37-Contact Acquirer Security Department, Retain Card
- Response Code 41-Lost Card
- Response Code 43-Stolen Card
- Response Code 67-Capture Card or a decline response of:
- Response Code 14-Invalid Account Number (No Such Number)
- Response Code 54-Expired Card
- Response Code 57-Transaction Not Permitted

The time threshold is a moving window calculated backwards from the moment of the transaction. Counts Account verification, Sale, Preauth or Transfer transactions in Declined status for listed decline reasons. Correct decline reasons should be supported by a connected PSP. Limits are calculated separately for each gate descriptor.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
ignore period in months	if this parameter value greater zero, only one declined transaction allowed during the specified period, independently from its decline code	Type: Int Default: 0

Error codes

#	Code	Name
10090	1100	Visa rules violation for preauthorized transaction(DELAY)
10135	1145	PSP rules violation for no CVV transaction (CANCEL)
10136	1146	PSP rules violation for no CVV transaction (PICKUP)

MCC 6211 restrictions

Security Brokers/Dealers

Score: No

Enabled by default: N

Parameters

Name	Description	Value
allow MasterCard for NA	allows MasterCard processing for North America	Type: String Default: N
allow France	allows France processing	Type: String Default: N
allow Jamaica	allows Jamaica processing	Type: String Default: N
allow Uganda	allows Uganda processing	Type: String Default: N

Error codes

#	Code	Name
10099	1109	MCC 6211 rules violation

MCC 7995 restrictions

Betting/Casino Gambling

Score: No

Enabled by default: N

Parameters

Name	Description	Value
allow Germany	allows Germany processing	Type: String Default: N
allow UK	allows United Kingdom processing	Type: String Default: N

Error codes

#	Code	Name
10124	1134	MCC 7995 rules violation

CDB processing restrictions

Deny all countries except USA and EU countries. Also deny the following EU countries: Austria, Belgium, Denmark, Malta, Portugal, Romania, Slovakia, Estonia, Latvia; and the following US states: Alabama, Georgia, Missouri, South Dakota, Nebraska, North Carolina.

Score: No

Enabled by default: N

Parameters

Name	Description	Value
check credit card BIN country		Type: String Default: N
check customer billing address country		Type: String Default: N

continues on next page

Table 241 – continued from previous page

Name	Description	Value
check customer IP country		Type: String Default: N

Error codes

#	Code	Name
10167	1177	CDB processing restrictions violation.

10.10.8 Customer Level filters

These filters can be set up on Customer level in Customer Management System.

Customer id usage frequency for last 24 hours (daily limit)

This check fires when the number or amount of transactions associated with exact Customer id exceeds the configured thresholds. The time threshold is a 24 hours window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 24 hours. Counts Sale and Preauth transactions in the approved status for Deposits or Reversal, Transfer, Payout transactions in the approved status for Withdrawals.

score: NO

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last 24 hours for exact Customer id	Type: Decimal Default: 99999
for all merchant currencies in Customer Management Service	current total transactions amount or count for the last month for this customer id would be calculated - Yes: for all merchant levels in all currencies in payment direction (Deposit or Withdrawal) and converted to the currency of the current level for comparison with the amount limit - No: for current level only	Type: String Default: No

continues on next page

Table 243 – continued from previous page

Name	Description	Value
quantity limit	maximum total transactions count for the last 24 hours for exact Customer id	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: No
use calendar day	“Y” For calculation using calendar days instead of calculation from moment when filter was enabled “N” for calculation from moment when filter was enabled.	Type: String Default: No

Error codes

#	Code	Name
19000		Daily amount limit exceeded for customer id
19001		Daily quantity limit exceeded for customer id

Customer id usage frequency for last 7 days (weekly limit)

This check fires when the number or amount of transactions associated with exact Customer id exceeds the configured thresholds. The time threshold is a 7 days window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to hours. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in 168 hours. Counts Sale and Preauth transactions in the approved status for Deposits or Reversal, Transfer, Payout transactions in the approved status for Withdrawals.

score: Yes

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last week for exact Customer id	Type: Decimal Default: 99999

continues on next page

Table 245 – continued from previous page

Name	Description	Value
calendar week starts from Sunday	- Yes: calendar week starts from Sunday - No: calendar week starts from Monday	Type: String Default: No
for all merchant currencies in Customer Management Service	current total transactions amount or count for the last month for this customer id would be calculated Yes: for all merchant levels in all currencies in payment direction (Deposit or Withdrawal) and converted to the currency of the current level for comparison with the amount limit No: for current level only	Type: String Default: No
quantity limit	maximum total transactions count for the last 24 hours for exact Customer id	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: No
use calendar day	“Y” For calculation using calendar days instead of calculation from moment when filter was enabled “N” for calculation from moment when filter was enabled	Type: String Default: No

Error codes

#	Code	Name
19002		Weekly amount limit exceeded for customer id
19003		Weekly quantity limit exceeded for customer id

Customer id usage frequency for last month (monthly limit)

