THE Berlin GROUP

A EUROPEAN STANDARDS INITIATIVE

Joint Initiative on a PSD2 Compliant XS2A Interface

NextGenPSD2 XS2A Framework

Implementation Guidelines

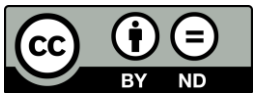
Version 1.3.8

30 October 2020

License Notice

This Specification has been prepared by the Participants of the Joint Initiative pan-European PSD2-Interface Interoperability* (hereafter: Joint Initiative). This Specification is published by the Berlin Group under the following license conditions:

- "Creative Commons Attribution-NoDerivatives 4.0 International Public License"

  This means that the Specification can be copied and redistributed in any medium or format for any purpose, even commercially, and when shared, that appropriate credit must be given, a link to the license must be provided, and indicated if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. In addition, if you remix, transform, or build upon the Specification, you may not distribute the modified Specification.

- Implementation of certain elements of this Specification may require licenses under third party intellectual property rights, including without limitation, patent rights. The Berlin Group or any contributor to the Specification is not, and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.
- The Specification, including technical data, may be subject to export or import regulations in different countries. Any user of the Specification agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import (parts of) the Specification.

* 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

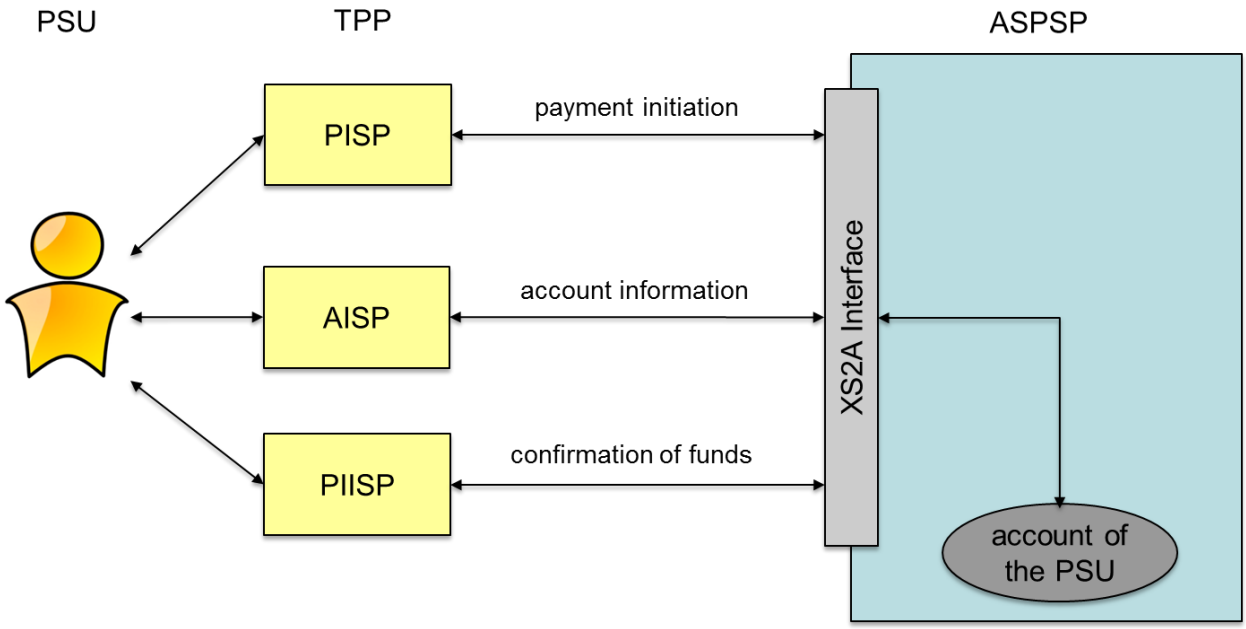
1.1 Background

With [PSD2] the European Union has published a new directive on payment services in the internal market. Member States had to adopt this directive into their national law until 13th of January 2018.

Among others [PSD2] contains regulations of new services to be operated by so called Third Party Payment Service Providers (TPP) on behalf of a Payment Service User (PSU). These new 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 of the Availability of Funds service to be used by Payment Instrument Issuing Service Provider (PIISP) TPP as defined by article 65 of [PSD2].

For operating the new services a TPP needs to access the account of the PSU which is usually managed by another PSP called the Account Servicing Payment Service Provider (ASPSP). As shown in the following figure, an ASPSP has to provide an interface (called "PSD2 compliant Access to Account Interface" or short "XS2A Interface") to its systems to be used by a TPP for necessary accesses regulated by [PSD2]:



Further requirements on the implementation and usage of this interface are defined by a Regulatory Technical Standard (short RTS) from the European Banking Authority (short EBA), published in the Official Journal of the European Commission.

