THE Berlin GROUP
A EUROPEAN STANDARDS INITIATIVE

openFinance API Framewok

XS2A API as PSD2 Interface Implementation Guidelines

Version 2.4

31 October 2025

License Notice

[*] The 'Joint Initiative pan-European PSD2-Interface Interoperability' brings together participants of the Berlin Group with additional European banks (ASPSPs), banking associations, payment associations, payment schemes and interbank processors.

Contents

1 Introduction

1.1 From Core XS2A Interface to openFinance API

With [PSD2] the European Union has published a directive on payment services in the internal market. Among others [PSD2] contains regulations on services to be operated by so called Third Party Payment Service Providers (TPP) on behalf of a Payment Service User (PSU). These services are

- Payment Initiation Service (PIS) to be operated by a Payment Initiation Service Provider (PISP) TPP as defined by article 66 of [PSD2],
- Account Information Service (AIS) to be operated by an Account Information Service Provider (AISP) TPP as defined by article 67 of [PSD2], and
- Confirmation on the Availability of Funds Service (FCS) to be used by a Payment Instrument Issuing Service Provider (PIISP) TPP as defined by article 65 of [PSD2].

To implement these services (subject to PSU consent) a TPP needs to access the account of the PSU. The account is managed by another PSP called the Account Servicing Payment Service Provider (ASPSP). To support the TPP in accessing the accounts managed by an ASPSP, each ASPSP has to provide an "access to account interface" (XS2A interface). Such an interface has been defined in the Berlin Group NextGenPSD2 XS2A Framework.

This XS2A Framework is now extended to extended services and developed into a Version 2 API family. This interface is addressed in the following as **openFinance API**. This openFinance API differs from the XS2A interface in several dimensions:

- The extended services might not rely anymore solely on PSD2.
- Other important regulatory frameworks which apply are e.g. GDPR.
- The openFinance API can address different types of **API Clients** as access clients, e.g. TPPs regulated by an NCA according to PSD2, or corporates not regulated by an NCA.
- The extended services might require contracts between the access client and the ASPSP.
- While the client identification at the openFinance API can still be based on eIDAS certificates, they do not need to be necessarily PSD2 compliant eIDAS certificates.
- The extended services might require e.g. the direct involvement of the access client's bank for KYC processes.

Note: The notions of API Client and ASPSP are used because of the technical standardisation perspective of the openFinance API. These terms are analogous to "asset broker" and "asset holder" resp. in the work of the ERPB on a SEPA API access scheme.

Note: In implementations, the API services of several ASPSPs might be provided on an aggregation platform. Such platforms will be addressed in the openFinance API Framework as "API provider".

The following account access methods are covered by this framework:

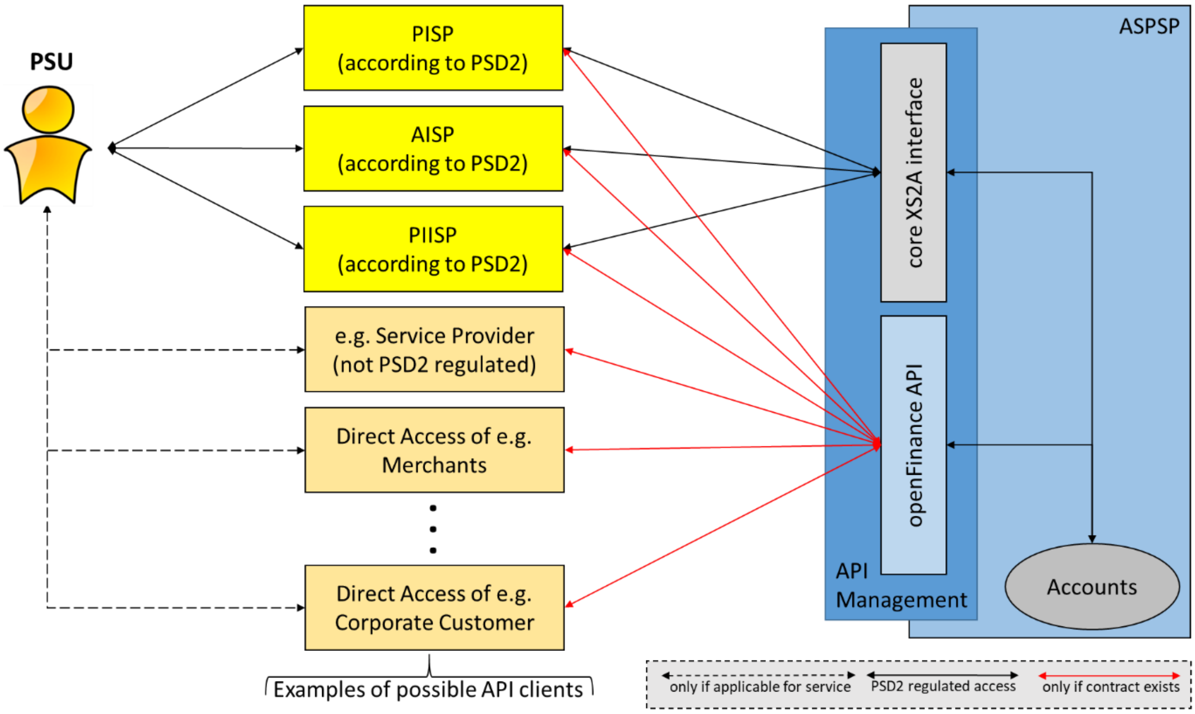

PSU
PISP (according to PSD2)
AISP (according to PSD2)
PIISP (according to PSD2)
e.g. Service Provider (not PSD2 regulated)
Direct Access of e.g. Merchants
Direct Access of e.g. Corporate Customer
Examples of possible API clients
ASPSP
core XS2A interface
openFinance API
API Management
Accounts
only if applicable for service
PSD2 regulated access
only if contract exists

Figure 1: Core XS2A interface and openFinance API

1.2 Core Compliance Services: XS2A API within the openFinance API Framework

The openFinance API Framework as displayed above can of course still be restricted to an **XS2A API** which is fulfilling core PSD2 compliant services accessible for all related and licensed TPPs, while sharing new architecture items and/or data modelling with the full openFinance API Framework.

This document covers the service definitions for the PSD2 compliance related part of the openFinance API Framework. Please note that this document is using securitiy measures and protocol functions as defined in the framework document [oFA PFSM]. This applies specifically to the usage of header parameters as defined in the framework documentation. This document also uses complex data structures and code lists in JSON syntax as defined in [oFA DaD].

1.3 Document Structure

This document specifies the core XS2A API Services within the openFinance API Framework in detail. Section 2 introduces an overview on the access methods available at the XS2A API. The details for the three service types as defined in [PSD2] are then described in the following sections:

- Section 3 introduces the Payment Initiation Services,
- Section 4 introduces the Account Information Services
- Section 5 introduces the Confirmation of Funds Services.

1.4 Document History

Version	Change/Note	Approved
Draft 2.0	Final Draft for public market consultation	5 April 2023
2.0	Published as compliance services	5 October 2023
2.1	Adaptation to openFinance API Framework Release 2.1	31 July 2024
2.2	Added timestamps for standing orders for startDate and endDate for instant payments Added an optional endpoint for retrieving the extended status of a JSON based bulk payment, in analogy to pain.002 formats Several errata and clarificatinos added	24 January 2025
2.3	Added new attribute cardBrand in the query parameters of Read Transaction List and Read Card Account Transaction List	10.March.2025
2.4	Adaption of changes in other openFinance API Framework documentation references Review of References in Section 6.	31 October 2025

Note: A detailed change log is contained in Section 7.

Remark for Future: Please note that the Berlin Group openFinance API Framework is still under constant development. Technical issues, which are already in discussion within the Berlin Group openFinance working structure are mentioned in this document by "Remark for Future" to make the reader aware of upcoming potential changes.

2 Application Layer: Guiding Principles

2.1 API Access Methods

The following tables gives an overview on the HTTP access methods supported by the API endpoints and by resources created through this API.

Conditions in the following tables

It is further defined, whether this method support is mandated for the ASPSP by this specification or whether it is an optional feature for the ASPSP. Please note that this condition is given relative to the parent node of the path, i.e. the condition e.g. on a method on /v2/payments/{payment-product}/{paymentId} applies only if the endpoint /v2/payments/{payment-product} is supported at all.

NOTE: All methods submitted by a TPP, which are addressing dynamically created resources in this API, may only apply to resources which have been created by the same TPP before.

Examples

Please further note, that sections are referred in the Description's column. These sections provide example for all related access methods.

2.1.1 Payments Endpoints

Endpoints/Resources	Method	Condition	Description
payments/{payment-product}	POST	Mandatory	Create a payment initiation resource addressable under {paymentId} with all data relevant for the corresponding payment product. This is the first step in the API to initiate the related payment. Section 3.4.1 and 3.4.2
payments/{payment-product}/{paymentId}	GET	Mandatory	Read the details of an initiated payment. Section 3.7
payments/{payment-product}/{paymentId}/status	GET	Mandatory	Read the transaction status of the payment Section 3.5

Endpoints/Resources	Method	Condition	Description
bulk-payments/{payment-product}	POST	Optional	Create a bulk payment initiation resource addressable under {paymentId} with all data relevant for the corresponding payment product. This is the first step in the API to initiate the related bulk payment. Section 3.4.3
bulk-payments/{payment-product}/{paymentId}	GET	Mandatory	Read the details of an initiated bulk payment. Section 3.7
bulk-payments/{payment-product}/{paymentId}/status	GET	Mandatory	Read the transaction status of the bulk payment Section 3.5
bulk-payments/{payment-product}/{paymentId}/extended-status	GET	Optional	Read the extended transaction status of the bulk payment. The response contains an array with status information for every rejected transaction. This is an analog structure to pain.002 for JSON based bulk payments. Section 3.6
periodic-payments/{payment-product}	POST	Optional	Create a standing order initiation resource for recurrent i.e. periodic payments addressable under {paymentId} with all data relevant for the corresponding payment product and the execution of the standing order. This is the first step in the API to initiate the related recurring/periodic payment. Section 3.4.4

Endpoints/Resources	Method	Condition	Description
periodic-payments/{payment-product}/{paymentId}	GET	Mandatory	Read the details of an initiated standing order for recurring/periodic payments. Section 3.7
periodic-payments/{payment-product}/{paymentId}/status	GET	Mandatory	Read the transaction status of the standing order for recurring/periodic payments. Section 3.5
{payment-service}/{payment-product}/{paymentId}	DELETE	Optional	Cancels the addressed payment with resource identification paymentId if applicable to the payment-service, payment-product and received in product related timelines (e.g. before end of business day for scheduled payments of the last business day before the scheduled execution day). The response to this DELETE command will tell the TPP whether the • access method was rejected • access method was successful, or • access method is generally applicable, but further authorisation processes are needed. Section 3.8

NOTE: Access methods on authorisation related sub-endpoints for payment initiation are not described in this overview, e.g. the access methods for the endpoint v2/payments/{payment-product}/{paymentId/authorisations. Please refer to the related sections in [oFA PFSM] for more details on SCA and authorisation related access methods or to the related OpenAPI file.

2.1.2 Accounts Endpoint

Endpoints/Resources	Method	Condition	Description
accounts	GET	Mandatory	Read all identifiers of the accounts, to which an account access has been granted to through the /consents endpoint by the PSU. In addition, relevant information about the accounts and hyperlinks to corresponding account information resources are provided if a related consent has been already granted. **Remark:** Note that the standard offers the option of an ASPSPto grant an access to the account list of all current accounts of a PSU. In this case, this endpoint will deliver the the list of all addressable current accounts of the PSU at this ASPSP. Section 4.4.1
accounts?withBalance	GET	Optional	Read the identifiers of the payment account addressed implicitly by the related consent together with booking balance information, depending on the consent granted Section 4.4.1
accounts/{account-id}	GET	Mandatory	Give detailed information about the addressed account. Section 4.4.2
accounts/{account-id}?withBalance	GET	Optional	Give detailed information about the addressed account together with balance information Section 4.4.2
accounts/{account-id}/balances	GET	Mandatory	Give detailed balance information about the addressed account

Endpoints/Resources	Method	Condition	Description
			Section 4.4.3
accounts/{account-id}/transactions	GET	Mandatory	Read transaction lists of a given account. For a given account, additional parameters are e.g. the attributes "dateFrom" and "dateTo". The ASPSP might add balance information, if transaction lists without balances are not supported. Section 4.4.4
accounts/{account-id}/transactions?withBalance	GET	Optional	Read transaction lists of a given account, depending on the steering parameter "bookingStatus" together with balances. Section 4.4.4
accounts/{account-id}/transactions/{transactionId}	GET	Optional	Read transaction details of an addressed transaction. Section 4.4.5

NOTE: All access methods on the v2/accounts API need a technical consent mechanisms first as specified in [oFA-CO] for the endpoints v2/consents/account-access.

Remark: Note that the {account-id} parameters are resource identifications and thus can be tokenised by the ASPSP such that the actual account numbers like IBANs or PANs are not part of the path definitions of the API for data protection reasons. This tokenisation is managed by the ASPSP. It is highly recommended by this specification to use UUIDs for the resource identfications.

2.1.3 Card-accounts Endpoint

This endpoint delivers credit card account related account information, where the account is used to reconcile credit card transactions with the PSU. This endpoint is not directly related to credit cards as such, but the financial account behind the related cards.

Remark: The access methods to card accounts are less detailed compared to access methods to accounts due to the reduced functionality compared to generic payment accounts.

Endpoints/Resources	Method	Condition	Description
card-accounts	GET	Optional	Read all identifiers of the card accounts, to which an account access has been granted to through the /consents endpoint by the PSU. In addition, relevant information about the card accounts and hyperlinks to corresponding account information resources are provided if a related consent has been already granted. Section 4.5.1
card-accounts/{account-id}	GET	Optional	Give detailed information about the addressed card account. Section 4.5.2 **Remark for Future**: This endpoint might be made mandatory for future versions of the specification.
card-accounts/{account-id}/balances	GET	Optional	Give detailed balance information about the addressed card account. Section 4.5.3 **Remark for Future**: This endpoint might be made mandatory for future versions of the specification.
card-accounts/{account-id}/transactions	GET	Mandatory	Read transaction lists related to a given card account. For a given card account, additional parameters are e.g. the attributes "dateFrom" and "dateTo".

Endpoints/Resources	Method	Condition	Description
			Section 4.5.4

NOTE: All access methods on the v2/card-accounts API need a technical consent mechanisms first as specified in [oFA-CO] for the endpoint v2/consents/account-access.

Remark: Note that the {card-account-id} parameters are resource identifications and thus aretokenised by the ASPSP such that the actual card account or card number like IBANs or PANs are not part of the path definitions of the API for data protection reasons. This tokenisation is managed by the ASPSP. It is highly recommended by this specification to use UUIDs for the resource identfications.

2.1.4 Funds-Confirmations Endpoint

Endpoints/Resources	Method	Condition	Description
funds-confirmations	POST	Mandatory	Checks whether a specific amount is available at point of time of the request on an account linked to a given tuple card issuer(TPP)/card number, or addressed by IBAN and TPP respectively Section 5.2

NOTE: The access to the v2/funds-confirmations endpoint needs a consent between ASPSP and PSU. This consent might be managed in a bank branch or via the online channels. In addition, this consent mechanism can be offered as an extended service via the v2/consents/funds-confirmations endpoints as specified in [oFA-CO]. In this case, the TPP needs to use the related consent token to access the v2/funds-confirmations endpoint.

2.2 Multicurrency Accounts Specifics for Compliance Services

The concept of a multicurrency account is introduced in [oFA PFSM].

Remark: The multi-currency account product is in use in some markets in Europe, e.g. in Online-Banking products within the Belgium market. The support of this functionality in the XS2A API is only applicable in these markets.

2.3 Status Information

2.3.1 Status Information for PIS

The backend systems of ASPSPs are supporting for payments a transaction status, which is defined in the ISO20022 and is addressed in this specification as the data element "transactionStatus". ASPSPs will deliver this status within all response messages after a payment initiation resource has been established and if no error occurs.

The transaction status of a payment initiation is changing during the initiation process, depending on the results of sub-steps like format checks, SCA checks, PSU related profile checks, funds availability checks or depending on the start of backend clearing processes. At the end of a payment process, the transaction status in the ASPSPs backend is either “RJCT”, which stands for "Rejected", or “ACSC", which stands for "AcceptedSettlementCompleted” where complete is here referring to the debtor account. For instant payments, the additional transaction status "ACCC", which stands for "AcceptedSettlementCompleted" regarding the creditor account might be used in addition. Depending on the booking process of the ASPSP, the risk of the actual payment, the financial account status of the PSU account or the initiation date and time, the latter status might be reached after some period and after the payment initiation process as such has been finalised. These later transaction statuses do not need to be reflected in the XS2A interface which is only providing the status information immediately after the initiation of the payment.

A typical end status within a successful PIS process for a batch booking process is therefore

- "ACTC" which stands for “AcceptedTechnicalCorrect”, where the PSU authentication, syntactical and semantical (product) checks had been successful,
- "ACWC" which stands for "AcceptedWithChanges", where the PSU authentication, syntactical and semantical (product) checks had been successful and the ASPSP is informing the PISP that some changes have been applied to the payment initiation, e.g. on the requested execution date,
- "ACCP“, which stands for "AcceptedCustomerProfile”, where in addition the financial risk profile of the PSU has been checked positively, or
- "ACFC“, which stands for "AcceptedFundsChecked”, where in addition to the customer profile the funds availability has been checked positively.

Realtime booking processes for batch payments might result for the time period of the payment initiation in

- "ACSP", which stands for “AcceptedSettlementInProcess”, where the settlement routine regarding the debtor account of the payment has already been initiated.

- "ACSC", which stands for “AcceptedSettlementCompleted”, indicating that the money has been booked already from the debtor account.

For instant payments, the final backend status "ACCC", which stands for "AcceptedSettlementCompleted" regarding the creditor account, will normally be reported at the end of the payment initiation process.

For payments which need more than one PSU to authorise the transaction, an additional status code as analogon to "ACTC" is introduced:

- "PACT", which stands for "PartiallyAcceptedTechnicalCorrect", indicating that some authorisations have been applied successfully, but more authorisations are need to process the transaction.

In bulk payment initiation, ASPSPs might choose either to process the bulk only partially and reject some of the contained payments. This results in

- "PART", which stands for "PartiallyAccepted", indicating that all mandated authorisations have been applied, but not all payment have been transformed due to other reasons.

Funds Availability

For ASPSP, which are not booking the money directly from the account, this specification provides the optional data element

"fundsAvailable": true/false

to be used together with the codes "ACTC", "ACWC" and "ACCP" in a GET Status Response Message early in the process chain to indicate that a funds check has been processed with the indicated result. This is the same data element as used in the confirmation of funds request and might be used by the ASPSP to inform the PISP about the funds availability, following requirements from [EBA-RTS].

Even if the funds check has been positive, the payment might be rejected later during the batch booking phase due to other bookings on the account. In case of no funds available, the payment might not be rejected yet due to the practice of the ASPSP in online channels, that it will wait for liquidity for a certain period.

Example 1: Batch booking bank, no profile checks but funds available positive

```
{"transactionStatus": "ACTC",
 "fundsAvailable": true}
```

Example 2: Batch booking bank, profile check positive, no funds available, no rejection yet

```
{"transactionStatus": "ACCP",
 "fundsAvailable": false}
```

The ASPSP might also use the status "PDNG", which stands for “Pending” to inform the TPP about the fact, that the next status of the payment has not been reached yet.

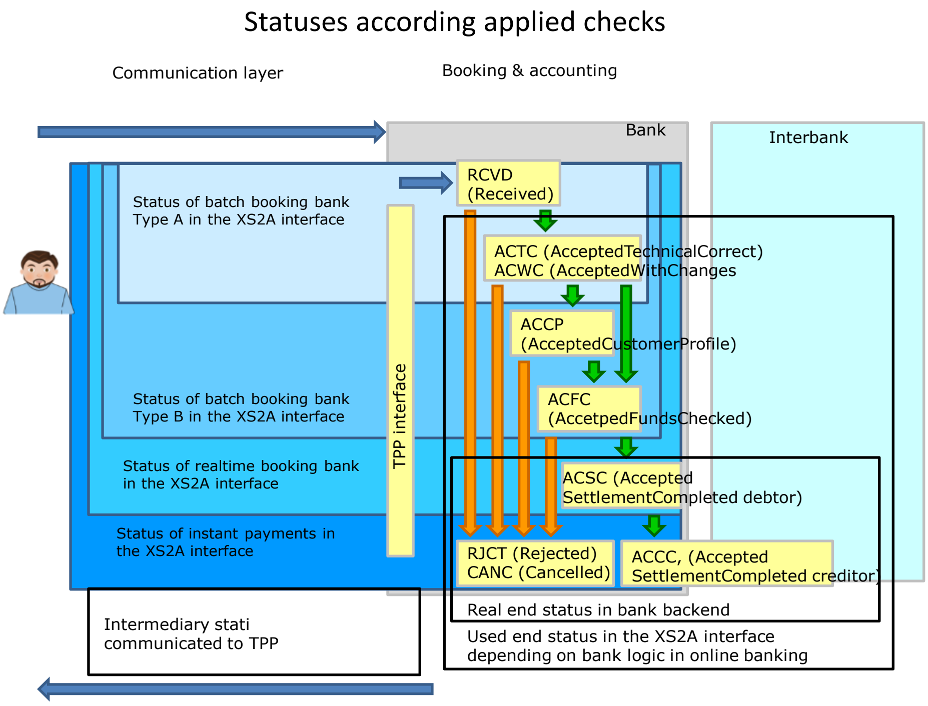
In addition, the ASPSP will inform the TPP about the status of the technical SCA process for a payment initiation within the GET SCA Status Response Message. For this status reporting the data element “scaStatus” is used.