This check fires when the number or amount of transactions associated with exact Customer id exceeds the configured thresholds. The time threshold is a one month window calculated backwards from the moment of the transaction. For window calculation all transaction dates are truncated to days. The risk fires on the transaction after the set threshold. So, if you set a threshold of 10 transactions, it fires on the 11th transaction in one month. Month calculation based on calendar i.e. 28th, 29th, 30th and 31st of March would be bumped to 28th of February during window calculation. Counts Sale and Preauth transactions in the approved status for Deposits or Reversal, Transfer, Payout transactions in the approved status for Withdrawals.

score: Yes

Enabled by default: N

Parameters

Name	Description	Value
amount limit	maximum total transactions amount for the last 24 hours for exact Customer id	Type: Decimal Default: 99999
for all merchant currencies in Customer Management Service	current total transactions amount or count for the last month for this customer id would be calculated Yes: for all merchant levels in all currencies in payment direction (Deposit or Withdrawal) and converted to the currency of the current level for comparison with the amount limit No: for current level only	Type: String Default: No
quantity limit	maximum total transactions count for the last 24 hours for exact Customer id	Type: Int Default: 99999
subtract Cancel transactions	ignore cancel transactions for two-stage payments	Type: String Default: No

continues on next page

Table 247 – continued from previous page

Name	Description	Value
use calendar day	“Y” For calculation using calendar days instead of calculation from moment when filter was enabled “N” for calculation from moment when filter was enabled	Type: String Default: No

Error codes

#	Code	Name
19004		Monthly amount limit exceeded for customer id
19005		Monthly quantity limit exceeded for customer id

10.11 Glossary

Payment Gateway	Doc2.0 system that provides accepting, processing, storage and transmitting of payment data between members of payment process (Payer, Connecting Party, payment institutions).
Connecting Party	A third-party organization which implements integration to Payment Gateway following API Use-cases using API commands.
Merchant	User account in Payment Gateway provided to the merchant’s representatives.
Manager	User account in Payment Gateway provided to the representatives of PSP or payment institutions.
Reseller	User account in Payment Gateway provided to the agent, which engage merchants for PSP and payment institutions.
Payer	A person who pays for goods or services provided by Connecting Party. Also referred to as Customer.
Sender	A person who initiates the sending of funds from his side to another.
Receiver	A person who requests/receives funds from Sender or Connecting Party.
Acquirer	Member of payment system, which is licensed to accept payment cards.
Issuer	Member of payment system, which is licensed to issue payment cards.
Endpoint	Uniquely identified terminal in Payment Gateway, which is assigned to the Merchant and has to be provided in the commands within Payment Gateway API.
Endpoint Group	A set of endpoints with different currencies consolidated and available as one logical uniquely identified unit, which has to be provided in the commands within Payment Gateway API instead of endpoint.

continues on next page

Table 249 – continued from previous page

Master Endpoint	Special endpoints that combine auxiliary endpoints. When making a payment, the Payer can choose the payment method he needs on the payment form. When choosing a payment method, auxiliary transactions on the corresponding endpoints are called on the side of the Payment Gateway, which leads to different payment methods being initiated. Relevant for Payment Cashier integration. See Payment Cashier .
Auxiliary Endpoint	Regular endpoints united by master endpoint with their own settings and forms. Each auxiliary endpoint has its own payment method such as Credit card, Bank Transfer, Cryptocurrency, etc. Relevant for Payment Cashier integration. See Payment Cashier .
Project	Payment Gateway entity which determines the conditions for receiving a payment message and its further routing to the connected Processor.
Gate	A set of parameters, which identify account registered in a third-party processing system. These parameters can be used to process payment data in an external system using the messaging protocol implemented in the Processor.
Processor	Payment Gateway internal entity, which encapsulates interconnection with third-party processing system (e.g. Acquirer).
PAN	Payment card number or Primary account number.
RPI	Recurring Payment Identification number. Identifier of tokenized cardholder data, gathered in initial transaction, or registered without a transaction, that can be used multiple times for recurring transactions. Some Payment Gateway APIs are designed to accept RPI, others are designed to accept Card reference ID. See respective Use-Case for details.
Card reference ID	Identifier of tokenized cardholder data, gathered in initial transaction that can be used multiple times for recurring transactions. Some Payment Gateway APIs are designed to accept RPI, others are designed to accept Card reference ID. See respective Use-Case for details.
Form	Form in which payment data is indicated. In Form APIs, the payment form is hosted on Payment Gateway side. In Direct APIs, the payment form is hosted on Connecting Party side.
ACS	3DS Access Control Server.
3DS Method	A scripting call provided by the 3DS Integrator that is placed on the 3DS Requestor website (the Connecting Party payment form or Payment Gateway payment form). Optionally used to obtain additional browser information to facilitate risk-based decisioning.
PaReq	Payment authentication request message. See PaReq in 3DS Decision Making Schema of the respective Use-Case.
PaRes	Payment authentication response message. See PaRes in 3DS Decision Making Schema of the respective Use-Case.
CReq	Challenge Request Message. See CReq in 3DS Decision Making Schema of the respective Use-Case.
CRes	Challenge Response Message. See CRes in 3DS Decision Making Schema of the respective Use-Case.