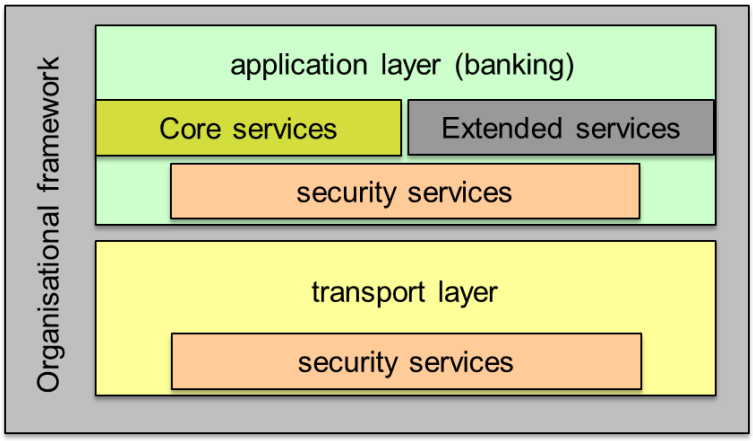
1.2 XS2A Interface Specification

This document is part of the NextGenPSD2 XS2A Specification which defines a standard for an XS2A Interface and by this reaching interoperability of the interfaces of ASPSPs at least for the core services defined by [PSD2]. An ASPSP may then use this standard as a basis for the implementation of its XS2A Interface to be compliant with PSD2.

The XS2A Interface is designed as a B2B interface between a TPP server and the ASPSP server. For time being, the protocol defined in this document is a pure client-server protocol, assuming the TPP server being the client, i.e. all API calls are initiated by the TPP. In future steps, this protocol might be extended to a server-server protocol, where also the ASPSP initiates API calls towards the TPP.

The Interoperability Framework defines operational rules, requirements on the data model and a process description in [XS2A-OR].

This document details the standard in defining messages and detailed data structures for the XS2A Interface. For the specification the two layers shown in the following figure are distinguished:



At the application layer only the core services will be specified in the first version of the framework. In addition the framework will be prepared such that the interface of an ASPSP may be extended with its own additional corporate specific services (included in the figure as "Extended services"). In future versions of this framework, some extended services will also be part of the standard. This framework documentation will point out extended services where the market need is already identified.

Using defined parameters different versions and variants of this protocol can be distinguished and implemented.

1.3 Structure of the Document

This document first outlines notations in Section 2 and requirements on the transport layer in Section 3. In Section 4, guiding principles for the definition of the XS2A interface and the API structure with API endpoints and permitted access methods are described. Section 5 then specifies in detail how a Payment Initiation Service Provider (PISP) can initiate payments within the Berlin Group XS2A. Section 6 then repeats this for the access of Account Information Service Provider (AISP) to a Payment Service User (PSU) account. The AIS and the PIS service are sharing potentially some API calls, specifically for authorising transactions directly through the TPP / ASPSP interface. These methods used within both services are specified in Section 7. Section 8 is introducing signing baskets as a new feature. These baskets allow to authorise several transactions at the same time, i.e. with one SCA. Section 9 then shortly explains how AIS and PIS services might be technically combined within one TPP / ASPSP business session.

The Confirmation of Funds Service for Payment Instrument Initiation Service Provider (PIISP) is specified in detail in Section 10. Following these chapters with functional character, Section 11 to Section 14 specify core payment structure, requirements on the optional integration of OAuth2, the usage of electronic seals for authentication on application level and general complex data structures.

Remark for Future: Please note that the Berlin Group NextGenPSD2 XS2A interface is still under constant development. Technical issues, which are already in discussion within the Berlin Group NextGenPSD2 working structure are mentioned in this document by "Remark for Future" to make the reader aware of upcoming potential changes.