Future dated payments and periodic payments

Future dated payments and periodic payments are both payment types which are not directly executed after initiation. For both types of payments, ASPSPs might have a reduced or no check on customer profile or funds availability due to the fact that the actual payments are performed later. The end status during the payment initiation process then is "ACTC" or "ACCP" depending on the ASPSPs procedures in its online channels. The fundsAvailable data element might be contained in addition, in case a funds availability check has been performed during payment initiation.

Status Model Overview

The following picture gives an overview on the transaction status:


Statuses according applied checks
Communication layer
Booking & accounting
Bank
Interbank
Status of batch booking bank Type A in the XS2A interface
Status of batch booking bank Type B in the XS2A interface
Status of realtime booking bank in the XS2A interface
Status of instant payments in the XS2A interface
TPP interface
RCVD (Received)
ACTC (AcceptedTechnicalCorrect)
ACWC (AcceptedWithChanges
ACCP (AcceptedCustomerProfile)
ACFC (AccetpedFundsChecked)
ACSC (Accepted SettlementCompleted debtor)
RJCT (Rejected)
CANC (Cancelled)
ACCC, (Accepted SettlementCompleted creditor)
Real end status in bank backend
Used end status in the XS2A interface depending on bank logic in online banking
Intermediary stati communicated to TPP

Status of cancelled Payments

After a successful cancellation of a payment initiation, the corresponding transaction status transforms to "CANC" for cancelled. This transaction status will be returned as long as the cancelled payment initiation resource is addressable.

Status of partially authorised payments within a multilevel SCA process

Payment initiations which are at least authorised by one PSU, but which are not yet finally authorised by all applicable PSU will be transformed into the status "PATC" for "PartiallyAcceptedTechnicalCorrect".

2.3.2 Status Information for the AIS within the Establish Consent Process

As described above, the TPP needs a technical consent token for accessing account related information. The lifecycle of these tokens and related status information is specified in [oFA-CO].

3 Payment Initiation Service

Remark: The API design differs across the various SCA approaches (Embedded, Redirect, OAuth2 or Decoupled, cp.[oFA-OR-Com]), but most between the Embedded SCA Approach and the others, since the Embedded SCA Approach demands the support of the full SCA complexity within the API itself. For that reason, all data or processes, which are needed for the Embedded SCA Approach only, are shown with a light blue background, to increase the readability of the specification.

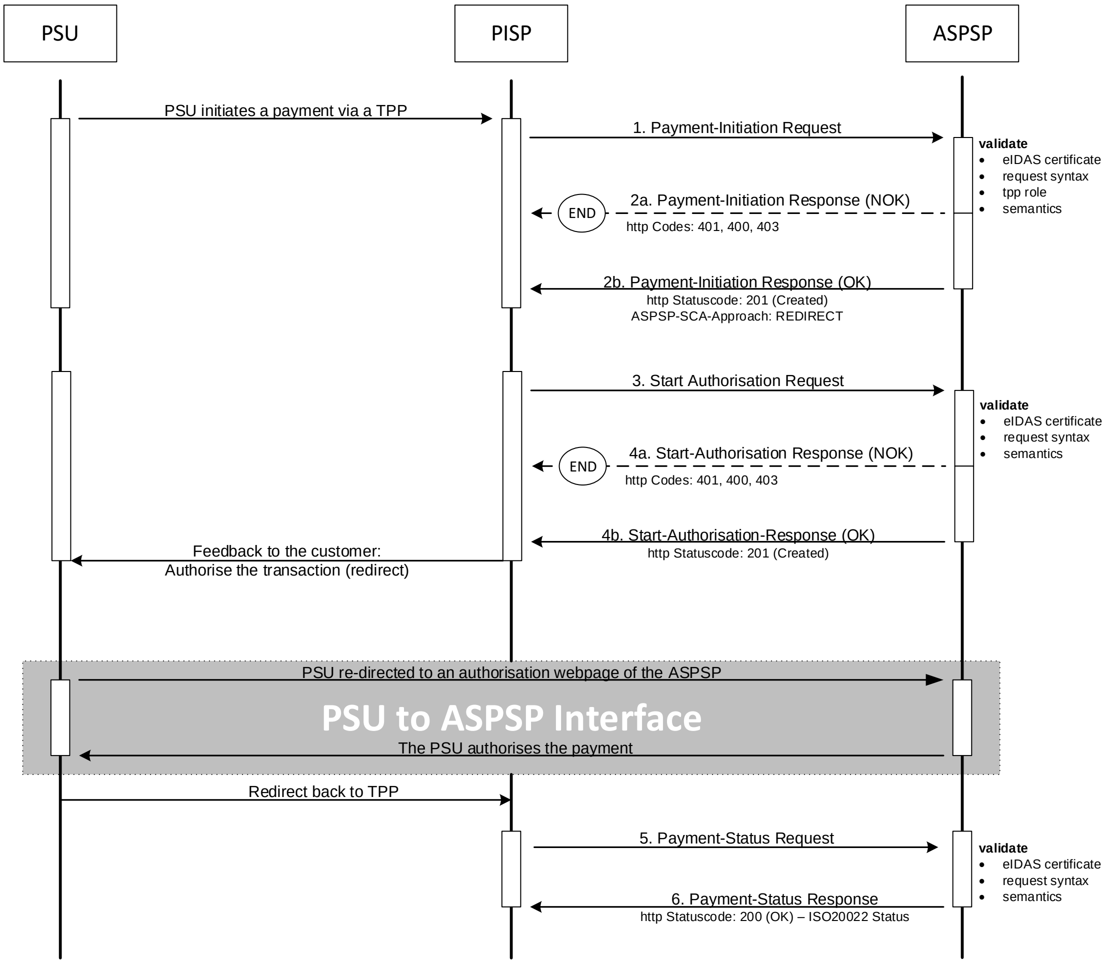
3.1 Payment Initiation Flows

The payment initiation flow depends heavily on the SCA approach implemented by the ASPSP. The most complex flow is the flow for the Embedded SCA Approach, which further differs on whether there are various authentication methods available for the PSU. In the following, the different API flows are provided as an overview for these different scenarios.

> **Remark:** The flows do not always cover all variances or complexities of the implementation and are exemplary flows.

3.1.1 Redirect SCA Approach: Explicit Start of the Authorisation Process

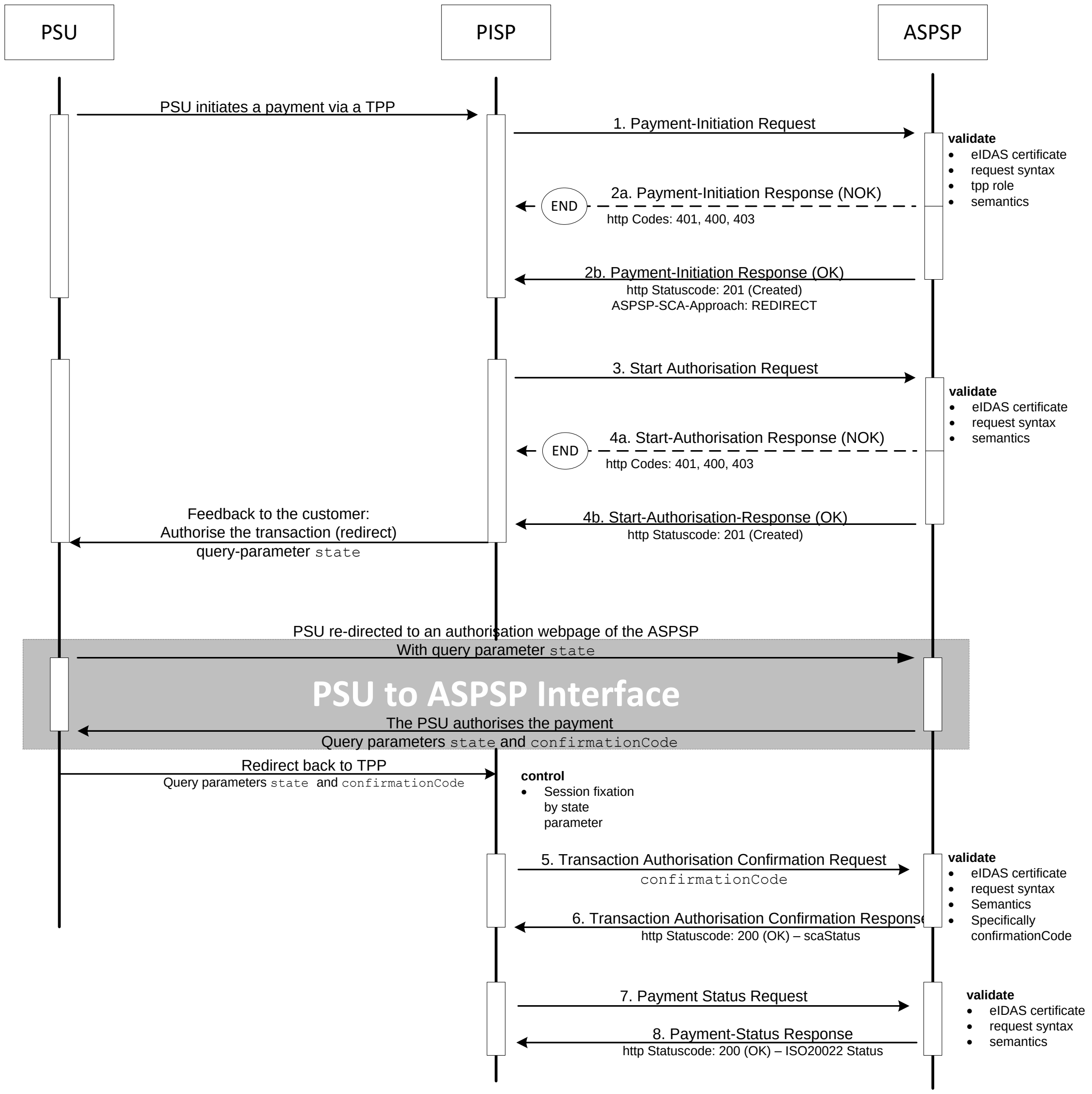
If the ASPSP supports the Redirect SCA Approach, the message flow within the payment initiation service is simple. The Payment Initiation Request is followed by an explicit request of the TPP to start the authorisation. This is followed by a redirection to the ASPSP SCA authorisation site. A status request might be requested by the TPP after the session is re-redirected to the TPP's system.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
3. Start Authorisation Request
validate
• eIDAS certificate
• request syntax
• semantics
END
4a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
4b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
Feedback to the customer:
Authorise the transaction (redirect)
PSU re-directed to an authorisation webpage of the ASPSP
PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
5. Payment-Status Request
validate
• eIDAS certificate
• request syntax
• semantics
6. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

3.1.2 Redirect SCA Approach: Explicit Start of the Authorisation Process with Confirmation Code

In addition to the scenario above, an authorisation confirmation request might be requested by the ASPSP from the TPP after the session is re-redirected to the TPP's system and after the

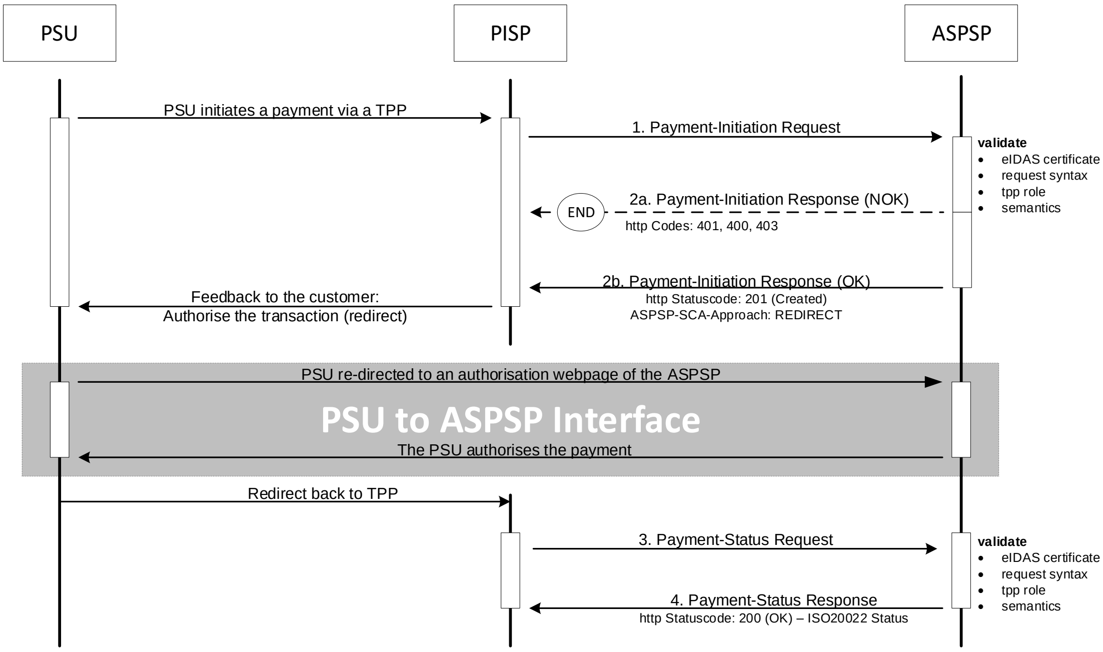
TPP's control on session fixation. In the end, a payment status request might be needed by the TPP to control the exact status of the payment initiation.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
3. Start Authorisation Request
validate
• eIDAS certificate
• request syntax
• semantics
END
4a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
Feedback to the customer:
Authorise the transaction (redirect)
query-parameter state
4b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
PSU re-directed to an authorisation webpage of the ASPSP
With query parameter state
PSU to ASPSP Interface
The PSU authorises the payment
Query parameters state and confirmationCode
Redirect back to TPP
Query parameters state and confirmationCode
control
• Session fixation by state parameter
5. Transaction Authorisation Confirmation Request
confirmationCode
validate
• eIDAS certificate
• request syntax
• Semantics
• Specifically confirmationCode
6. Transaction Authorisation Confirmation Respons
http Statuscode: 200 (OK) – scaStatus
7. Payment Status Request
validate
• eIDAS certificate
• request syntax
• semantics
8. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

3.1.3 Redirect SCA Approach: Implicit Start of the Authorisation Process

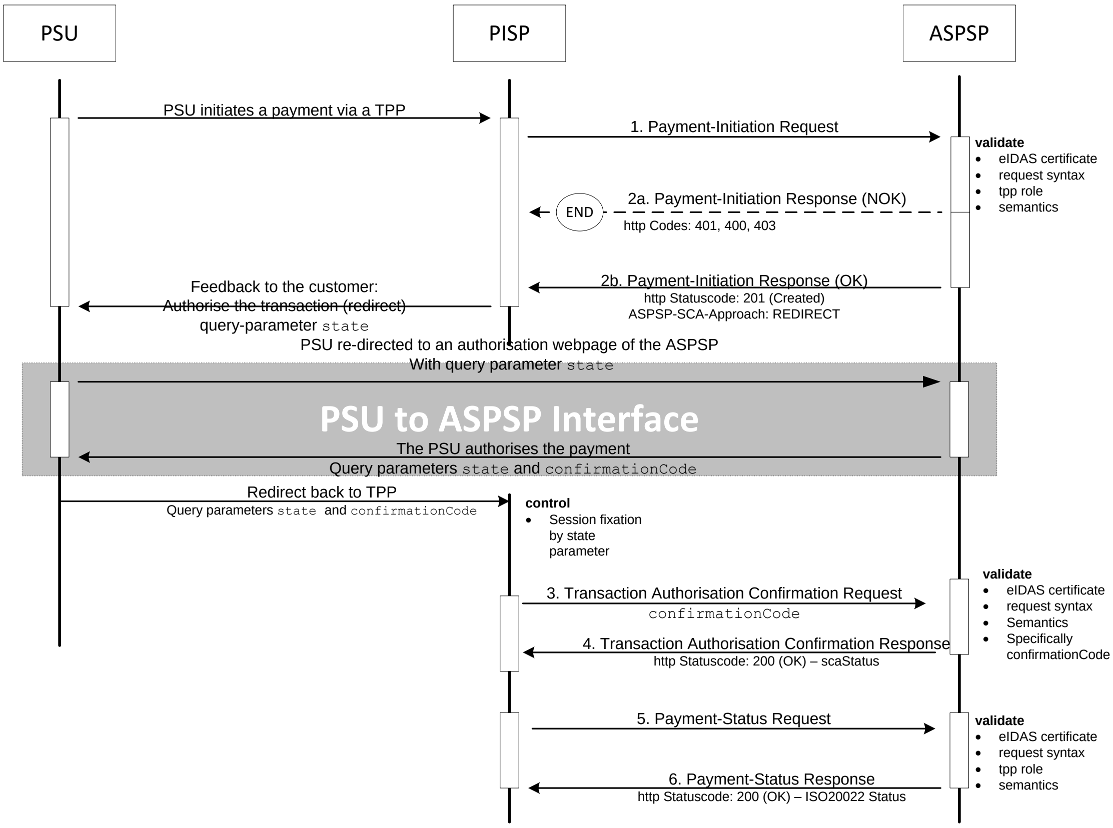
ASPSPs might start the authorisation process implicitly in case of no additional data is needed from the TPP. This optimisation process results in the following flow (which is exactly the Redirect SCA Approach flow from the version 1.0 and 1.1 of the Implementation Guideline before authorisation sub-resources have been established). In this case, the redirection of the PSU browser session happens directly after the Payment Initiation Response. In addition an

SCA status request can be sent by the TPP to follow the SCA process (not shown in the diagram).


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
Feedback to the customer:
Authorise the transaction (redirect)
PSU re-directed to an authorisation webpage of the ASPSP
PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
3. Payment-Status Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
4. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

3.1.4 Redirect SCA Approach: Implicit Start of the Authorisation Process with Confirmation Code

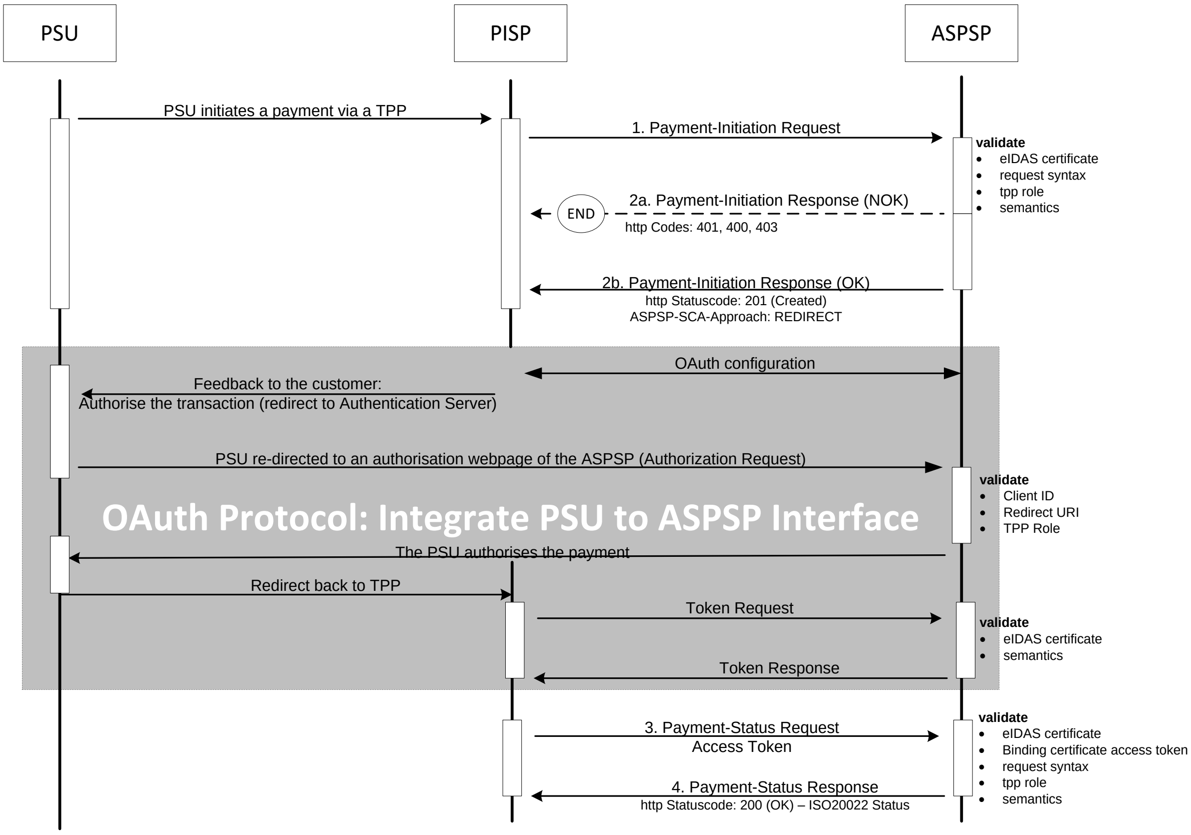
In addition to the scenario above, an authorisation confirmation request might be requested by the ASPSP from the TPP after the session is re-redirected to the TPP's system and after the TPP's control on session fixation. In the end, a payment status request might be needed by the TPP to control the exact status of the payment initiation.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
Feedback to the customer:
~~Authorise the transaction (redirect)~~
query-parameter state
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
PSU re-directed to an authorisation webpage of the ASPSP
With query parameter state
PSU to ASPSP Interface
The PSU authorises the payment
Query parameters state and confirmationCode
Redirect back to TPP
Query parameters state and confirmationCode
control
• Session fixation by state parameter
validate
• eIDAS certificate
• request syntax
• Semantics
• Specifically confirmationCode
3. Transaction Authorisation Confirmation Request
confirmationCode
4. Transaction Authorisation Confirmation Response
http Statuscode: 200 (OK) – scaStatus
5. Payment-Status Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
6. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

3.1.5 OAuth2 SCA Approach: Implicit Start of the Authorisation Process

If the ASPSP supports the OAuth2 SCA Approach, the flow is very similar to the Redirect SCA Approach with implicit start of the Authorisation Process. Instead of redirecting the PSU directly to an authentication server, the OAuth2 protocol is used for the transaction authorisation process.

Remark: The OAuth2 SCA Approach with explicit start of the Authorisation Process is treated analogously.

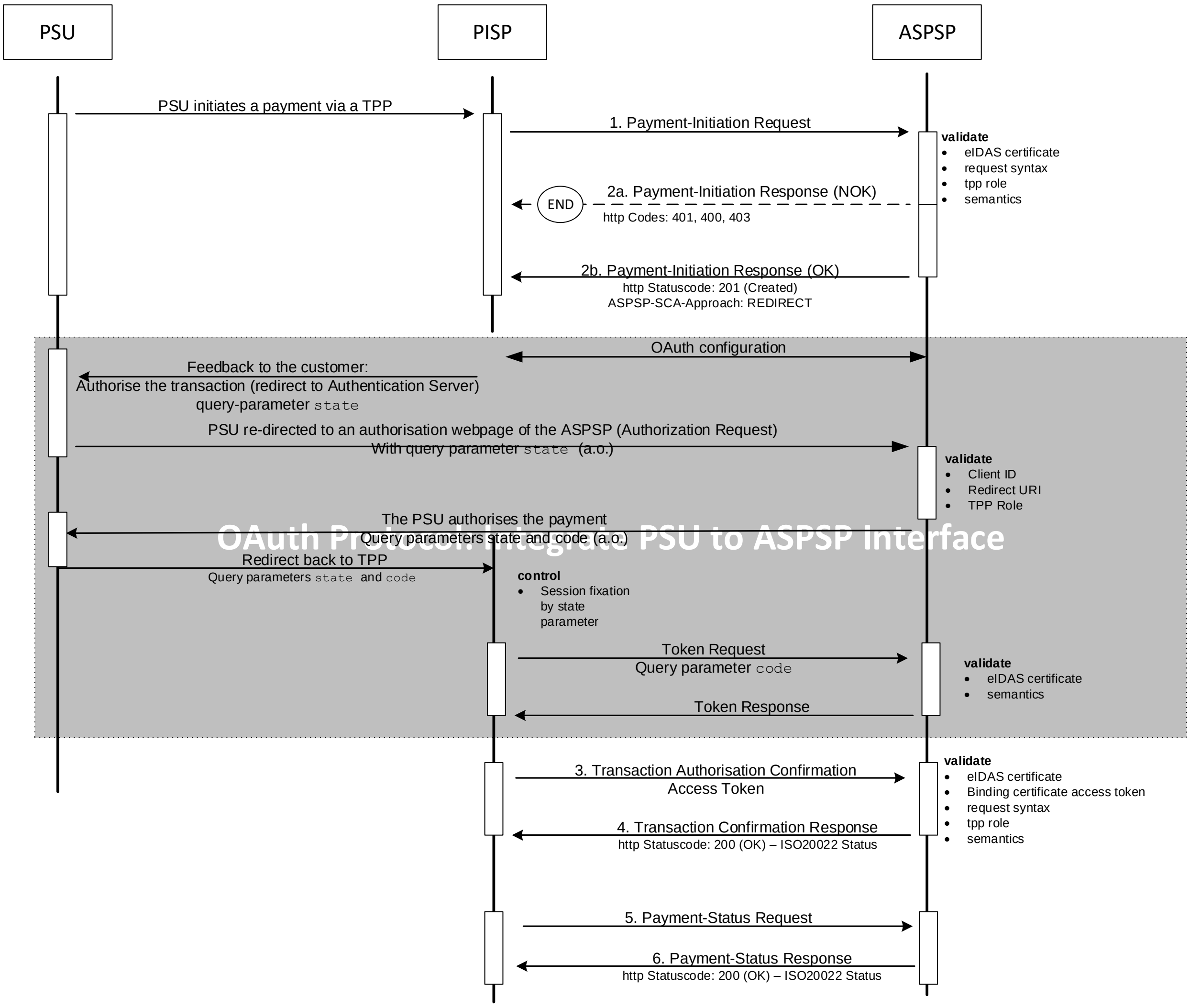

PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
OAuth configuration
Feedback to the customer:
Authorise the transaction (redirect to Authentication Server)
PSU re-directed to an authorisation webpage of the ASPSP (Authorization Request)
validate
Client ID
Redirect URI
TPP Role
OAuth Protocol: Integrate PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
Token Request
validate
eIDAS certificate
semantics
Token Response
3. Payment-Status Request
Access Token
validate
eIDAS certificate
Binding certificate access token
request syntax
tpp role
semantics
4. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

3.1.6 OAuth2 SCA Approach: Implicit Start of the Authorisation Process with Confirmation Code

In addition to the scenario above, an authorisation confirmation request might be requested by the ASPSP from the TPP after the session is re-redirected to the TPP's system and after the

TPP's control on session fixation. In the end, a payment status request might be needed by the TPP to control the exact status of the payment initiation.

Remark: The OAuth2 SCA Approach with explicit start of the Authorisation Process and with transaction confirmation step is treated analogously.

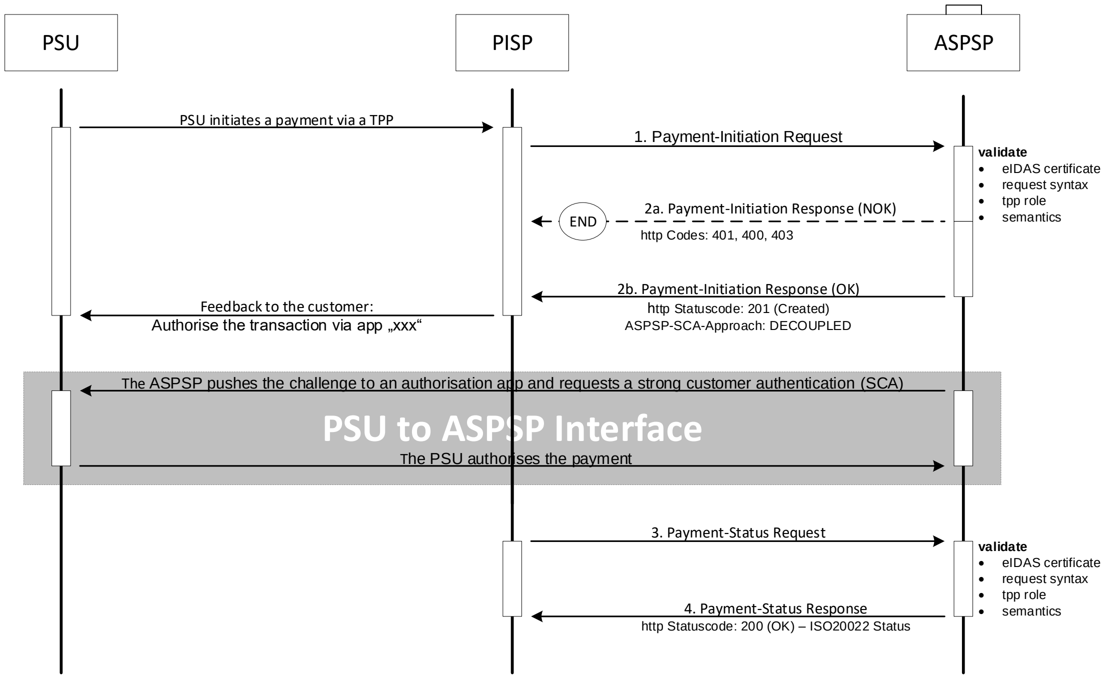

PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
OAuth configuration
Feedback to the customer:
Authorise the transaction (redirect to Authentication Server)
query-parameter state
PSU re-directed to an authorisation webpage of the ASPSP (Authorization Request)
With query parameter state (a.o.)
validate
• Client ID
• Redirect URI
• TPP Role
The PSU authorises the payment
Query parameters state and code (a.o.)
OAuth Protocol: Integrate PSU to ASPSP interface
Redirect back to TPP
Query parameters state and code
control
• Session fixation by state parameter
Token Request
Query parameter code
validate
• eIDAS certificate
• semantics
Token Response
3. Transaction Authorisation Confirmation
Access Token
validate
• eIDAS certificate
• Binding certificate access token
• request syntax
• tpp role
• semantics
4. Transaction Confirmation Response
http Statuscode: 200 (OK) – ISO20022 Status
5. Payment-Status Request
6. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

It is further recommended for ASPSPs and TPPs in this case to follow the Security Best Practice definitions as defined in [OA-SecTop]. This reference will also be added in the next version of the Implementation Guidelines.

3.1.7 Decoupled SCA Approach: Implicit Start of the Authorisation Process

The transaction flow in the Decoupled SCA Approach is similar to the Redirect SCA Approach. The difference is that the ASPSP is asking the PSU to authorise the payment e.g. via a dedicated mobile app, or any other application or device which is independent from the online banking frontend. The ASPSP is asking the TPP to inform the PSU about this authentication by sending a corresponding PSU Message like "Please use your xxx App to authorise the payment".

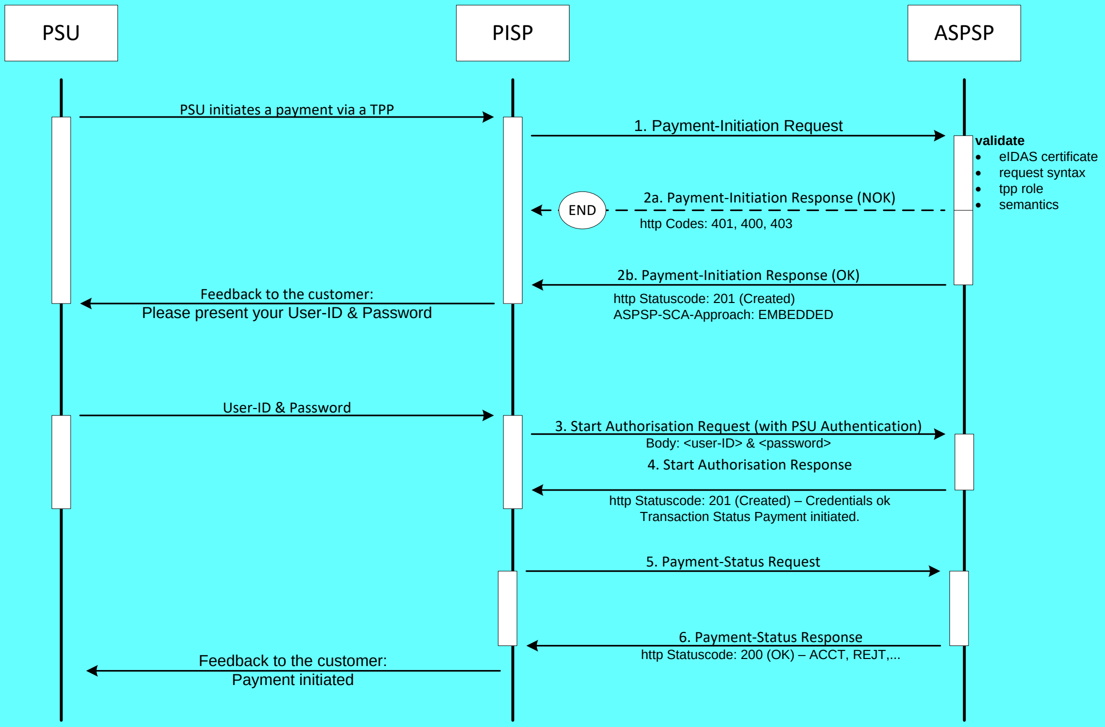
After the SCA having been processed between ASPSP and PSU, the TPP then needs to ask for the result of the transaction. In the following, a flow with an implicit start of the authorisation process is shown:


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: DECOUPLED
Feedback to the customer:
Authorise the transaction via app „xxx“
The ASPSP pushes the challenge to an authorisation app and requests a strong customer authentication (SCA)
PSU to ASPSP Interface
The PSU authorises the payment
3. Payment-Status Request
validate
eIDAS certificate
request syntax
tpp role
semantics
4. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

3.1.8 Embedded SCA Approach without SCA method (e.g. Creditor in Exemption List)

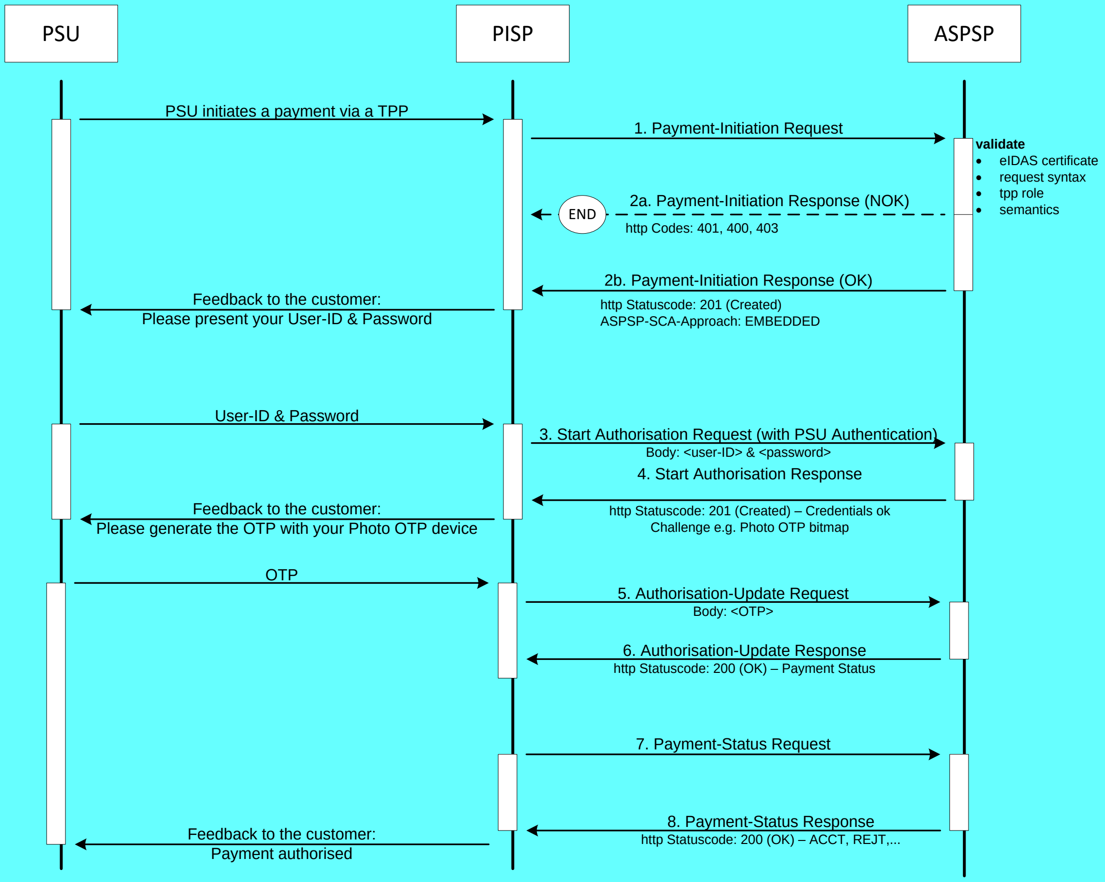
In the following, several exemplary flows are shown, where the ASPSP has chosen to process the SCA methods through the PISP – ASPSP interface. In any case, the PSU normally will need to authenticate himself with a first factor, before any account or SCA method details will be available to the PISP. So even in case where the Payment Initiation is accepted without an SCA method due e.g. to an exemption list, the PSU is asked via the PISP to provide the PSU Identification and e.g. a password or an OTP. The later exemplary flows then will show scenarios, where complexities like SCA processing and choosing an SCA method will be added.

Remark: In case where OAuth2 is requested by the ASPSP as a pre-step for PSU authentication, the sequence of the PSU authentication with the first authentication factor is omitted. This applies also for all examples for the Embedded SCA Approach.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: EMBEDDED
Feedback to the customer:
Please present your User-ID & Password
User-ID & Password
3. Start Authorisation Request (with PSU Authentication)
Body: <user-ID> & <password>
4. Start Authorisation Response
http Statuscode: 201 (Created) – Credentials ok
Transaction Status Payment initiated.
5. Payment-Status Request
6. Payment-Status Response
http Statuscode: 200 (OK) – ACCT, REJT,...
Feedback to the customer:
Payment initiated

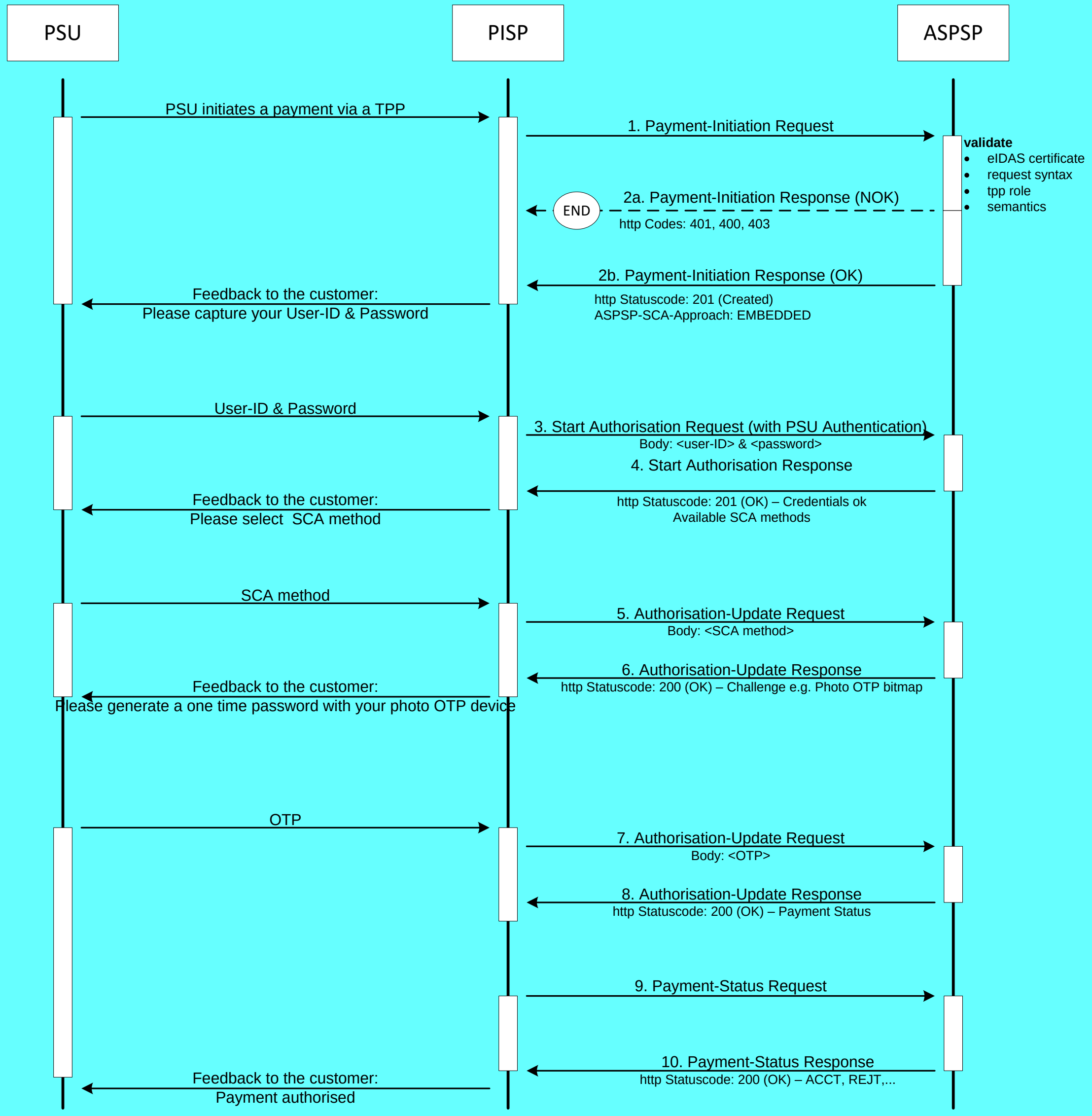
3.1.9 Embedded SCA Approach with only one SCA method available

In case where only one SCA method is available, the "Authorise Transaction Request" is added to the flow, where the TPP is transmitting the authentication data of the customer, e.g. an OTP with included dynamic linking to the transaction details.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: EMBEDDED
Feedback to the customer:
Please present your User-ID & Password
User-ID & Password
3. Start Authorisation Request (with PSU Authentication)
Body: <user-ID> & <password>
4. Start Authorisation Response
http Statuscode: 201 (Created) – Credentials ok
Challenge e.g. Photo OTP bitmap
Feedback to the customer:
Please generate the OTP with your Photo OTP device
OTP
5. Authorisation-Update Request
Body: <OTP>
6. Authorisation-Update Response
http Statuscode: 200 (OK) – Payment Status
7. Payment-Status Request
8. Payment-Status Response
http Statuscode: 200 (OK) – ACCT, REJT,...
Feedback to the customer:
Payment authorised

3.1.10 Embedded SCA Approach with Selection of an SCA method

In the following flow, there is a selection of an SCA method added in case of the ASPSP supporting several SCA methods for the corresponding PSU. The ASPSP transmits first the available methods to the PISP. The PISP might filter them, if not all authentication methods can be technically supported. The available methods then are presented to the PSU for choice.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
Feedback to the customer:
Please capture your User-ID & Password
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: EMBEDDED
User-ID & Password
3. Start Authorisation Request (with PSU Authentication)
Body: <user-ID> & <password>
4. Start Authorisation Response
Feedback to the customer:
Please select SCA method
http Statuscode: 201 (OK) – Credentials ok
Available SCA methods
SCA method
5. Authorisation-Update Request
Body: <SCA method>
6. Authorisation-Update Response
Feedback to the customer:
http Statuscode: 200 (OK) – Challenge e.g. Photo OTP bitmap
Please generate a one time password with your photo OTP device
OTP
7. Authorisation-Update Request
Body: <OTP>
8. Authorisation-Update Response
http Statuscode: 200 (OK) – Payment Status
9. Payment-Status Request
10. Payment-Status Response
Feedback to the customer:
http Statuscode: 200 (OK) – ACCT, REJT,...
Payment authorised

3.1.11 Combination of Flows due to mixed SCA Approaches

If an ASPSP supports for a PSU at least one decoupled SCA method and at the same time at least one SCA method that is not decoupled, then the above flows might be mixed as follows, since the ASPSP then needs to start the process with the assumption of one specific SCA approach to offer all available SCA methods to the PSU.