1.4 Document History

Version	Change/Note	Approved
0.99	Market consultation draft of the Berlin Group XS2A Interface Framework	NextGenPSD2 Taskforce, 27 September 2017
1.0	Version 1.0 for publication. Takes into account the results of the market consultation and the final EBA-RTS on SCA and CSC.	NextGenPSD2 Taskforce, 08 February 2018
1.1	Minor release update, integrating results of convergence discussions with other API initiatives as well as some additional functionality and errata. A detailed change log will be published separately.	NextGenPSD2 Taskforce, 11 May 2018
1.2	Major release with the following major changes - support of multiple authorisations - support of payment cancellations - support of signing baskets instead of "pseudo-multi" payments as signing vehicle for multiple transactions - simplification in the /payments path: The resource will be addressable directly under /payments/paymentId even if posted on /payments/{payment-product} - the authorisation and cancellation-authorisation sub-resources have been separated from the payment, consent resp. signing basket resources and are exposed explicitly as resources in the API. In addition some minor functionality and errata have been considered. A detailed change log will be published separately.	NextGenPSD2 Taskforce, 25 July 2018
1.3	Payment Products are again involved in the path for starting authorisations, as it was	NextGenPSD2 Taskforce, 19 October 2018

Version	Change/Note	Approved
	foreseen till version 1.1 of this specification, cp. Bulletin No 001. More details on the usage for Multilevel SCA for Establish Account Information Consent Requests and Signing Baskets. Added a new endpoint for card accounts. Added a new additional variant to report additional error information following [RFC7807]. Clarifications and errata throughout the document. A change log document will be published separately.	
1.3.4	Errata on Version 1.3 Integration of Extended Services Added new headers to deal with payments which are neither rejected nor executed due to missing funds. Added a new functionality to update an authorisation resource by an additional password. A detailed change log document is published separately.	NextGenPSD2 Taskforce, 5 July 2019
1.3.5	Internal version	
1.3.6	Errata on Version 1.3.4 Integration of Extended Services on displaying account owner names and standing orders. Add new functionality on transaction confirmation as introduced by the security bulletin in Autumn 2019. Extend transaction report and payment initiation data models.	3 February 2020, NextGenPSD2 TF

Version	Change/Note	Approved
	Add new http headers e.g. for transporting TPP brand information for logging. A detailed change log document is published separately.	
1.3.7	Internal version	
1.3.8	Errata on Version 1.3.6 Integration of currency conversion fee information added to payment initiation to support conversion fee transparency requirements. Integration of a new SCA method for OTP transmission via email. Add clarifications to fulfil requirements resulting from the EBA Opinion on Obstacles from June 2020, cp. [EBA-OP2]. A detailed change log document is published separately.	30 October 2020

2 Character Sets and Notations

2.1 Character Set

The character set is UTF 8 encoded. This specification is only using the basic data elements "String", "Boolean", "ISODateTime", "ISODate", "UUID" and "Integer" (with a byte length of 32 bits) and ISO based code lists. For codes defined by ISO, a reference to the corresponding ISO standard is given in 14.34.

Max35Text, Max70Text, Max140Text and Max500Text are defining strings with a maximum length of 35, 70, 140 and 500 characters respectively.

ASPSPs will accept for strings at least the following character set:

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

/ - ? : ( ) . , ' +

Space

ASPSPs may accept further character sets for text fields like names, addresses, text. Corresponding information will be contained in the ASPSP documentation of the XS2A interface. ASPSPs might convert certain special characters of these further character sets, before forwarding e.g. submitted payment data.

2.2 Notation

2.2.1 Notation for Requests

For API request calls, query parameters, HTTP header parameters and body content parameters are specified within this specification as follows:

Attribute	Type	Condition	Description
attribute tag	type of attribute	condition	description of the semantic of the attribute and further conditions.

The following conditions may be used when describing data to be submitted by the client:

- Optional: The attribute is supported by the server, usage is optional for the client. The server may ignore the parameter if mentioned in the “Description” column of the table above.
- Conditional: The attribute is supported by the server and might be mandated by
  - the server provider in its own documentation of the support of this XS2A interface or
  - by certain rules as defined in the “Description” column of the table above.
- Mandatory: The attribute is supported by the server and shall be used by the client.
- Optional if supported by API provider: It is optional for the server to support this attribute. If the server is supporting the attribute as indicated in its own documentation of this XS2A interface, it might be used by the client optionally. If the server is not supporting the attribute, then the request is rejected when it is contained.

Remark: Please note that the conditions “Optional if supported by API provider” is used rarely in this specification.

2.2.2 Notation for Responses

For API call responses, parameters, HTTP header parameters and body content parameters are specified within this specification as follows:

Attribute	Type	Condition	Description
attribute tag	type of attribute	condition	description of the semantic of the attribute and further conditions.

The following conditions can be set on data to be provided by the server:

- Optional: The attribute is supported optionally by the server
- Conditional: The attribute is supported by the server under certain conditions as indicated in the “Description” column of the table above.
- Mandatory: The attribute is always supported by the server.