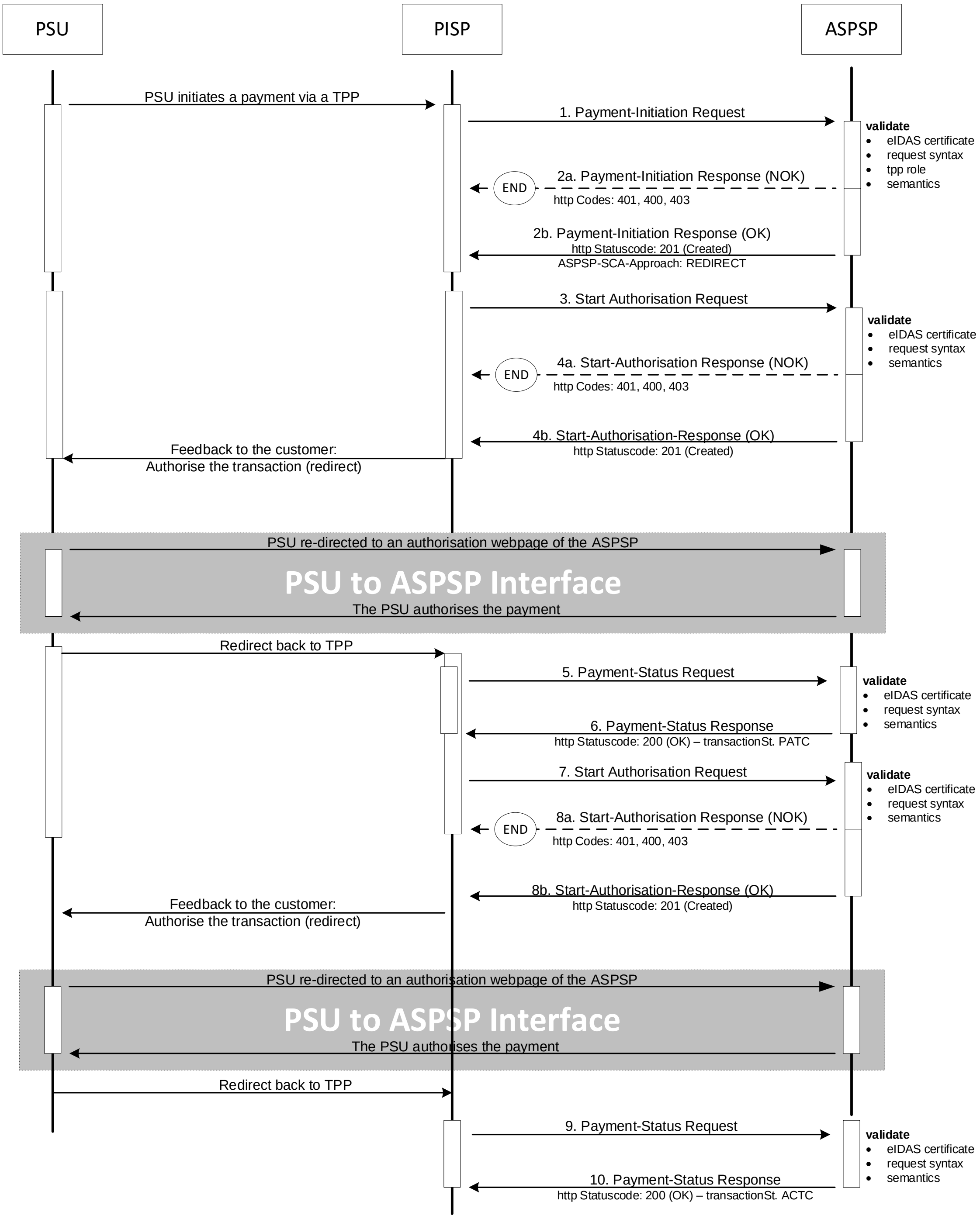
In case the ASPSP is starting the payment initiation flow with a redirect the PSU can choose on the authentication site of the ASPSP the decoupled authentication method. This is then transparent for the TPP and has no influence on the flows defined above.

In case the ASPSP is starting the payment initiation flow with the Embedded SCA Approach the ASPSP will provide a list of available SCA methods to the PSU via the TPP. If the PSU chooses an authentication method which requires the Decoupled SCA Approach, then the ASPSP is branching into the transaction flow for the Decoupled Approach as shown above: The ASPSP will return the corresponding HTTP header ASPSP-SCA-Approach with value "DECOUPLED" and the current status of the payment initiation, e.g. "ACTC" for correct technical checks but will return no hyperlink for further action other than the "self" and "status" hyperlink. The next request of the TPP then needs to be the GET Status Request to get the final status of the transaction after having processed the SCA method.

In case the ASPSP needs to decide between the Decoupled and the Redirect SCA approach, the ASPSP also might first offer the SCA methods available to the PSU and then branch after the selection of the PSU into the Decoupled or Redirect SCA Approach.

3.1.12 Multilevel SCA Approach: Example for the Redirect SCA Approach

The multilevel SCA Approach supports the authorisation of a payment by several users, e.g. in a 4 eyes principle authorisation. Multilevel SCA are always handled with Explicit start of the several Authorisation Mechanisms. In the following the flow for a 4 eyes principle authorisation is shown, where both SCA are performed by redirect.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
3. Start Authorisation Request
validate
• eIDAS certificate
• request syntax
• semantics
END
4a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
4b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
Feedback to the customer:
Authorise the transaction (redirect)
PSU re-directed to an authorisation webpage of the ASPSP
PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
5. Payment-Status Request
validate
• eIDAS certificate
• request syntax
• semantics
6. Payment-Status Response
http Statuscode: 200 (OK) – transactionSt. PATC
7. Start Authorisation Request
validate
• eIDAS certificate
• request syntax
• semantics
END
8a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
8b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
Feedback to the customer:
Authorise the transaction (redirect)
PSU re-directed to an authorisation webpage of the ASPSP
PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
9. Payment-Status Request
validate
• eIDAS certificate
• request syntax
• semantics
10. Payment-Status Response
http Statuscode: 200 (OK) – transactionSt. ACTC

Remark: This flow is not depending on the SCA Approach. Multilevel SCA transactions are performed by using n times the Start Authorisation Request for n times SCA, where the corresponding SCA flow is replacing the Redirect SCA flow above. These SCA processes could also be performed simultaneously.

3.2 Data Overview Payment Initiation Service

The following table defines the technical description of the abstract data model as defined in [oFA-OR-Com]or the Payment Initiation service. The columns give an overview on the API protocols as follows:

- The "Data element" column is using the abstract data elements following [oFA-OR-Com] to deliver the connection to rules and role definitions in this document.

- The "Attribute encoding" is giving the actual encoding definition within the XS2A API as defined in this document.

- The "Location" columns define, where the corresponding data elements are transported as HTTP parameters on path, header or body level, resp. are taken from eIDAS certificates. For HTTP Parameters, the "path" subsumes host, port and API basepath.

  Remark: Please note that website authentication certificate related data elements are not elements of the actual API call. They are indicated here, since they are mandated in the backend processing and might be transported from the API endpoint internally to the backend on the application layer. Please note, that in difference to this, the certificate data for the electronic seal can be transported within a dedicated HTTP header field.

- The "Usage" column gives an overview on the usage of data elements in the different services and API Calls. Within [oFA-OR-Com], the XS2A calls are described as abstract API calls. These calls will be technically realised as HTTPS POST, PUT and GET commands. The calls are divided into the following calls for Payment Initiation:

  - The Initiation Request which shall be the first API Call for every transaction within the corresponding XS2A service Payment Initiation. This call generates the corresponding resource within the Payment Initiation Service. The Payment Initiation can address a single payment, bulk payments and recurring payments. The latter are implemented as an initiation of a standing order.

  - The Update Data Call is a call, where the TPP needs to add PSU related data, which is requested in the return of the first call. This call might be repeated.

  - The Authorisation Request is only used in an Embedded SCA Approach to authorise the transaction in case a second factor authentication is needed.

  - The Status Request is used e.g. in cases, where the SCA control is taken over by the ASPSP and the TPP needs later information about the outcome.

The following usage of abbreviations in the Location and Usage columns is defined, cp. also [oFA-OR-Com] for details.

- x: This data element is transported on the corresponding level.
- m: Mandatory
- o: Optional for the TPP to use
- c: Conditional. The condition is described in the addressed API Calls, condition defined by the ASPSP

The following table does not only define requirements on request messages but also requirements on data elements for the response messages. These requirements only apply to positive responses (i.e. HTTP response code 2xx). The specification of negative responses and error related communications is service independent and is thus provided already on framework level in [oFA PFSM]. For example, in the case of the Payment Initiation Response Message with HTTP response code 4xx or 5xx, no payment initiation resource has been created and therefore no resource related information can be returned.

Remark: The more technical functions like GET …/{paymentId} and GET …/{authorisationId} and the Cancellation Request are not covered by this table.

Data element	**Attribute encoding**	**Location**					**Usage**					
		Path	Query P.	Header	Body	Certificate[2]	Init Req.	Init Resp.	Canc. Req.	Canc.Resp	Stat. Req.	Stat. Resp
TPP Registration Number						x	m		m		m	
TPP Name						x	m		m		m	
TPP Roles						x	m		m		m	
TPP National Competent Authority						x	m		m		m	
Request Identification	X-Request-ID			x			m	m	m	m	m	m

[2] This refers to the certificate for website authentication.

Data element	**Attribute encoding**	**Location**					**Usage**					
		Path	Query P.	Header	Body	Certificate[2]	Init Req.	Init Resp.	Canc. Req.	Canc.Resp	Stat. Req.	Stat. Resp
Resource ID (received from the ASPSP server)	paymentId				x			m				
Resource ID[3] (used in follow up processes by the API Client)		x							m		m	
Transaction Fees	transactionFees				x			o				
Transaction Fee Indicator	transactionFeeIndicator				x			o				
Access Token (from optional OAuth2)	Authorization			x			c		c		c	
Further signature related data	Digest			x			c		c		c	
API Client Electronic Signature	x-jws-signature			x			c		c		c	
Transaction Status	transactionStatus				x			m		m		m
Funds Availability Flag	fundsAvailable				x							c
PSU Message Information	psuMessage				x			o		o		o
API Client Message Information	apiClientMessages				x			o		o		o
PSU Identification	PSU-ID			x			c		o			
PSU Identification Type	PSU-ID-Type			x			c		o			

[3] Please note that the Resource ID is transported in the path after the generation of the payment initiation resource. This is then a path parameter without an explicit encoding of the attribute name.

Data element	Attribute encoding	Location					Usage					
		Path	Query P.	Header	Body	Certificate[2]	Init Req.	Init Resp.	Canc. Req.	Canc.Resp	Stat. Req.	Stat. Resp
Corporate Identification	PSU-Corporate-ID			x			c		o		c	
Corporate ID Type	PSU-Corporate-ID-Type			x			c		o		c	
Available SCA Methods	scaMethods				x			c		c		
SCA Challenge Data	challengeData				x			c		c		
IP Address PSU	PSU-IP-Address			x			m		o		o	
IP Port PSU	PSU-IP-Port			x			o		o		o	
PSU User Agent	PSU-User-Agent[4]			x			o		o		o	
GEO Information	PSU-Geo-Location			x			o		o		o	
Redirect URL ASPSP	_links.scaRedirect				x			c		c		
ASPSP-SCA-Approach	ASPSP-SCA-Approach			x				c		c		
Further PSU related Information	PSU-Accept			x			o		o		o	
	PSU-Accept-Charset			x			o		o		o	
	PSU-Accept-Encoding			x			o		o		o	
	PSU-Accept-Language			x			o		o		o	
	PSU-Http-Method			x			o		o		o	
	PSU-Device-ID			x			o		o		o	
SCA Approach Preference	Client-SCA-Approach-Preference			x			o		o			

[4] This field transports key information for risk management like browser type or PSU device operating system. The forwarding of further HTTP header fields might be supported in future versions of the specification to transport other device related information.

Data element	Attribute encoding	Location					Usage					
		Path	Query P.	Header	Body	Certificate[2]	Init Req.	Init Resp.	Canc. Req.	Canc.Resp	Stat. Req.	Stat. Resp
Redirect URI TPP[5]	Client-Redirect-URI			x			c		o			
	Client-Nok-Redirect-URI			x			o		o			
Authorisation Preference	Client-Explicit-Authorisation-Preferred			x			o		o			
Rejection Preference	Client-Rejection-NoFunds-Preferred			x			o					
Client Notification URI	Client-Notification-URI			x			o		o			
Client Notfication Content Preference	Client-Notification-Content-Preferred			x			o		o			
Client Brand Information	Client-Brand-Logging-Information			x			o		o			
Payment Product	payment-product	x					m					

The XS2A Interface calls which represent the messages defined in [oFA-OR-Com] will be defined in the following sections.

3.3 Payment Initiation Specifics for Multi-currency Accounts

The payment data contained in the request body can also address sub-accounts which are provided in specific currencies, cp. definition of multi-currency accounts in Section 2.2. This is independent of the coding in JSON or XML.

3.4 Payment Initiation Request

3.4.1 Payment Initiation with JSON encoding of the Payment Instruction

Call

```
POST /v2/payments/{payment-product}
```

[5] This redirect link must be contained, if the Client-Redirect-Preferred flag is contained and equals "true" or if the "Client-Redirect-Preferred" flag is not used.

Creates a payment initiation request at the ASPSP.

Path Parameters

Attribute	Type	Description
payment-product	String	The addressed payment product endpoint, e.g. for SEPA Credit Transfers (SCT). The default list of products supported in this standard is: • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers The ASPSP will publish which of the payment products/endpoints will be supported. For definitions of basic non euro generic products see [oFA DomDef]. Further products might be published by the ASPSP within its XS2A documentation. These new product types will end in further endpoints of the XS2A Interface.

Query Parameters

No Query Parameter

Request Header

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	application/json
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party. This is the unique ID of TPP for the payment initiation regarding PSD2 article 47 and EBA RTS article 29.
Authorization	String	Conditional	Bearer Token. Is contained only, if an OAuth2 based authentication was performed in a pre-step

Attribute	Type	Condition	Description
			or an OAuth2 based SCA was performed in a preceding AIS service in the same session.
Consent-ID	String	Optional	This data element may be contained, if the payment initiation transaction is part of a session, i.e. combined AIS/PIS service. This then contains the "consentId" of the related AIS consent, which was performed prior to this payment initiation. The access attribute in this case needs to contain the "paymentInitiations" access right for the related account.
PSU-IP-Address	String	Mandatory	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. If not available, the TPP shall use the IP Address used by the TPP when submitting this request.
Client-Rejection-NoFunds-Preferred	Boolean	Optional	If it equals "true" then the TPP prefers a rejection of the payment initiation in case the ASPSP is providing an integrated confirmation of funds request an the result of this is that not sufficient funds are available. If it equals "false" then the TPP prefers that the ASPSP is dealing with the payment initiation like in the ASPSPs online channel, potentially waiting for a certain time period for funds to arrive to initiate the payment. This parameter may be ignored by the ASPSP.

Remark: Note that a reference of the payment to payer/payee following [PSD2], Article 46 (b), will be handled on application layer with the data attributes related to end2end identification and remittance information.

NOTE: The data overview in Section 3.2 defines more request headers, like for PSU identification, PSU environment information, SCA Approach or explicit authorisation preferences of the TPP as well as redirect information. These parameters are described in detail in [oFA PFSM] or in the related Open API files. These parameters apply to all Transaction Initiation Requests, i.e. transactions, which need an authorisation within the openFinance API Framework and are left out here to achieve a better readablilty.

Request Body

The payment data to be transported in the request body is dependent on the chosen API endpoint. Some standard definitions related to the above mentioned standard products defined in [oFA PFSM] and domestic formats are defined in [oFA DomDef]. Further definitions might be given community or ASPSP specific. ASPSP or community definitions shall reuse standard attribute names.

Response Code

The HTTP response code equals 201.

Response Header

Attribute	Type	Condition	Description
Location	String	Mandatory	Location of the created resource (if created)
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	Mandatory	Payment transaction status, cp. Section 2.3.1.
paymentId	Max70Text	Mandatory	resource identification of the generated payment initiation resource.
transactionFees	Amount	Optional	Might be used by the ASPSP to transport the total transaction fee relevant for the underlying payments. This field includes the entry of the currencyConversionFees if applicable.
currency Conversion Fee	Amount	Optional	Might be used by the ASPSP to transport specific currency conversion fees related to the initiated credit transfer.
estimatedTotal Amount	Amount	Optional	The amount which is estimated to be debted from the debtor account.

Attribute	Type	Condition	Description
			Note: This amount includes fees.
estimated Interbank Settlement Amount	Amount	Optional	The estimated amount to be transferred to the payee.
transactionFee Indicator	Boolean	Optional	If equals true, the transaction will involve specific transaction cost as shown by the ASPSP in their public price list or as agreed between ASPSP and PSU. If equals false, the transaction will not involve additional specific transaction costs to the PSU unless the fee amount is given specifically in the data elements transactionFees and/or currencyConversionFees. If this data element is not used, there is no information about transaction fees unless the fee amount is given explicitly in the data element transactionFees and/or currencyConversionFees.
scaMethods	Array of Authentication Objects	Conditional	This data element might be contained, if SCA is required and if the PSU has a choice between different authentication methods. Depending on the risk management of the ASPSP this choice might be offered before or after the PSU has been identified with the first relevant factor, or if an access token is transported. If this data element is contained, then there is also a hyperlink of type "startAuthorisationWith AuthenticationMethodSelection" contained in the response body. These methods shall be presented towards the PSU for selection by the TPP.
chosenSca Method	Authentication Object	Conditional	This data element is only contained in the response if the ASPSP has chosen the Embedded SCA Approach, if the PSU is already identified e.g. with the first relevant factor or alternatively an

Attribute	Type	Condition	Description
			access token, if SCA is required and if the authentication method is implicitly selected.
challengeData	Challenge	Conditional	It is contained in addition to the data element "chosenScaMethod" if challenge data is needed for SCA. In rare cases this attribute is also used in the context of the "startAuthorisationWithPsuAuthentication" or "startAuthorisactionWithEncryptedPsuAuthentication" link.
_links	Links	Mandatory	A list of hyperlinks to be recognised by the API Client. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. The potential links for this response message are generically defined in [oFA PFSM] for all Transaction Initiation Response messages. These links will also be contained in the related Open API files. **Remark**: All links can be relative or full links, to be decided by the ASPSP.
psuMessage	Max500Text	Optional	Text to be displayed to the PSU
apiClientMessages	Array of Client Message Information	Optional	Messages to the API Client on operational issues.

Example

Request

```
POST https://api.testbank.com/psd2/v2/payments/sepa-credit-transfers
Content-Type:            application/json
X-Request-ID:            99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-IP-Address:          192.168.8.78
PSU-GEO-Location:        GEO:52.506931;13.144558
```

```
PSU-User-Agent:         Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
Date:                   Sun, 06 Aug 2017 15:02:37 GMT

{
   "debtorAccount": {"iban": "DE40100100103307118608"},
   "instructedAmount": {"currency": "EUR", "amount": "123.50"},
   "creditorAccount": {"iban": "DE02100100109307118603"},
   "creditor": {"name": "Merchant123"},
   "remittanceInformationUnstructured": ["Ref Number Merchant"]
}
```

Response in case of a redirect with an implicitly created authorisation sub-resource

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     REDIRECT
Date:                   Sun, 06 Aug 2017 15:02:42 GMT
Location:               https://www.testbank.com/psd2/v2/payments/sepa-
credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "3d9a81b3-a47d-4130-8765-a9c0ff861100",
  "_links": {
        "scaRedirect": {"href": "https://www.testbank.com/asdfasdfasdf"},
        "self": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-
a47d-4130-8765-a9c0ff861100"},
        "status": {"href": "/psd2/v2/payments/sepa-credit-
transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100/status"},
        "scaStatus": {"href": "/psd2/v2/payments/sepa-credit-
transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100/authorisations/3d9a81b3-
a47d-4130-9999-a9c0ff861100"}
  }
}
```

Same example in case where an explicit authorisation start is needed

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     REDIRECT
Date:                   Sun, 06 Aug 2017 15:02:42 GMT
Location:               https://www.testbank.com/psd2/v2/payments/sepa-
credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100
```

```
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "3d9a81b3-a47d-4130-8765-a9c0ff861100",
  "_links": {
        "self": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100"},
        "status": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100/status"},
        "startAuthorisation": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100/authorisations"}
  }
}
```

Response in case of an OAuth2 SCA approach with implicitly creating an authorisation sub-resource

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     REDIRECT
Date:                   Sun, 06 Aug 2017 15:02:42 GMT
Location:               https://www.testbank.com/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "3d9a81b3-a47d-4130-8765-a9c0ff861100",
  "_links": {
        "scaOAuth": {"href": "https://www.testbank.com/oauth/.well-known/oauth-authorization-server"},
        "self": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100"},
        "status": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100/status"},
        "scaStatus": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100/authorisations/3d9a81b3-a47d-4130-9999-a9c0ff861100"}
  }
}
```

Response in case of the decoupled approach with explicit start of authorisation needed (will be done with the update PSU identification function)

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     DECOUPLED
Date:                   Sun, 06 Aug 2017 15:03:47 GMT
Location:               https://www.testbank.com/psd2/v2/payments/sepa-
credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "3d9a81b3-a47d-4130-8765-a9c0ff861100",
  "_links": {
     "startAuthorisationWithPsuIdentification": {"href":
"/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-
a9c0ff861100/authorisations"},
     "self": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-
a47d-4130-8765-a9c0ff861100"}
    }
}
```

Response in case of the signature approach with explicit start of authorisation needed (will be done with the update PSU identification function)

```
HTTP/1.x 201 Created
X-Request-ID:                 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:           SIGNATURE
Body-Sig-Profile-Accepted:    JAdES_JS
Date:                         Sun, 06 Aug 2017 15:03:47 GMT
Location:               https://www.testbank.com/psd2/v2/payments/sepa-
credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100
Content-Type:                 application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "3d9a81b3-a47d-4130-8765-a9c0ff861100",
  "_links": {
     "startAuthorisationWithPsuIdentification": {"href":
"/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-
a9c0ff861100/authorisations"},
     "self": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-
a47d-4130-8765-a9c0ff861100"}
```

```
    }
}
```

Response in case of the embedded approach with explicit start of authorisation

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     EMBEDDED
Date:                   Sun, 06 Aug 2017 15:03:47 GMT
Location:               https://www.testbank.com/psd2/v2/payments/sepa-
credit-transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100
Content-Type:           application/json

{
    "transactionStatus": "RCVD",
    "paymentId": "3d9a81b3-a47d-4130-8765-a9c0ff861100",
    "_links": {
        "startAuthorisationWithPsuAuthentication": {"href":
"/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-a47d-4130-8765-
a9c0ff861100/authorisations"},
         "self": {"href": "/psd2/v2/payments/sepa-credit-
transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100"}
               }
}
```

3.4.2 Payment Initiation with pain.001 XML message as Payment Instruction

Call

```
POST /v2/payments/{payment-product}
```

Creates a payment initiation request at the ASPSP.

Remark: The underlying pain.001 structure which is transported in the content body of this request may only contain one payment. In cases of the initiation of bulk payments, the endpoint defined in Section 3.4.3.2 shall be used.

Path Parameters

Attribute	Type	Description
payment-product	String	The addressed payment product, e.g. SCT. The default list of products supported in this standard is: • pain.001-sepa-credit-transfers • pain.001-instant-sepa-credit-transfers • pain.001-target-2-payments • pain.001-cross-border-credit-transfers Further products might be published by the ASPSP within its XS2A documentation. **Remark**: For all SEPA Credit Transfer based endpoints which accept XML encoding, the XML pain.001 schemes provided by EPC[6] are supported by the ASPSP as a minimum for the body content. Further XML schemes might be supported by some communities. **Remark**: For cross-border and target-2 payments only community wide pain.001 schemes do exist, cp. [oFA PFSM].

Query Parameters

The same query parameter definition as in Section 3.4.1 applies.

[6] Compare https://www.europeanpaymentscouncil.eu/what for all related payment initiation schema definitions.

Request Header

The same header as in Section 3.4.1, only the content type indicates XML encoding ("application/xml").

Request Body

A pain.001 structure corresponding to the chosen payment product, see above on XML schema support.

Response

The same response as in Section 3.4.1.

Example

Request

```
POST https://api.testbank.com/psd2/v2/payments/pain.001-sepa-credit-
transfers
Content-Type:            application/xml
X-Request-ID:            "123e4567-e89b-12d3-a456-426655440000"
PSU-IP-Address:          "192.168.8.78"
PSU-User-Agent:          "Chrome_v12"

<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <CstmrCdtTrfInitn>
    <GrpHdr>
      <MsgId>MIPI-123456789RI-123456789</MsgId>
      <CreDtTm>2017-02-14T20:23:34.000Z</CreDtTm>
      <NbOfTxs>1</NbOfTxs>
      <CtrlSum>123</CtrlSum>
      <InitgPty>
        <Nm>PaymentInitiator</Nm>
        <Id><OrgId><Othr><Id>DE10000000012</Id>
          <SchmeNm><Prptry>PISP</Prptry></SchmeNm></Othr></OrgId></Id>
      </InitgPty>
    </GrpHdr>
    <PmtInf>
      <PmtInfId>BIPI-123456789RI-123456789</PmtInfId>
      <PmtMtd>TRF</PmtMtd>
      <NbOfTxs>1</NbOfTxs>
      <CtrlSum>123</CtrlSum>
      <PmtTpInf><SvcLvl><Cd>SEPA</Cd></SvcLvl></PmtTpInf>
      <ReqdExctnDt>2017-02-15</ReqdExctnDt>
      <Dbtr><Nm>PSU Name</Nm></Dbtr>
      <DbtrAcct><Id><IBAN>DE87200500001234567890</IBAN></Id></DbtrAcct>
      <ChrgBr>SLEV</ChrgBr>
      <CdtTrfTxInf>
        <PmtId><EndToEndId>RI-123456789</EndToEndId></PmtId>
        <Amt><InstdAmt Ccy="EUR">123</InstdAmt></Amt>
        <Cdtr><Nm>Merchant123</Nm></Cdtr>
        <CdtrAcct><Id><IBAN> DE23100120020123456789</IBAN></Id></CdtrAcct>
        <RmtInf><Ustrd>Ref Number Merchant-123456</Ustrd></RmtInf>
      </CdtTrfTxInf>
    </PmtInf>
  </CstmrCdtTrfInitn>
</Document>
```

Response

See the example responses in JSON encoding in Section 3.4.1

3.4.3 Payment Initiation for Bulk Payments

This function supports the upload of bulk payments. This function is an **optional** function of the ASPSP in the XS2A interface. It can be offered by the ASPSP in JSON or XML modelling of the payment data, i.e. the body content.

3.4.3.1 Bulk Payment Initiation with JSON encoding of the Payment Instruction

Call

```
POST /v2/bulk-payments/{payment-product}
```

Creates a bulk payment initiation request at the ASPSP.

Path Parameters

Attribute	Type	Description
payment-product	String	The addressed payment product endpoint for bulk payments e.g. for a bulk SEPA Credit Transfers (SCT). These endpoints are optional. Some default names are: • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers The ASPSP will publish which of the payment products/endpoints will be supported.

Attribute	Type	Description
		For definitions of basic non euro generic products see [oFA DomDef].. Further products might be published by the ASPSP within its XS2A documentation. These new product types will end in further endpoints of the XS2A Interface.

Query Parameters

The same query parameter definition as in Section 3.4.1 applies.

Request Headers

The same HTTP header definition as in Section 3.4.1 applies.

Request Body

The body definition with the JSON based SEPA bulk payments and other bulk payment formats are contained in [oFA PFSM].

Response

The responses definition is analogous to the initiation of single payments, cp. Section 3.4.1.

Remark: Please note that the optional batchBookingPreferred flag shall be ignored by the ASPSP if batch booking is not supported.

Recommendations:

- If the bulk entry contains transactions without a reference, which are rejected, then the whole bulk should be rejected.
- ASPSPs might make the usage of instructionIds or UETR mandatory for bulk initiations.

3.4.3.2 Bulk Payment Initiation with XML encoding of the Payment Instruction

Call

```
POST /v2/bulk-payments/{payment-product}
```

Creates a bulk payment initiation request at the ASPSP.

Path Parameters

Attribute	Type	Description
payment-product	String	The addressed payment product endpoint for bulk payments e.g. for a bulk SEPA Credit Transfers (SCT). These endpoints are optional. Some default names are: • pain.001-sepa-credit-transfers • pain.001-instant-sepa-credit-transfers • pain.001-proprietary-credit-transfers The ASPSP will publish which of the payment products/endpoints will be supported. **Remark**: For all SEPA Credit Transfer based endpoints which accept XML encoding, the XML pain.001 schemes provided by EPC[7] are supported by the ASPSP as a minimum for the body content. Further XML schemes might be supported by some communities. **Remark**: Payment Initiations might be further restricted by the ASPSP on size or on multiplicity of entries. This could be e.g. a restriction on the usage of one ordering party or/and one debtor account. **Remark**: For proprietary payments, only community wide pain.001 schemes do exist, cp [oFA PFSM].

Query Parameters

The same query parameter definition as in Section 3.4.2 applies.

Request Headers

The same HTTP header definition as in Section 3.4.2 applies

[7] Compare https://www.europeanpaymentscouncil.eu/what-we-do for all related payment initiation schema definitions.

Request Body

A pain.001 structure corresponding to the chosen payment product, see above on XML schema support.

Response

The responses definition is analogous to the initiation of single XML based payments, cp Section 3.4.2.

3.4.4 Initiation for Standing Orders for Recurring/Periodic Payments

The recurring payments initiation function will be covered in this specification as a specific standing order initiation: The TPP can submit a recurring payment initiation where the starting date, frequency and conditionally an end date is provided. Once authorised by the PSU, the payment then will be executed by the ASPSP, if possible, following this "standing order" as submitted by the TPP. No further TPP action is needed. This payment is called a periodic payment in this context to differentiate the payment from recurring payment types, where third parties are initiating the same amount of money e.g. payees for using credit card transactions or direct debits for recurring payments of goods or services. These latter types of payment initiations are not part of this interface.

3.4.4.1 Standing Orders for Recurring/Periodic Payments in JSON encoding

Call

```
POST /v2/periodic-payments/{payment-product}
```

Path Parameters

The same path parameter to determine the underlying payment type of the recurring payment as in Section 3.4.1 applies.

Query Parameters

The same query parameter definition as in Section 3.4.1 applies.

Request Header

For this initiation the same header as in Section 3.4.1 is used.

Request Body

First, any tag of the underlying payment product as defined in [oFA PFSM] can be used. In addition the following tags are used:

Tag	Type	Usage	Description
startDate	ISODate	Mandatory	The first applicable day of execution starting from this date is the first payment.
executionRule	String	Optional	"following" or "preceding" supported as values. This data attribute defines the behavior when recurring payment dates falls on a weekend or bank holiday. The payment is then executed either the "preceding" or "following" working day. ASPSP might reject the request due to the communicated value, if rules in Online-Banking are not supporting this execution rule.
endDate	ISODate	Optional	The last applicable day of execution If not given, it is an infinite standing order.
frequency	Frequency Code	Mandatory	The frequency of the recurring payment resulting from this standing order.
dayOfExecution	Max2Text	Conditional	"31" is ultimo. The format is following the regular expression \d{1,2}. Example: The first day is addressed by "1". The date is referring to the time zone of the ASPSP.
monthsOfExecution	Array of Max2Text	Conditional	The format is following the regular expression \d{1,2}. The array is restricted to 11 entries. The values contained In the array entries shall all be different and the maximum value of one entry is 12. This attribute is contained if and only if the frequency equals "MonthlyVariable".

Tag	Type	Usage	Description
			Example: An execution on January, April and October each year is addressed by ["1". "4", "10"].

Response

The formats of the Payment Initiation Response resp. the subsequent transaction authorisation process for standing orders with JSON based payment data equals the corresponding Payment Initiation Response resp. the subsequent transaction authorisation process for a single payment containing JSON based payment data.

> **Remark**: Please note that for the payment initiation of standing orders, the ASPSP will always mandate an SCA with dynamic linking, exemptions are not permitted.

Example

Request for Variant 1 with full JSON encoding

```
POST https://www.testbank.com/psd2/v2/periodic-payments/sepa-credit-
transfers
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-IP-Address:         192.168.8.78
PSU-User-Agent:         Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
Date:                   Sun, 06 Aug 2017 15:02:37 GMT
{

 "debtorAccount": {"iban": "DE40100100103307118608"},
 "instructedAmount": {"currency": "EUR", "amount": "123"},
 "creditor": {"name": "Merchant123"},
 "creditorAccount": {"iban": "DE23100120020123456789"},
 "remittanceInformationUnstructured": ["Ref Number Abonnement"],
 "startDate": "2018-03-01",
 "executionRule": "preceding",
 "frequency": "Monthly",
 "dayOfExecution": "1"
}
```

3.4.4.2 Payment Initiation for Standing Orders with XML based payment data

The standing order management data will be JSON based in the XS2A API also if the related payment data is based on XML syntax. For this reason, the Payment Initiation Request for standing orders is defined as an HTTP multipart message in this case.

Call

```
POST /v2/periodic-payments/{payment-product}
```

Path Parameters

The same path parameter to determine the underlying payment type of the recurring payment as in Section 3.4.2 applies.

Query Parameters

The same query parameter and HTTP header definition as in Section 3.4.1 applies.

Request Header

The same header definitions as in Section 3.4.1 are used with the exception of the Content-Type Header. Here the following requirement applies:

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	multipart/form-data; recommended boundary=AaaBbbCcc

Remark: The ASPSP shall also accept other boundaries chosen by the API Client.

Request Body, Part 1

The first part of the body contains first a sub-header section as defined by the following table:

Attribute	Type	Condition	Description
Content-Disposition	String	Mandatory	form-data; name="xml_sct"
Content-Type	String	Mandatory	application/xml

The first part content of the body is defined as for the Payment Initiation Request for a single request in an XML (pain.001) based format, cp. Section 3.4.2.

Request Body, Part 2

The second part of the body contains first a sub-header section as defined by the following table:

Attribute	Type	Condition	Description
Content-Disposition	String	Mandatory	form-data; name="json_standingorderType"
Content-Type	String	Mandatory	application/json

The second part content of the body is defined as follows:

Tag	Type	Usage	Description
startDate	ISODate	Mandatory	The first applicable day of execution starting from this date is the first payment.
executionRule	String	Optional	"following" or "preceding" supported as values. This data attribute defines the behavior when recurring payment dates falls on a weekend or bank holiday. The payment is then executed either the "preceding" or "following" working day.ASPSP might reject the request due to the communicated value, if rules in Online-Banking are not supporting this execution rule.
endDate	ISODate	Optional	The last applicable day of execution If not given, it is an infinite standing order.
frequency	Frequency Code	Mandatory	Frequency of the recurring payment resulting from this standing order.
dayOfExecution	Max2Text	Conditional	"31" is ultimo
monthsOfExecution	Array of Max2Text	Conditional	The format is following the regular expression \d{1,2}. The array is restricted to 11 entries. The values contained In the array entries shall all be different and the maximum value of one entry is 12.

Tag	Type	Usage	Description
			This attribute is contained if and only if the frequency equals "MonthlyVariable". Example: An execution on January, April and October each year is addressed by ["1". "4", "10"].

Response

The formats of the Payment Initiation Response resp. the subsequent transaction authorisation process for standing orders with XML based payment data equals the corresponding Payment Initiation Response resp. the subsequent transaction authorisation process for a single payment containing XML based payment data.

Example

Request with JSON Order Execution Information and XML Payment Information

```
POST https://www.testbank.com/psd2/v2/periodic-payments/sepa-credit-
transfers
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-IP-Address:         192.168.8.78
PSU-GEO-Location:       GEO:52.506931;13.144558
PSU-User-Agent:         Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
Date:                   Sun, 06 Aug 2017 15:02:37 GMT
Content-Type: multipart/form-data; boundary=AaaBbbCcc
--AaaBbbCcc
Content-Disposition: form-data; name="xml_sct"
Content-Type: application/xml
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <CstmrCdtTrfInitn>
    <GrpHdr>
      <MsgId>MIPI-123456789RI-123456789</MsgId>
      <CreDtTm>2017-02-14T20:23:34.000Z</CreDtTm>
      <NbOfTxs>1</NbOfTxs>
      <CtrlSum>123</CtrlSum>
      <InitgPty>
        <Nm>PaymentInitiator</Nm>
        <Id><OrgId><Othr><Id>DE10000000012</Id>
              <SchmeNm><Prptry>PISP</Prptry></SchmeNm></Othr></OrgId></Id>
      </InitgPty>
    </GrpHdr>
    <PmtInf>
      <PmtInfId>BIPI-123456789RI-123456789</PmtInfId>
      <PmtMtd>TRF</PmtMtd>
      <NbOfTxs>1</NbOfTxs>
      <CtrlSum>123</CtrlSum>
      <PmtTpInf><SvcLvl><Cd>SEPA</Cd></SvcLvl></PmtTpInf>
      <ReqdExctnDt>2017-02-15</ReqdExctnDt>
      <Dbtr><Nm>PSU Name</Nm></Dbtr>
      <DbtrAcct><Id><IBAN>DE87200500001234567890</IBAN></Id></DbtrAcct>
      <ChrgBr>SLEV</ChrgBr>
      <CdtTrfTxInf>
        <PmtId><EndToEndId>RI-123456789</EndToEndId></PmtId>
        <Amt><InstdAmt Ccy="EUR">123</InstdAmt></Amt>
        <Cdtr><Nm>Merchant123</Nm></Cdtr>
        <CdtrAcct><Id><IBAN>DE23100120020123456789</IBAN></Id></CdtrAcct>
        <RmtInf><Ustrd>Ref Number Merchant-123456</Ustrd></RmtInf>
      </CdtTrfTxInf>
    </PmtInf>
  </CstmrCdtTrfInitn>
</Document>
--AaaBbbCcc
Content-Disposition: form-data; name="json_standingordermanagement"
Content-Type: application/json
{"startDate": "2018-03-01",
 "frequency": "Monthly",
 "executionRule": "preceding",
 "dayOfExecution": "01"
}
--AaaBbbCcc—-
```

3.5 Get Transaction Status Request

Call

`GET /v2/{payment-service}/{payment-product}/{paymentId}/status`

Can check the status of a payment initiation.

Path Parameter

Attribute	Type	Description
payment-service	String	The possible values are "payments", "bulk-payments" and "periodic-payments"
payment-product	String	The payment product, under which the payment under paymentId has been initiated. It shall be checked by the ASPSP, if the payment-product is matching the payment initiation addressed by paymentId.
paymentId	String	Resource Identification of the related payment.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current PIS transaction.
Accept	String	Optional	The TPP can indicate the formats of status reports supported together with a prioritisation following the HTTP header definition. The formats supported by this specification are • xml • JSON If only one format is supported by the TPP, which is not supported by the ASPSP this can lead to a rejection of the request.

Query Parameters

No specific query parameters defined.

Request Body

No request body.

Response Code

The HTTP response code equals 200.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body in Case of JSON based endpoint

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	Mandatory	In case where the Payment Initiation Request was JSON encoded as defined in Section 3.4.1, the status is returned in this JSON based encoding. **Remark:** If the PSU does not complete a required SCA within the required timeframe the payment resource's status must be set to "RJCT". Particularly, if a multi-level-SCA is required and the number of successful SCAs during the required timeframe is insufficient, the status must also be set to "RJCT".
reasonCode	Status Reason Code	{Or - Optional	Additional information on the reason for e.g. rejecting the request
reasonProprietary	Max35Text	Or – Optional}	Proprietary additional information on the reason for e.g. rejecting the request. **Remark:** If an ISO Code is available it should be used instead of Proprietary Reasons. Further restrictions could be defined by API Access Schemes.
fundsAvailable	Boolean	Conditional	This data element is contained, if supported by the ASPSP, if a funds check has been performed and if the transactionStatus is "ACTC", "ACWC" or "ACCP".
transactionFees	Amount	Optional	Might be used by the ASPSP to transport the total transaction fee relevant in case of an

Attribute	Type	Condition	Description
			underlying payment initiation service. This field includes the entry of the currencyConversionFees if applicable.
currencyConversion Fees	Amount	Optional	Might be used by the ASPSP to transport specific currency conversion fees related to the initiated credit transfer, in case of an underlying payment initiation service.
estimatedTotalAmount	Amount	Optional	The amount which is estimated to be debted from the debtor account in case of an underlying payment initiation service. **Note:** This amount includes fees.
estimatedInterbank SettlementAmount	Amount	Optional	The estimated amount to be transferred to the payee in case of an underlying payment initiation service.
psuMessage	Max500Text	Optional	
ownerNames	Array of Account Owner	Optional	List of owner names. Should only be delivered after successful SCA. Could be restricted to the current PSU by the ASPSP.
_links	Links	Optional	Should refer to next steps if a potential situation blocking the finalisation of the payment can be resolved via the interface e.g. for re-submission of credentials.
apiClientMessages	Array of API Client Message Information	Optional	Messages to the TPP on operational issues.

NOTE: If a payment initiation was rejected for "banking" reasons, e.g. insufficient funds or limits, the "reasonCode" or "reasonProprietary" attribute will offer more information about the rejection reason. Please note, that some related proprietary message used before in version1.3.x of the NextGenPSD2 API have been removed from the standard to mandate "reasonCode" support by ASPSPs under certain conditions.

Response Body in Case of (SEPA-)XML based endpoint

If the Payment Initiation Request is encoded in XML, cp. Section 3.4.2, then the status might be returned by the ASPSP as a pain.002 structure or as JSON structure as defined above. The ASPSP can choose in this case one of the two status formats or offer both. In case of an XML format, the chosen XML schema of the Status Request is following the XML schema definitions of the original pain.001 schema.

Note: .If several pain.002 files are available for the addressed pain.001 message, several pain.002 files might be transported in the response body. The ASPSP might use standard compression methods on application level for the response message as indicated in the content encoding header. In all cases, the Content-Type header shall be adapted accordingly.

Example

Example for JSON based endpoint

Request

```
GET https://api.testbank.com/psd2/v2/payments/3d9a81b3-a47d-4130-8765-
a9c0ff861100/status
Accept:                 application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:07 GMT
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
Content-Type:           application/json

{
  "transactionStatus": "ACCP",
  "fundsAvailable": true
}
```

Example for XML based endpoint

Request

```
GET https://api.testbank.com/psd2/v2/payments/pain.001-sepa-credit-
transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100/status
Accept:                 application/xml, application/json;q=0.9
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:07 GMT
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
Content-Type:           application/xml
```

```
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03">
..<CstmrPmtStsRpt>
….<GrpHdr>
…...<MsgId>4572457256725689726906</MsgId>
......<CreDtTm>2017-02-14T20:24:56.021Z</CreDtTm>
......<DbtrAgt><FinInstnId><BIC>ABCDDEFF</BIC></FinInstnId></DbtrAgt>
......<CdtrAgt><FinInstnId><BIC>DCBADEFF</BIC></FinInstnId></CdtrAgt>
....</GrpHdr>
....<OrgnlGrpInfAndSts>
......<OrgnlMsgId>MIPI-123456789RI-123456789</OrgnlMsgId>
......<OrgnlMsgNmId>pain.001.001.03</OrgnlMsgNmId>
......<OrgnlCreDtTm>2017-02-14T20:23:34.000Z</OrgnlCreDtTm>
......<OrgnlNbOfTxs>1</OrgnlNbOfTxs>
......<OrgnlCtrlSum>123</OrgnlCtrlSum>
......<GrpSts>ACCT</GrpSts>
....</OrgnlGrpInfAndSts>
....<OrgnlPmtInfAndSts>
......<OrgnlPmtInfId>BIPI-123456789RI-123456789</OrgnlPmtInfId>
......<OrgnlNbOfTxs>1</OrgnlNbOfTxs>
......<OrgnlCtrlSum>123</OrgnlCtrlSum>
......<PmtInfSts>ACCT</PmtInfSts>
....</OrgnlPmtInfAndSts>
..</CstmrPmtStsRpt>
</Document>
```

3.6 Get Bulk Extended Status Request

Call

```
GET /v2/bulk-payments/{payment-product}/{paymentId}/extended-status
```

Can check the overall status of a bulk payment initiation together with reporting rejected transactions.

Path Parameter

Attribute	Type	Description
payment-product	String	The payment product, under which the payment under paymentId has been initiated. The standardised payment products covered for this service are - sepa-credit-transfers - instant-sepa-credit-transfers - crossborder-payments **NOTE**: This report is not supported for XML based bulk payments.
paymentId	String	Resource Identification of the related bulk payment.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current PIS transaction.

Query Parameters

No specific query parameters defined.

Request Body

No request body.

Response Code

The HTTP response code equals 200.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
groupStatus	Group Status	Mandatory	This is the bulk status as defined in the related bulk status endpoint.
reasonCode	Status Reason Code	{Or - Optional	Additional information on the reason for e.g. rejecting the request
reasonProprietary	Max35Text	Or – Optional}	Proprietary additional information on the reason for e.g. rejecting the request. Remark: If an ISO Code is available it should be used instead of Proprietary Reasons. Further restrictions could be defined by API Access Schemes.
fundsAvailable	Boolean	Conditional	This data element is contained, if supported by the ASPSP, if a funds check has been performed and if the transactionStatus is "ACTC", "ACWC" or "ACCP".
originalTransaction InformationAndStatus	Array of Original Transaction Information And Status	Conditional	Provides information on the original transactions to which the status report message refers and which have been rejected.
transactionFees	Amount	Optional	Might be used by the ASPSP to transport the total transaction fee relevant in case of an underlying payment initiation service. This field includes the entry of the

Attribute	Type	Condition	Description
			currencyConversionFees if applicable.
currencyConversion Fees	Amount	Optional	Might be used by the ASPSP to transport specific currency conversion fees related to the initiated credit transfer, in case of an underlying payment initiation service.
estimatedTotalAmount	Amount	Optional	The amount which is estimated to be debted from the debtor account in case of an underlying payment initiation service. Note: This amount includes fees.
estimatedInterbank SettlementAmount	Amount	Optional	The estimated amount to be transferred to the payee in case of an underlying payment initiation service.
psuMessage	Max500Text	Optional	
ownerNames	Array of Account Owner	Optional	List of owner names. Should only be delivered after successful SCA. Could be restricted to the current PSU by the ASPSP.
_links	Links	Optional	Should refer to next steps if a potential situation blocking the finalisation of the payment can be resolved via the interface e.g. for re-submission of credentials.
apiClientMessages	Array of API Client Message Information	Optional	Messages to the TPP on operational issues.

Example

Request

```
GET https://api.testbank.com/psd2/v2/bulk-payments/sepa-credit-transfers/
3d9a81b3-a47d-4130-8765-a9c0ff861100/extended-status
Accept:                 application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:07 GMT
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
Content-Type:           application/json

{
  "groupStatus": "PART",
  "originalTransactionInformationAndStatus": [
    {
      "originalUetr": "de2da6c9-18be-48d4-8053-867ed90a316a",
      "transactionStatus": "RJCT"
    },
    {
      "originalUetr": "de2da6c9-18be-48d4-8053-867ed90a317a",
      "transactionStatus": "RJCT"
    }
  ]
}
```

3.7 Get Payment Request

```
GET /v2/{payment-service}/{payment-product}/{paymentId}
```

Returns the content of a payment object.

Path Parameters

Attribute	Type	Description
payment-service	String	The possible values are “payments”, “bulk-payments” and “periodic-payments”

Attribute	Type	Description
payment-product	String	The payment product, under which the payment under paymentId has been initiated.
paymentId	String	ID of the corresponding payment initiation object as returned by an Payment Initiation Request

Query Parameters

No specific query parameter.

Request Headers

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current PIS transaction.

Request Body

No request body.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Code

The HTTP response code equals 200.

Response Body

The response body is dependent on the parameter {payment-service}. It contains the view of the ASPSP on the addressed payment resource.

For JSON based {payment-services}, the payment resources may contain e.g. in addition the transaction status data element.

Note: In addition, the payment resource may contain the debtorName field even if it was not provided by the TPP. This enables the ASPSP to transport the account owner name to the PISP in case where the regulatory need is provided and if not provided by other means like the List of Addressable Accounts Service or general AIS services for AISPs.

Note: According to item 40 of [EBA-OP2] the payment resource shall contain the debtorAccount after the payment has been initiated successfully, even if it was not provided by the TPP within the initial call.

For XML based {payment-services}, the pain.001 objects are returned. In case of a submitted standing order where the payment information has been submitted in a pain.001 format, the resource content is returned in a multipart message as the submission.

In all cases, the data element entries can be different from the submission entries, if the ASPSP has reformatted the content, e.g. the requested execution dates or character sets in the unstructured remittance information.

3.8 Payment Cancellation Request

Call

```
DELETE /v2/{payment-service}/{payment-product}/{paymentId}
```

It initiates the cancellation of a payment. Depending on the payment-service, the payment-product and the ASPSP's implementation, this TPP call might be sufficient to cancel a payment. If an authorisation of the payment cancellation is mandated by the ASPSP, a corresponding hyperlink will be contained in the response message. These two cases will be separated also in using different 2xx HTTP response codes.

Path Parameter

Attribute	Type	Description
payment-service	String	The possible values are “payments”, “bulk-payments” and “periodic-payments”
payment-product	String	The payment product, under which the payment under paymentId has been initiated. It shall be checked by the ASPSP, if the payment-product is matching the payment initiation addressed by paymentId.

Attribute	Type	Description
paymentId	String	Resource Identification of the related payment.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current PIS transaction.

NOTE: The data overview in Section 3.2 defines more request headers, like for PSU identification, PSU environment information, SCA Approach or explicit authorisation preferences of the TPP as well as redirect information. These parameters are described in detail in [oFA PFSM] or in the related Open API files. These parameters apply to all transactions, which (potentially) need an authorisation within the openFinance API Framework and are left out here to achieve a better readablilty.

Query Parameters

No specific query parameters defined.

Request Body

No request body.

Response Code

If the DELETE is sufficient for cancelling the payment: HTTP response code 204.

If the DELETE is not sufficient for cancelling the payment since an authorisation of the cancellation by the PSU is needed: HTTP response code 202.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

In case of HTTP code 204, no response body is used.

In case of HTTP code 202, the following body is used:

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	Mandatory	Transaction Status of the payment resource
scaMethods	Array of Authentication Objects	Conditional	This data element might be contained, if SCA is required and if the PSU has a choice between different authentication methods. Depending on the risk management of the ASPSP this choice might be offered before or after the PSU has been identified with the first relevant factor, or if an access token is transported. If this data element is contained, then there is also a hyperlink of type "startAuthorisationWith AuthenticationMethodsSelection" contained in the response body. These methods shall be presented towards the PSU for selection by the TPP.
chosenSca Method	Authentication Object	Conditional	This data element is only contained in the response if the ASPSP has chosen the Embedded SCA Approach, if the PSU is already identified e.g. with the first relevant factor or alternatively an access token, if SCA is required and if the authentication method is implicitly selected.
challengeData	Challenge	Conditional	It is contained in addition to the data element "chosenScaMethod" if challenge data is needed for SCA.
			In rare cases this attribute is also used in the context of the "startAuthorisationWith

Attribute	Type	Condition	Description
			PsuAuthentication" or "startAuthorisactionWithEncryptedPsuAuthentication" link.
_links	Links	Conditional	A list of hyperlinks to be recognised by the API Client. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. The potential links for this response message are generically defined in [oFA PFSM] for all Transaction Initiation Response messages or response message for other requests which need authorisation. These links will also be contained in the related Open API files. **Remark**: All links can be relative or full links, to be decided by the ASPSP.

NOTE: The opening of all related authorisation links in the response also allows implicit authorisation for DELETE processes in difference to earlier versions of these specifications.

Example in case the DELETE process as such is already sufficient for cancelling the payment

Request

```
DELETE https://api.testbank.com/psd2/v2/payments/sepa-credit-
transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100

Content-Type            application/json
X-Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7769
Date                    Sun, 13 Aug 2017 17:05:37 GMT
```

Response

```
HTTP/1.x 204
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7769
Date:                   Sun, 13 Aug 2017 17:05:38 GMT
```

Example in case an authorisation of the cancellation is needed by the PSU

Request

Request

```
DELETE https://api.testbank.com/psd2/v2/payments/sepa-credit-
transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100

Content-Type            application/json
X-Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7769
Date                    Sun, 13 Aug 2017 17:05:37 GMT
```

Response

```
HTTP/1.x 202
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7769
Date:                   Sun, 13 Aug 2017 17:05:38 GMT
{"transactionStatus": "ACTC",
"_links": {
        "self": {"href": "/psd2/v2/payments/sepa-credit-transfers/3d9a81b3-
a47d-4130-8765-a9c0ff861100"},
        "status": {"href": "/psd2/v2/payments/sepa-credit-
transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100/status"},
        "startAuthorisation": {"href": "/psd2/v2/payments/sepa-credit-
transfers/3d9a81b3-a47d-4130-8765-a9c0ff861100/cancellation-
authorisations"}
  }
}
```

4 Account Information Service

Supported Sub-Services

This specification foresees different types of account information services:

- Transaction lists for a given account with transactions with booking status booked or pending including balances if applicable,
- List of standing orders of a given account, reported as transactions with booking status information,
- Balances of a given account,
- A list of addressable accounts,
- Account details of a given account or of the list of all accessible accounts relative to a granted consent, and
- Account details might include the account owner name, where specific requirements on the consent process might apply, see below.

Hereby the definition of the list of **addressable** and **accessible** accounts is as follows:

Definition: The list of **addressable** accounts of an ASPSP related to a PSU is the list of accounts of a PSU which are open for access through the XS2A API according to the definition of payment accounts provided by [PSD2].

Definition: The list of **accessible** accounts of an ASPSP related to a PSU's consent is the list of accounts, where the consent of the PSU has been granted to at least one of the defined account information types.

Note: The Read Data Request for the list of addressable accounts and for account details of a given account is syntactically identical. The difference is only in the underlying consent resource, referred to through the HTTP header parameter "Consent-ID".

Example: An ASPSP is providing IBAN1 and IBAN2 to a PSU. The PSU has granted the TPP the consent to access transactions and balances of IBAN1 only. In this case, the addressable accounts are IBAN1 and IBAN2, the list of accessible accounts consists only of IBAN1.

Establishing Consent and Reading Account Data

Within this specification, the Account Information Service is separated in two phases:

- Establish Consent on Account Information as specified in detail in [oFA-CO].

  Within this phase of the Account Information Service, the PSU is giving the consent to the AISP on

  - the accounts accessible for the Account Information Service,
  - the type of Account Information Service to grant an access to (see list at the beginning of this section),
  - the multiplicity of the Account Information Service, i.e. a one-off or recurring access, and
  - in the latter case on the duration of the consent in days or the maximum offered by the ASPSP and optionally the frequency of a recurring request.

  This consent is then authorised by the PSU towards the ASPSP with an SCA as mandated by [EBA-RTS].

  The result of this process is a consent resource. A link to this resource is returned to the AISP within this process. The TPP can retrieve the consent object by submitting a GET method on this resource. This object contains a.o. the detailed access rights, the current validity and a Consent-ID token.

  NOTE: Please note that the Establish Consent on Account Information process as defined in [oFA-CO] technically allows to address accounts not covered by PSD2, e.g. loan accounts. Such a consent request will only be authorised within premium services of a related API Access Schemes and rejected in a PSD2 compliance solution.

- Read Account Data

  Within this phase, the AISP gets access to the account data as defined by the PSU’s consent, see above. The Read Account Data Request is addressing the corresponding consent resource by using the above mentioned link to this resource.

  The Read Account Data Request will indicate

  - the type of account data to be accessed,
  - the identification of the addressed account, where applicable,
  - whether a PSU has directly initiated the request real-time,

- whether balances should be delivered in addition where applicable,
- in case of transaction lists as Account Information type additionally
    - the addressed account identification and
    - the period of the transaction list
    - in addition optionally a delta-flag indicating the request for a delta-report relative to the last request with additional data.
    - the preferred formats of the transaction lists.

For the account access, the usual bank accounts and (credit) card accounts are separated on end-points, since the data is usually separated in the ASPSP backend.

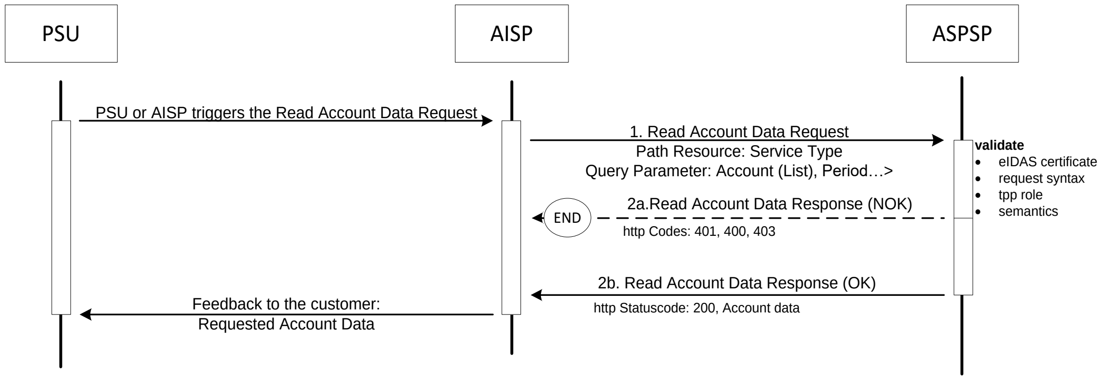
In case of a one-off consent, the access might be denied if the AISP is requesting the data more than once or if the validity of the consent has been timed out, e.g. after 20 minutes of the finalisation of the consent mechanism, depending on the ASPSP implementation.

The read data access will be further denied in case where the type of Account Information Service does not comply with the consented service, or if the actual access is not matching the consented duration or frequency.

If the PSU's consent is given to access a list of accounts, the frequency of the access is checked by the ASPSP per account that has been accessed and per PSU that has given consent for the access.

4.1 Read Account Data Flow

The Read Account Data flow requires first the flow for establishing the related consent within the XS2A APIs. These flows are provided in [oFA-CO]. The actual Read Account Data flow then is independent from the corresponding Consent Management flow. It is a simple Request/Response process as follows:


PSU
AISP
ASPSP
PSU or AISP triggers the Read Account Data Request
1. Read Account Data Request
Path Resource: Service Type
Query Parameter: Account (List), Period…>
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
2a.Read Account Data Response (NOK)
END
http Codes: 401, 400, 403
2b. Read Account Data Response (OK)
http Statuscode: 200, Account data
Feedback to the customer:
Requested Account Data

4.2 Data Overview Account Information Service

The following table defines the technical description of the abstract data model as defined in [oFA-OR-Com]for the account information service. The columns give an overview on the API protocols as follows:

- The "Data element" column is using the abstract data elements following [oFA-OR-Com] to deliver the connection to rules and role definitions in this document.
- The "Attribute encoding" is giving the actual encoding definition within the XS2A API as defined in this document.
- The "Location" columns define, where the corresponding data elements are transported as HTTP parameters, resp. are taken from eIDAS certificates. For HTTP Parameters, the "path" subsumes host, port and API basepath.
- The "Usage" column gives an overview on the usage of data elements in the different API Calls. Within [oFA-OR-Com], the XS2A calls are described as abstract API calls. These calls will be technically realised as HTTP POST, PUT, DELETE and GET commands. The calls are divided into the following calls:
  - The Read Data Request is the request to retrieve Account Information data, which is addressed to different endpoints with different parameters.

NOTE: The account information service requires first a technical consent on account access confirmed by SCA of the PSU. This service is described in [oFA-CO].

The following usage of abbreviations in the Location and Usage columns is defined, cp. also [oFA-OR-Com] for details.

- x: This data element is transported on the corresponding level.
- m: Mandatory
- o: Optional for the TPP to use
- c: Conditional. The Condition is described in the API Calls, condition defined by the ASPSP

The following table does not only define requirements on request messages but also requirements on data elements for the response messages. Tese requirements only apply to positive responses (i.e. HTTP response code 2xx). In case of error, generic error information as defined in [oFA PFSM] is provided together with an HTTP response code 40x.

Remark: The more technical functions like GET …/{consentId} and GET …/{authorisationId} and the Cancellation Request are not provided in this overview but only covered in detail in the related Open API files.

Data element	Attribute encoding	Location					Usage	
		Path	Query Param.	Header	Body	Certificate	Read Data Req.	Read Data Resp
Provider Identification		x					m	
TPP Registration Number						x	m	
TPP Name						x	m	
TPP Roles						x	m	
TPP National Competent Authority						x	m	
Request Identification	X-Request-ID			x			m	m
Consent ID[8]	Consent-ID			x			m	
Access Token (from optional OAuth2)	Authorization			x			c	
API Client Signing Electronic Signature	x-jws-signature			x			c	
Further signature related data	Digest			x			c	
PSU Message Information	psuMessage				x			o
API Client Message Information	apiClientMessages				x			o
IP Address PSU	PSU-IP-Address			x			c	
PSU IP Port	PSU-IP-Port			x			o	

[8] The related token is retrieved by processes defined in [oFA-CO]. Please note that the syntax in the Consent API is consentId since the attribute is transported on body level there.

Data element	Attribute encoding	Location					Usage	
		Path	Query Param.	Header	Body	Certificate	Read Data Req.	Read Data Resp
Further PSU related Information	PSU-Accept			x			o	
	PSU-Accept-Charset			x			o	
	PSU-Accept-Encoding			x			o	
	PSU-Accept-Language			x			o	
	PSU-Http-Method			x			o	
	PSU-Device-ID			x			o	
PSU User Agent	PSU-User-Agent			x			o	
GEO Information	PSU-Geo-Location			x			o	
PSU Account	accountId	x					c	
PSU Account	account				x			m
Date From	dateFrom		x				c	
Date To	dateTo		x				c	
Transaction From	entryReferenceFrom		x				o	
Booking Status	bookingStatus		x				o	
Delta Indicator	deltaList		x				o	
With Balance Flag	withBalance		x				o	
PSU Account List	Array of accountDetails				x			c
PSU Account Details	accountDetails				x			c
Balances	balances				x			c
Transactions	transactions				x			c

Data element	Attribute encoding	Location					Usage	
		Path	Query Param.	Header	Body	Certificate	Read Data Req.	Read Data Resp
PSU Card Account Details	accountDetails				x			c
Card Transactions	cardTransactions				x			c

PSU IP Address/Port and Further PSU related Information

The above table addresses several PSU related context data. These data, its importance and its usage are defined in detail in [oFA PFSM]. They are not mentioned anymore in the following detailed definitions for matter of better readability, as long as the usage is not mandated.

4.3 Multi-currency Account Specifics for Account Information

The methods on multicurrency accounts for account information differ in the inter-face due to the fact, that a collection of accounts is addressed. In the following the differences are described on abstract level. Of course, the sub-accounts can also be addressed as dedicated accounts by adding the sub account currency explicitly. This case is not further mentioned specifically.

Multicurrency Accounts in Reading Accounts or Account Details

The ASPSP will decide in its implementation whether to grant data access to a multicurrency account on aggregation level, on aggregation and sub-account level, or only on sub-account level.

Multicurrency Accounts in Reading Balances on Aggregation Level

The consequence for this function is that an array of balances of all sub-accounts are returned, if a multicurrency account is addressed on aggregation level. The currency of the respective sub-account is implicitly provided as the currency of the balanceAmount element within the balance, cp Section 4.4.3 for an example.

Multicurrency Accounts in Reading Transactions on Aggregation Level

The consequence for this function is that the list of transactions will contain all transactions of all sub-accounts, if a multicurrency account is addressed on aggregation level. In this case the payment transactions contained in the report may have different transaction currencies, see Section 4.4.4 for an example.

4.4 Read Account Data Requests

The following sections define reading access on current accounts.

4.4.1 Read Account List

Call

```
GET /v2/accounts {query-parameters}
```

Reads a list of current accounts, with balances where required. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system. The addressed list of accounts depends then on the PSU ID and the stored consent addressed by consentId, respectively the OAuth2 access token.

Note: If the consent is granted only to show the list of addressable accounts, much less details are displayed about the accounts. Specifically hyperlinks to balances or transaction endpoint should not be delivered then.

Note: If the details returned in this call with the access rights "accountDetails", "balances", "transactions" or "ais" are not sufficient, then more details can be retrieved by addressing the /accounts/{account-id} endpoint, cp. Section 4.4.2.

Query Parameters

Attribute	Type	Condition	Description
withBalance	Boolean	Optional	If contained, this function reads the list of accessible payment accounts including the booking balance, if granted by the PSU in the related consent and available by the ASPSP. This parameter might be ignored by the ASPSP.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Consent-ID	Max70Text	Mandatory	resulting from the "Establish Consent on Account Information" performed via this API before.
PSU-IP-Address	String	Conditional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained

Attribute	Type	Condition	Description
			if and only if this request was actively initiated by the PSU.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.

Request Body

No request body

Response Code

HTTP Response Code equals 200.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
accounts	Array of Account Details	Mandatory	In case, no account is accessible, the ASPSP shall return an empty array.As this is also considered a positive response, the Response code must still be 200.

Example

Response body (Example 1)

Response in case of an example, where the consent has been given on two different IBANs

```
{"accounts":
   [
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e80f",
       "iban": "DE2310010010123456789",
       "currency": "EUR",
       "product": "Girokonto",
       "cashAccountType": "CACC",
       "name": "Main Account",
       "_links": {
            "balances": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80f/balances"},
            "transactions": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80f/transactions"}}
      },
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e81e",
       "iban": "DE2310010010123456788",
       "currency": "USD",
       "product": "Foreign Currency Account",
       "cashAccountType": "CACC",
       "name": "US Dollar Account",
       "_links": {
            "balances": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e81e/balances" }}
       }
]}
```

Response body (Example 2)

Response in case of an example where consent on transactions and balances has been given to a multicurrency account which has two sub-accounts with currencies EUR and USD, and where the ASPSP is giving the data access only on sub-account level:

```
{"accounts":
   [
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e80f",
       "iban": "DE2310010010123456788",
       "currency": "EUR",
       "product": "Girokonto",
       "cashAccountType": "CACC",
       "name": "Main Account",
       "_links": {
            "balances": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80f/balances"},
            "transactions": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80f/transactions"}}
      },
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e81e",
       "iban": "DE2310010010123456788",
       "currency": "USD",
       "product": "Foreign Currency Account ",
       "cashAccountType": "CACC",
       "name": "US Dollar Account",
       "_links": {
            "balances": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e81e/balances"},
            "transactions": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e81e/transactions"} }
       }
]}
```

Response body (Example 3)

Response in case of an example where consent on balances and transactions has been given to a multicurrency account which has two sub-accounts with currencies EUR and USD and where the ASPSP is giving the data access on aggregation level and on sub-account level:

```
{"accounts":
   [
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e80f",
       "iban": "DE2310010010123456788",
       "currency": "XXX",
       "product": "Multi currency account",
       "cashAccountType": "CACC",
       "name": "Aggregation Account",
       "_links": {
            "balances": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e333/balances"},
            "transactions": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e333/transactions"}}
      },
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e80e",
       "iban": "DE2310010010123456788",
       "currency": "EUR",
       "product": "Girokonto",
       "cashAccountType": "CACC",
       "name": "Main Account",
       "_links": {
            "balances": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80e/balances"},
            "transactions": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80e/transactions"}}
      },
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e81d",
       "iban": "DE2310010010123456788",
       "currency": "USD",
       "product": "Fremdwährungskonto",
       "cashAccountType": "CACC",
       "name": "US Dollar Account",
       "_links": {
            "balances": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e81d/balances"},
            "transactions": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e81d/transactions"} }
       }
]}
```

4.4.2 Read Account Details

Call

```
GET /v2/accounts/{account-id} {query-parameters}
```

Reads details about a current account, with balances where required. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system. The addressed details of this account depends then on the stored consent addressed by consentId, respectively the OAuth2 access token.

Remark: This account-id can be a tokenised identification due to data protection reason since the path information might be logged on intermediary servers within the ASPSP sphere. This account-id then can be retrieved by the "GET Account List" call, cp. Section 4.4.1 or from the related consent resource.

NOTE: The account-id can represent a multicurrency account. In this case the currency code is set to "XXX".

Path Parameters

Attribute	Type	Description
account-id	Max70Text	This identification is denoting the addressed account. The account-id is retrieved by using a "Read Account List" call. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Query Parameters

Attribute	Type	Condition	Description
withBalance	Boolean	Optional	If contained, this function reads the details of the addressed account including the booking balance, if granted by the PSU's consent and if supported by

Attribute	Type	Condition	Description
			ASPSP. This data element might be ignored by the ASPSP.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Consent-ID	Max70Text	Mandatory	
PSU-IP-Address	String	Conditional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.

Request Body

No request body

Response Code

HTTP Response Code equals 200.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
account	Account Details	Mandatory	

Example

Response body for a regular account

```
 {"account":
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e80f",
       "iban": "FR7612345987650123456789014",
       "currency": "EUR",
       "ownerName": "Heike Mustermann",
       "product": "Girokonto",
       "cashAccountType": "CACC",
       "name": "Main Account",
       "_links": {
            "balances": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80f/balances"},
            "transactions": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80f/transactions"}}
        }
}
```

Response body for a multi-currency account

```
 {"account":
   {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e80f",
     "iban": "FR7612345987650123456789014",
     "currency": "XXX",
     "ownerName": "Heike Mustermann",
     "product": "Multicurrency Account",
     "cashAccountType": "CACC",
     "name": "Aggregation Account",
     "_links": {
            "balances": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80f/balances"},
            "transactions": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-
9853-f5400a64e80f/transactions"}}
      }
}
```

4.4.3 Read Balance

Call

```
GET /v2/accounts/{account-id}/balances
```

Reads balance data from a given current account addressed by "account-id".

Path Parameters

Attribute	Type	Description
account-id	Max70Text	This identification is denoting the addressed account. The account-id is retrieved by using a "Read Account List" call. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Query Parameters

No specific query parameters.

Response Code

HTTP Response Code equals 200.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Conditional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.
Consent-ID	String	Mandatory	
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.

Request Body

No request body.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
account	Account Reference	Mandatory	Identifier of the addressed account.
balances	Array of Balance	Mandatory	A list of balances regarding this account, e.g. the current balance, the last booked balance.

Example

Response body (Example 1)

Response in case of a regular account.

```
{
  "account": {"iban": "FR7612345987650123456789014"},
  "balances":
      [{"balanceAmount": {"currency": "EUR", "amount": "500.00"},
        "balanceType": "closingBooked",
        "referenceDate": "2017-10-25"
       },
       {"balanceAmount": {"currency": "EUR","amount": "900.00"},
        "balanceType": "expected",
        "lastChangeDateTime": "2017-10-25T15:30:35.035Z"
       }]
}
```

Response body (Example 2)

Response in case of a multicurrency account with one account in EUR, one in USD, where the ASPSP has delivered a link to the balance endpoint relative to the aggregated multicurrency account (aggregation level)

```
{
  "balances":
      [{"balanceAmount": {"currency": "EUR", "amount": "500.00"},
        "balanceType": "closingBooked",
        "referenceDate": "2017-10-25"
       },
       {"balanceAmount": {"currency": "EUR", "amount": "900.00"},
        "balanceType": "expected",
        "lastChangeDateTime": "2017-10-25T15:30:35.035Z"
       },
       {"balanceAmount": {"currency": "USD", "amount": "350.00"},
        "balanceType": "closingBooked",
        "referenceDate": "2017-10-25"
       },
       {"balanceAmount": {"currency": "USD", "amount": "350.00"},
        "balanceType": "expected",
        "lastChangeDateTime": "2017-10-24T14:30:21Z"
           }]
}
```

Response body (Example 3)

Response in case of a regular account where the corresponding balances in the online channel is reported independently from account statements with fixed dates, i.e. always displaying running balance for current time.

```
{
   "balances": [
      {
       "balanceAmount": {"currency": "EUR", "amount": "1000.00"},
       "balanceType": "interimBooked"
         },
      {
         "balanceAmount": {"currency": "EUR", "amount": "300.00"},
         "balanceType": "interimAvailable"
      },
      {
         "balanceAmount": {"currency": "EUR", "amount": "5300.00"},
         "balanceType": "interimAvailable",
         "creditLimitIncluded": true
      }
```

```
    ]
}
```

4.4.4 Read Transaction List

Call

```
GET /v2/accounts/{account-id}/transactions {query-parameters}
```

Reads account transaction data from a given account addressed by "account-id". This can be either booked or pending transactions or a list of standing orders as further transactional information.

Note: The ASPSP might use standard compression methods on application level for the response message as indicated in the content encoding header. In case of returning camt.05x formats, several camt.05x files might be contained in one response. Some ASPSPS e.g. separate camt.05x files per booking day – in analogy to the same provision in online channels.

Note: In case of using pagination, the call on the given pagination links follows the same requirements as for this call, just exchanging the path itself by the pagination path.

Remark: Please note that all path parameters might be already given in detail implicitly by the response of the "Read Account List" call within the _links subfield.

Path Parameters

Attribute	Type	Description
account-id	String	This identification is denoting the addressed account. The account-id is retrieved by using a "Read Account List" call. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Query Parameters

Attribute	Type	Condition	Description
dateFrom	ISODate	Conditional	Starting date (inclusive the date dateFrom) of the transaction list, mandated if no delta access is required and if bookingStatus does not equal "information". Might be ignored if a delta function is used or if bookingStatus equals "information". For booked transactions, the relevant date is the booking date. For pending transactions, the relevant date is the entry date, which may not be transparent neither

Attribute	Type	Condition	Description
			in this API nor other channels of the ASPSP. If the bookingStatus equals "all", this date might be ignored for all transactions referred to by bookingStatus "information".
dateTo	ISODate	Optional	End date (inclusive the data dateTo) of the transaction list, default is "today" if not given. Might be ignored if a delta function is used. For booked transactions, the relevant date is the booking date. For pending transactions, the relevant date is the entry date, which may not be transparent neither in this API nor other channels of the ASPSP. If the bookingStatus equals "all", this date might be ignored for all transactions referred to by bookingStatus "information".
entryReferenceFrom	Max35Text	Optional if supported by API provider	This data attribute is indicating that the AISP is in favour to get all transactions after the transaction with entryReference identification entryReferenceFrom alternatively to the above defined period. This is an implementation of a delta access. If this data element is contained, the entries "dateFrom" and "dateTo" might be ignored by the ASPSP if a delta report is supported.
pageSize	Integer	Optional if supported by API provider	This query parameter defines the transaction entries per call to be retrieved for **extended** services. If not supported, then the call is rejected. If supported by the ASPSP and if the value is higher than maxPageSize as defined by the ASPSP in its documentation, then the call is rejected.

Attribute	Type	Condition	Description
bookingStatus	Max35Text	Mandatory	Permitted codes are "booked", "pending", "both", "information" and "all". "booked" shall be supported by the ASPSP. To support the "pending" feature is conditional for the ASPSP. It is mandated if the ASPSP online channels provide pending transactions to the PSU. Error code if not supported. To support the "both" feature is optional for the ASPSP, Error code if not supported. If supported, "both" means to request lists of transactions of bookingStatus either "pending" or "booked". To support the "information" feature is optional for the ASPSP. Currently the booking status “information” only covers standing orders. Error code if not supported. To support the "all" feature is optional for the ASPSP, Error code if not supported. If supported, "all" means to request lists of transaction of any bookingStatus ("pending", "booked" or "information").
deltaList	Boolean	Optional if supported by API provider	This data attribute is indicating that the AISP is in favour to get all transactions after the last report access for this PSU on the addressed account. This is another implementation of a delta access-report. This delta indicator might be rejected by the ASPSP if this function is not supported. If this data element is contained, the entries "dateFrom" and "dateTo" might be ignored by the ASPSP if a delta report is supported.

Attribute	Type	Condition	Description
withBalance	Boolean	Optional	If contained, this function reads the list of transactions including the booking balance, if granted by the PSU in the related consent and available by the ASPSP. This parameter might be ignored by the ASPSP.
cardBrand	Max35Text	Optional, if supported by the ASPSP	This attribute filters transactions according to their corresponding brand.

NOTE: In case of bookingStatus equals "information", the query parameters dateFrom, dateTo, withBalance deltaList and entryReferenceFrom will be ignored and have no effect on the result.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Conditional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.
Consent-ID	Max70Text	Mandatory	
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.

Attribute	Type	Condition	Description
Accept	String	Optional	The TPP can indicate the formats of account reports supported together with a priorisation following the HTTP header definition. The formats supported by this specification are • xml • JSON • text

Remark: The Berlin Group intends to apply for vnd-entries within the "accept" attribute for camt.05x and MT94x formats to scope with different account report formats available for the PSU e.g. in a corporate context. These values will be added to this specification as soon as available. This will then lead to expressions like /application/vnd.BerlinGroup.camt.053+xml etc. The TPP then could e.g. say: "I prefer camt.054, but take camt.053 if this is not available." This solution is recommended as a best practice until it is fully specified. In this example this would deliver the following accept header expression:

```
Accept: /application/vnd.BerlinGroup.camt.054+xml;q=0.9, /application/vnd.BerlinGroup.camt.053+xml;q=0.8
```

In addition, these best practices allow to differentiate technical sub versions of camt, i.e. it could be stated that "I prefer camt.054.001.08 (the new sub version), but take (the older sub version) camt.054.001.02 if this is not available." This is to supports ASPSPs in migrating the technical camt formats.

```
Accept: /application/vnd.BerlinGroup.camt.054.001.08+xml;q=0.9, /application/vnd.BerlinGroup.camt.054.001.02+xml;q=0.8
```

Request Body

No request body.

Response Code

HTTP Response Code equals 200.

Response Header

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	Possible values are: • application/json • application/xml • text/plain
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

In case the ASPSP returns a **camt.05x** XML structure, the response body consists of either a camt.052 or camt.053 format. The camt.052 may include pending payments which are not yet finally booked. The ASPSP will decide on the format due to the chosen parameters, specifically on the chosen dates relative to the time of the request. In addition the ASPSP might offer camt.054x structure e.g. in a corporate setting.

In case the ASPSP returns **a MT94x** content, the response body consists of an MT940 or MT942 format in a text structure. The MT942 may include pending payments which are not yet finally booked. The ASPSP will decide on the format due to the chosen parameters, specifically on the chosen dates relative to the time of the request.

A JSON response is defined as follows:

Attribute	Type	Condition	Description
account	Account Reference	Mandatory	Identifier of the addressed account.
transactions	Account Report	Optional	JSON based account report. This account report contains transactions resulting from the query parameters. Note: It is recommended to provide remittance information attributes already here and not only in transaction details as specified in Section4.4.5. Such an implementation avoids too frequent usage of transaction details by API Clients.

Attribute	Type	Condition	Description
balances	Array of Balance	Optional	A list of balances regarding this account, which might be restricted to the current balance.
_links	Links	Optional	A list of hyperlinks to be recognised by the TPP. Type of links admitted in this response: "download": a link to a resource, where the transaction list might be downloaded from in case where transaction lists have a huge size. **Remark:** This feature shall only be used where camt-data is requested which has a huge size.

Examples for AIS for booked and pending transactions

Request

```
GET https://api.testbank.com/psd2/v2/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/transactions?dateFrom=2017-07-01&dateTo=2017-07-
30&bookingStatus=both
Accept: application/json, text/plain;q=0.9, application/xml;q=0.8
Consent-ID: 123cons456
```

Response (Example 1)

Response in JSON format for an access on a regular account

```
HTTP/1.x 200 Ok
X-Request-ID:            99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                    Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:            application/json

{“account”: {"iban": "DE2310010010123456788" },
"transactions":
   {"booked":
     [{
       "transactionId": "1234567",
       "creditor": {"name": "John Miles"},
       "creditorAccount": {"iban": "DE67100100101306118605"},
       "transactionAmount": {"currency": "EUR", "amount": "-256.67"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": ["Example 1"]
      },{
       "transactionId": "1234568",
       "debtor": {"name": "Paul Simpson"},
       "debtorAccount": {"iban": "NL76RABO0359400371"},
       "transactionAmount": {"currency": "EUR", "amount": "343.01"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": ["Example 2"]
      }],
   "pending":
     [{
       "transactionId": "1234569",
       "creditor": {"name": "Claude Renault"},
       "creditorAccount": {"iban": "FR7612345987650123456789014"},
       "transactionAmount": {"currency": "EUR", "amount": "-100.03"},
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": ["Example 3"]
      }],
   "_links":
      {"account": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f"}}
   }
}
```

Response (Example 2)

Response in case of huge data amount as a download.

```
HTTP/1.x 200 OK
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json

{
  "_links": {"download": {"href": "www.test-
api.com/xs2a/v2/accounts/12345678999/transactions/download/"}}
}
```

Response (Example 3)

Response in JSON format for an access on a multicurrency account on aggregation level

```
HTTP/1.x 200 OK
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
```

```
Content-Type:           application/json

{"account": {"iban": "DE40100100103307118608"},
 "transactions":
   {"booked":
     [{
       "transactionId": "1234567",
       "creditor": {"name": "John Miles"},
       "creditorAccount": {"iban": "DE67100100101306118605"},
       "transactionAmount": {"currency": "EUR", "amount": "-256.67"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": ["Example 1"]
      },{
       "transactionId": "1234568",
       "debtor": {"name": "Paul Simpson"},
       "debtorAccount": {"iban": "NL76RABO0359400371"},
       "transactionAmount": {"currency": "EUR", "amount": "343.01"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": ["Example 2"]
      },{
       "transactionId": "1234569",
       "debtor": {"name": "Pepe Martin"},
       "debtorAccount": {"iban": "SE9412309876543211234567"},
       "transactionAmount": {"currency": "USD", "amount": "100"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": ["Example 3"]
      }],
   "pending":
     [{
       "transactionId": "1234570",
       "creditor": {"name": "Claude Renault"},
       "creditorAccount": {"iban": "FR7612345987650123456789014"},
       "transactionAmount": {"currency": "EUR", "amount": "-100.03"},
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": ["Example 4"]
      }],
   "_links":
      {"account": {"href": "/psd2/v2/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f"}}
   }
}
```

Examples for AIS for standing orders

Request

```
GET https://api.testbank.com/psd2/v2/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/transactions? bookingStatus=information
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 06 Aug 2017 15:05:45 GMT
Accept:                 application/json
Consent-ID:             123cons456
```

Response

Response in JSON format for a list of standing orders

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json

{"account": {"iban": "DE2310010010123456788" },
 "transactions":
   {"information":
     [{
       "creditor": {"name": "John Miles"},
       "creditorAccount": {"iban": "DE67100100101306118605"},
       "transactionAmount": {"currency": "EUR", "amount": "256.67"},
       "remittanceInformationUnstructured": ["Example 1"],
       "bankTransactionCode": "PMNT-ICDT-STDO",
       "additionalInformationStructured":
         {"standingOrderDetails":
           {"startDate": "2018-03-01",
            "endDate": "2020-06-31",
            "executionRule": "preceding",
            "frequency": "Monthly",
            "dayOfExecution": "24"
      }]
    }
}
```

4.4.5 Read Transaction Details

Call

```
GET /v2/accounts/{account-id}/transactions/{transactionId}
```

Reads transaction details from a given transaction addressed by “transactionId” on a given account addressed by "account-id". This call is only available on transactions as reported in a JSON format.

Remark: Please note that all path parameters might be already given implicitly by the corresponding entry of the response of the "Read Transaction List" call within the _links subfield.

Path Parameters

Attribute	Type	Description
account-id	String	This identification is denoting the addressed account, where the transaction has been performed.
transactionId	String	This identification is given by the attribute transactionId of the corresponding entry of a transaction list.

Query Parameters

No Query Parameters

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Conditional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.
Consent-ID	String	Mandatory	
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.

Request Body

No request body.

Response Code

HTTP Response Code equals 200.

Response Header

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	Possible values are: • application/json
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
transactionsDetails	Transactions	Optional	

Example

Request

```
GET https://api.testbank.com/psd2/v2/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/transactions/1234567
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 06 Aug 2017 15:05:46 GMT
Consent-ID:             123cons456
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json

{"transactionsDetails":
   {
       "transactionId": "1234567",
       "creditor": {"name": "John Miles"},
       "creditorAccount": {"iban": "DE67100100101306118605"},
       "mandateId": "Mandate-2018-04-20-1234",
       "transactionAmount": {"currency": "EUR", "amount": "-256.67"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": ["Example 1"],
       "bankTransactionCode": "PMNT-RDDT-ESDD",
   }
}
```

Remark: As shown by this example, a very typical additional details of a transaction is a SEPA Mandate ID.

4.5 Read Card Account Data Requests

4.5.1 Read Card Account List

Call

```
GET /v2/card-accounts
```

Reads a list of card reconciliation accounts with additional information, e.g. balance information. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system. The addressed list of card reconciliation accounts depends then on the PSU ID and the stored consent addressed by consentId, respectively the OAuth2 access token.

Query Parameters

No query parameter supported.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Conditional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.

Attribute	Type	Condition	Description
Consent-ID	String	Mandatory	Resulting from the "Establish Consent on Account InformationTransaction" performed via this API before.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.

Request Body

No request body

Response Code

HTTP Response Code equals 200.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
cardAccounts	Array of Card Account Details	Mandatory	In case, no card-account is accessible, the ASPSP shall return an empty array. As this is also considered a positive response, the Response Code must still be 200.

Example

Response body

```
{
  "cardAccounts": [
    {
      "resourceId": "3d9a81b3-a47d-4130-8765-a9c0ff861b99",
      "maskedPan": "525412******3241",
      "currency": "EUR",
      "name": "Main",
      "product": "Basic Credit",
      "status": "enabled",
      "creditLimit": { "currency": "EUR", "amount": "15000" },
      "balances": [
        {
          "balanceType": "interimBooked",
          "balanceAmount": { "currency": "EUR", "amount": "14355.78" }
        },{
          "balanceType": "nonInvoiced",
          "balanceAmount": { "currency": "EUR", "amount": "4175.86" }
        }
      ],
      "_links": {
        "transactions": {
          "href": "/psd2/v2/card-accounts/3d9a81b3-a47d-4130-8765-
a9c0ff861b99/transactions"
        }
      }
    }
  ]
}
```

4.5.2 Read Card Account Details

Call

```
GET /v2/card-accounts/{account-id}
```

Reads details about a card reconciliation account. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system. The addressed details of this account depends then on the stored consent addressed by consentId, respectively the OAuth2 access token.

Path Parameters

Attribute	Type	Description
account-id	String	This identification is denoting the addressed card account. The account-id is retrieved by using a "Read Card Account List" call. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Query Parameters

No query parameters defined.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Conditional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.
Consent-ID	String	Mandatory	Identification of the access consent as granted by the PSU.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.

Request Body

No request body

Response Code

HTTP Response Code equals 200.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
cardAccount	Card Account Details	Mandatory	

Example

```
{
  "cardAccount":
    {
      "resourceId": "3d9a81b3-a47d-4130-8765-a9c0ff861b99",
      "maskedPan": "525412******3241",
      "currency": "EUR",
      "debitAccounting": true,
      "ownerName": "Heike Mustermann",
      "name": "Main",
      "product": "Basic Credit",
      "status": "enabled",
      "creditLimit": { "currency": "EUR", "amount": "15000" },
      "balances": [
        {
          "balanceType": "interimBooked",
          "balanceAmount": { "currency": "EUR", "amount": "14355.78" }
        },{
          "balanceType": "nonInvoiced",
          "balanceAmount": { "currency": "EUR", "amount": "4175.86" }
        }
      ],
      "_links": {
        "transactions": {
          "href": "/psd2/v2/card-accounts/3d9a81b3-a47d-4130-8765-
a9c0ff861b99/transactions"
        }
      }
    }
}
```

4.5.3 Read Card Account Balance

Call

```
GET /v2/card-accounts/{account-id}/balances
```

Reads balance data from a given card reconciliation account addressed by "account-id".

Remark: This account-id can be a tokenised identification due to data protection reason since the path information might be logged on intermediary servers within the ASPSP sphere. This account-id then can be retrieved by the "GET Card Account List" call, cp. Section 4.5.1 or from the related consent resource.

Path Parameters

Attribute	Type	Description
account-id	String	This identification is denoting the addressed card account. The account-id is retrieved by using a "Read Account List" call. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Query Parameters

No specific query parameters.

Response Code

HTTP Response Code equals 200.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Conditional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.
Consent-ID	String	Mandatory	Identification of the corresponding consent as granted by the PSU.

Attribute	Type	Condition	Description
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.

Request Body

No request body.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
cardAccount	Account Reference	Mandatory	Identifier of the addressed card account.
debitAccounting	Boolean	Optional	If true, the amounts of debits on the reports are quoted positive with the related consequence for balances. If false, the amount of debits on the reports are quoted negative.
balances	Array of Balance	Mandatory	A list of balances regarding this card account, e.g. the current balance, the last booked balance.

Example

```
{
  "cardAccount": {"maskedPan": "525412******3241"},
  "debitAccounting": true,
  "balances": [
    {
      "balanceAmount": { "currency": "EUR", "amount": "14355.78"},
      "balanceType": "interimBooked"
    },{
      "balanceAmount": { "currency": "EUR", "amount": "4175.86"},
      "balanceType": "nonInvoiced",
    }
  ]
}
```

4.5.4 Read Card Account Transaction List

Call

```
GET /v2/card-accounts/{account-id}/transactions {query-parameters}
```

Reads account data from a given card reconciliation account addressed by "account-id".

Remark: This account-id is a tokenised identification due to data protection reason since the path information might be logged on intermediary servers within the ASPSP sphere. This account-id then can be retrieved by the "GET Card Account List" call, cp. Section 4.5.1.

Note: The ASPSP might use standard compression methods on application level for the response message as indicated in the content encoding header.

Remark: Please note that all path parmetersmight be already given implicitly by the response of the "Read Card Account List" call within the _links subfield.

Path Parameters

Attribute	Type	Description
account-id	String	This identification is denoting the addressed card reconcilitation account. The account-id is retrieved by using a "Read Card Account List" call. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Query Parameters

Attribute	Type	Condition	Description
dateFrom	ISODate	Conditional	Starting date (inclusive the date dateFrom) of the transaction list, mandated if no delta access is required
dateTo	ISODate	Optional	End date (inclusive the data dateTo) of the transaction list, default is "today" if not given.
pageSize	Integer	Optional if supported by API provider	This query parameter defines the transaction entries per call to be retrieved for **extended** services. If not supported, then the call is rejected. If supported by the ASPSP and if the value is higher than maxPageSize as defined by the ASPSP in its documentation, then the call is rejected.
bookingStatus	String	Mandatory	Permitted codes are "booked", "pending" and "both" "booked" shall be supported by the ASPSP. To support the "pending" and "both" feature is optional for the ASPSP, Error code if not supported in the online banking frontend
deltaList	Boolean	Optional if supported by API provider	This data attribute is indicating that the AISP is in favour to get all transactions after the last report access for this PSU on the addressed account. This delta indicator might be rejected by the ASPSP if this function is not supported.
cardBrand	Max35Text	Optional, if supported by the ASPSP	This attribute filters transactions according to their corresponding brand.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Conditional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.
Consent-ID	String	Mandatory	Identification of the consent for this access as granted by the PSU.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.

Request Body

No request body.

Response Code

HTTP Response Code equals 200.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
cardAccount	Account Reference	Mandatory	Identifier of the addressed card reconliation account.

Attribute	Type	Condition	Description
debitAccounting	Boolean	Optional	If true, the amounts of debits on the reports are quoted positive with the related consequence for balances. If false, the amount of debits on the reports are quoted negative.
cardTransactions	Card Account Report	Optional	JSON based account report.
balances	Array of Balance	Optional	A list of balances regarding this account, which might be restricted to the current balance.
_links	Links	Optional	A list of hyperlinks to be recognised by the TPP. Type of links admitted in this response: "download": a link to a resource, where the transaction list might be downloaded from in case where transaction lists have a huge size.

Example

```
GET https://api.testbank.com/psd2/v2/card-accounts/3d9a81b3-a47d-4130-8765-
a9c0ff861b99/transactions?dateFrom=2017-10-01&dateTo= 2017-10-30
Accept: application/json, text/plain;q=0.9, application/xml;q=0.8
X-Request-ID:          99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Consent-ID:            123cons456
```

Response (Example 1)

Response in JSON format for an access on a regular account

```
HTTP/1.x 200 Ok
X-Request-ID:          99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                  Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:          application/json

{ "cardAccount": { "maskedPan": "525412******3241"},
  "debitAccounting": true,
  "cardTransactions": {
    "booked": [
      {
        "cardTransactionId": "201710020036959",
        "transactionAmount": { "currency": "EUR", "amount": "256.67" },
        "transactionDate": "2017-10-25",
        "bookingDate": "2017-10-26",
        "originalAmount": { "currency": "SEK", "amount": "2499" },
        "cardAcceptorAddress": {
          "townName": "STOCKHOLM",
          "country": "SE"
        },
        "maskedPan": "525412******3241",
        "proprietaryBankTransactionCode": "PURCHASE",
        "invoiced": false,
        "transactionDetails": "WIFIMARKET.SE"
      }, {
        "cardTransactionId": "201710020091863",
        "transactionAmount": { "currency": "EUR", "amount": "10.72" },
        "transactionDate": "2017-10-25",
        "bookingDate": "2017-10-26",
        "originalAmount": { "currency": "SEK", "amount": "99" },
        "cardAcceptorAddress": {
          "townName": "STOCKHOLM",
          "country": "SE"
        },
        "maskedPan": "525412******8999",
        "proprietaryBankTransactionCode": "PURCHASE",
        "invoiced": false,
        "transactionDetails": "ICA SUPERMARKET SKOGHA"
      }
    ],
    "pending": [ ],
    "_links": {
      "cardAccount": {
        "href": "/psd2/v2/card-accounts/3d9a81b3-a47d-4130-8765-
a9c0ff861b99"
      }
    }
  }
}
```

5 Confirmation of Funds Service

5.1 Overview Confirmation of Funds Service

The following table defines the technical description of the abstract data model as defined [oFA-OR-Com] for the three PSD2 services. The columns give an overview on the API protocols as follows:

- The "Data element" column is using the abstract data elements following [oFA-OR-Com] to deliver the connection to rules and role definitions in this document.
- The "Attribute encoding" is giving the actual encoding definition within the XS2A API as defined in this document.
- The "Location" columns define, where the corresponding data elements are transported as HTTP parameters, resp. are taken from eIDAS certificates. For HTTP Parameters, the "path" subsumes host, port and API basepath.
- The "Usage" column gives an overview on the usage of data elements in the different services and API Calls. Within [oFA-OR-Com], the XS2A calls are described as abstract API calls. These calls will be technically realised as HTTP POST command. The calls are divided into the following calls:
    - Confirmation Request, which is the only API Call for every transaction within the Confirmation of Funds service.

The following table does not only define requirements on request messages but also requirements on data elements for the response messages. These requirements only apply to positive responses (i.e. HTTP response code 2xx). Error handling is a generic framework feature and is specified in [oFA PFSM].

The following usage of abbreviations in the Location and Usage columns is defined, cp. also [oFA-OR-Com] for details.

- x: This data element is transported on the corresponding level.
- m: Mandatory
- o: Optional for the TPP to use
- c: Conditional. The Condition is described in the API Calls, condition defined by the ASPSP

Data element	Attribute encoding	Location				Usage	
		Path	Header	Body	Certificate	Conf. Req.	Conf Resp.
Provider Identification		x				m	
TPP Registration Number					x	m	
TPP Name					x	m	
TPP Role					x	m	
TPP National Competent Authority					x	m	
Request Identification	X-Request-ID		x			m	m
Consent ID	Consent-ID		x			c	
Further signature related data	Digest		x			c	
API Client Electronic Signature	x-jws-signature		x			c	
API Client Message Information	apiClientMessages			x			o
Card Number	cardNumber			x		o	
Account Number	account			x		m	
Name Payee	payee			x		o	
Transaction Amount	instructedAmount			x		m	

5.2 Confirmation of Funds Request

Call

```
POST /v2/funds-confirmations
```

Creates a confirmation of funds request at the ASPSP.

Query Parameter

No specific query parameter.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Optional	This field might be used in case where a consent was agreed between ASPSP and PSU through an OAuth2 based protocol, facilitated by the TPP.
Consent-ID	String	Conditional	Shall be provided if the consent of the PSU has been provided through the consent process for fundsconfirmations as defined in [oFA-CO]. Otherwise not used.

Request Body

Attribute	Type	Condition	Description
cardNumber	Max35Text	Optional	Card Number of the card issued by the PIISP. Should be delivered if available.
account	Account Reference	Mandatory	PSU's account number.
payee	Max70Text	Optional	The merchant where the card is accepted as an information to the PSU.
instructedAmount	Amount	Mandatory	Transaction amount to be checked within the funds check mechanism.

Response Code

The HTTP response code equals 200.

Response Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Response Body

Attribute	Type	Condition	Description
fundsAvailable	Boolean	Mandatory	Equals true if sufficient funds are available at the time of the request, false otherwise.

The following rules will apply in interpreting the Confirmation of Funds Request for multicurrency accounts:

The additional card number might support the choice of the sub-account.

If no card number, but the PSU account identifier is contained: check on default account registered by customer.

If no card number but the PSU and the account identifier with currency is contained: check the availability of funds on the corresponding sub-account.

If card number and the PSU account identifier is contained: check on sub-account addressed by card, if the addressed card is registered with one of the sub-accounts.

If the card number is not registered for any of the sub-accounts, or if the card number is registered for a different sub-account the card number might be ignored.

Example

```
POST https://api.testbank.com/psd2/v2/funds-confirmations
Content-Type:            application/json
X-Request-ID:            99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                    Sun, 06 Aug 2017 15:02:37 GMT

{  "cardNumber": "12345678901234",
   "account": {"iban": "DE23100120020123456789"},
   "instructedAmount": {"currency": "EUR", "amount": "123"}
}
```

Response Body

```
{"fundsAvailable": true}
```

6 References

[oFA-OR-Com] openFinance API Framework, Compliance Services, Operational Rules, Version 2.0, 31 October 2025

[oFA-CO] openFinance API Framework, Consent API for V2, Implementation Guidelines, Version 2.2, 31 October 2025

[oFA DaD] openFinance API Framework, Data Dictionary for V2.x, Version 2.3, 31 October 2025

[oFA DomDef] openFinance API Framework, Domestic AIS/PIS Definitions for Version 2.x, current version

[oFA PFSM] openFinance API Framework, Security Measures and Protocol Functions for Version 2.x, Version 2.3, 31 October 2025

[XS2A-SecB] NextGenPSD2 XS2A Framework, Security Bulletin, Version 1.1, 30 October 2020

[EBA-OP2] Opinion of the European Banking Authority on obstacles under Article 32(3) of the RTS on SCA and CSC, EBA/OP/2020/10, published 4 June 2020

[EBA-RTS] Commission Delegated Regulation (EU) 2018/389 of 27 November 2017 supplementing Directive 2015/2366 of the European Parliament and of the Council with regard to Regulatory Technical Standards for Strong Customer Authentication and Common and Secure Open Standards of Communication, C(2017) 7782 final, published 13 March 2018

[EBA-FR] Final Report, Draft Regulatory Technical Standards, amending Commission Delegated Regulation (EU) 2018/389 supplementing Directive (EU) 2015/2366 of the European Parliament and of the Council with regard to regulatory technical standards for strong customer authentication and common and secure open standards of communication, published 5 April 2022

[eIDAS] Regulation (EU) No 910/2014 of the European Parliament and of the Council on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market, 23 July 2014, published 28 August 2014

[ETSI PSD2] ETSI TS 119 495 V1.1.2; Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Qualified Certificate Profiles and TSP Policy Requirements under the payment services Directive (EU) 2015/2366

[PSD2] Directive (EU) 2015/2366 of the European Parliament and of the Council on payment services in the internal market, published 23 December 2015

[HAL] Kelley, M., "HAL - Hypertext Application Language", 2013-09-18, http://stateless.co/hal_specification.html

[FAPI-CBPIA] OpenID Foundation, Financial-grade API (FAPI) Working Group, Cross-Browser Payment Initiation Attack, https://bitbucket.org/openid/fapi/src/master/TR-Cross_browser_payment_initiation_attack.md, 3.01.2019

[OA-SecTop] OAuth 2.0 Security Best Current Practice draft-ietf-oauth-security-topics-13, Lodderstedt et al., 8 July 2019, https://tools.ietf.org/html/draft-ietf-oauth-security-topics-13

7 Annex A: Change Log

7.1 Changes from Version 2.0 to Version 2.1

The following changes have been applied in version 2.1 relative to version 2.0:

Section	Change	Reason
All Sections	The attribute consentId and paymentId have been restricted to max70Text, where not yet applied.	General approach for resource identifications.
All Sections	Some query parameters have been restricted to Max35Text	General approach for query parameters of type "string".
3.2	Rename TPP-SCA-Preference to Client-SCA-Preference. Rename TPP-Explicit-Authorisation-Preferred to Client-Explicit-Authorisation-Preferred Rename TPP-Rejection-NoFunds-Preferred to Client-Rejection-NoFunds-Preferred	Make these parameters also usable for direct access models for corporates. Prepare the API Framework better for corporate direct access.
3.4.1	Rename TPP-Rejection-NoFunds-Preferred to Client-Rejection-NoFunds-Preferred	Make these parameters also usable for direct access models for corporates. Prepare the API Framework better for corporate direct access.
4.4.4, 4.5.4	A pagination query parameter for transaction reports has been added for premium services.	CR115

7.1 Changes from Version 2.1 to Version 2.2

The following changes have been applied in version 2.2 relative to version 2.1:

Section	Change	Reason
2.1.1	Added an access method in the overview to retrieve the extended status of a bulk payment in JSON format	CR118

Section	Change	Reason
3.4.3.1	Added (optional) timestamps for startDate and endDate in standing orders for recurring instant payments.	CR123
3.4.3.2	Added a recommendation for rejecting bulk payment initiations with missing transaction identifiers on entry level.	Clarification
3.6	Added an optional endpoint to retrieve an extended status in JSON format for bulk payments, in analogy to pain.002 formats	CR118
4.5.4	corrected the attribute townName in examples for card account transaction lists.	Erratum
Several sections	Clarified that "card accounts" are referring to "card reconciliation accounts".	Clarification
6	Updated some references, specifically the data dictionary to be considered. This will lead to material changes in th API definitiions.	Updated openFinance API Framework

7.2 Changes from Version 2.2 to Version 2.3

The following changes have been applied in version 2.3 relative to version 2.2:

Section	Change	Reason
3.5	A remark has been added how to deal with the situation that several pain.002 messages are returned.	Clarification
4.4.4 and 4.5.4	Attribute cardBrand is added in the query parameters of Read Transaction List of Read Accound Data Request. And in Read Card Account Transaction List of Read Card Account Data Requests.	CR125

7.3 Changes from Version 2.3 to Version 2.4

The following changes have been applied in version 2.3 relative to version 2.4:

Section	Change	Reason
6	Updated some references, specifically the protocol function and security measures which contain a section on payment formats in the new version to be considered. This will lead to material changes in th API definitiions.	Clarification