3 Transport Layer

The communication between the TPP and the ASPSP is always secured by using a TLS-connection using TLS version 1.2 or higher. For the choice of cipher suite selections, NIST recommendations on the cryptographical strength should be followed. For ASPSPs, further cipher suite requirements of their national IT security agency might apply.

This TLS-connection is set up by the TPP. It is not necessary to set up a new TLS-connection for each transaction, however the ASPSP might terminate an existing TLS-connection if required by its security setting.

The TLS-connection has to be established always including client (i.e. TPP) authentication. For this authentication the TPP has to use a qualified certificate for website authentication. This qualified certificate has to be issued by a qualified trust service provider according to the eIDAS regulation [eIDAS]. The content of the certificate has to be compliant with the requirements of [EBA-RTS]. The certificate of the TPP has to indicate all roles the TPP is authorised to use.

4 Application Layer: Guiding Principles

4.1 Location of Message Parameters

The XS2A Interface definition follows the REST service approach. This approach allows to transport message parameters at different levels:

- message parameters as part of the HTTP level (HTTP header)
- message parameters by defining the resource path (URL path information) with additional query parameters and
- message parameters as part of the HTTP body.

The content parameters in the corresponding HTTP body will be encoded either in JSON or in XML syntax. XML syntax is only used where

- an ISO 20022 based payment initiation (pain.001 message) with the corresponding payment initiation report (pain.002 message) or
- ISO 20022 based account information message (camt.052, camt.053 or camt.054 message)

is contained.

As an exception, response messages might contain plain text format in account information messages to support MT940, MT941 or MT942 message formats.

The parameters are encoded in

- in spinal-case (small letters) on path level,
- in Spinal-case (starting capital letters) on HTTP header level and
- in lowerCamelCase for query parameters and JSON based content parameters.

The following principle is applied when defining the API:

Message parameters as part of the HTTP header:

- Definition of the content syntax,
- Certificate and Signature Data where needed,
- PSU identification data (the actual data from the online banking frontend or access token),
- Protocol level data like Request Timestamps or Request/Transaction Identifiers

Message parameters as part of the path level:

- All data addressing a resource:
  - Provider identification,
  - Service identification,
  - Payment product identification,
  - Account Information subtype identification,
  - Resource ID

Query Parameters:

- Additional information needed to process the GET request for filtering information,

Message parameters as part of the HTTP body:

- Business data content,
- PSU authentication data,
- Messaging Information
- Hyperlinks to steer the full TPP – ASPSP process

4.2 Signing Messages at Application Layer

The ASPSP may require the TPP to sign request messages. This requirement shall be stated in the ASPSP documentation.

The signature shall be included in the HTTP header as defined by [signHTTP], chapter 4.

The electronic signature of the TPP has to be based on a qualified certificate for electronic seals. This qualified certificate has to be issued by a qualified trust service provider according to the eIDAS regulation [eIDAS]. The content of the certificate has to be compliant with the requirements of [EBA-RTS]. The certificate of the TPP has to indicate all roles the TPP is authorised to use.

This specification uses on a pure protocol level the following HTTP header in all HTTP requests uniformously for the support of the signature function:

Request Header

Attribute	Type	Condition	Description
Digest	String	Conditional	Is contained if and only if the "Signature" element is contained in the header of the request.
Signature	cp. Section 12	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
TPP-Signature-Certificate	String	Conditional	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained, see above.

For a better readability, the definition of these headers is not repeated throughout this specification. If no other condition describes otherwise, then the definitions here apply to all requests.

Remark: An ASPSP will ignore signatures on application level used by the TPP if signatures are not supported by the ASPSP.

4.3 Optional Usage of OAuth2 for PSU Authentication or Authorisation

The XS2A API will allow an ASPSP to implement OAuth2 as a support for the authorisation of the PSU towards the TPP for the payment initiation and/or account information service. In this case, the TPP will be the client, the PSU the resource owner and the ASPSP will be the resource server in the abstract OAuth2 model.

This specification supports two ways of integrating OAuth2. The first support is an authentication of a PSU in a pre-step, translating this authentication into an access token to be used at the XS2A interface afterwards. This usage of OAuth2 will be referred to in this specification as "if OAuth2 has been used as PSU authentication". Further details shall be defined in the documentation of the ASPSP of this XS2A interface.

Remark: When implementing the OAuth pre-step, the requirements on e.g. registration steps or no mandatory two SCA usage in specific PIS only scenarios as defined by [EBA-OP2] should be recognised by the ASPSP.

The second option to integrate OAuth2 is an integration as an OAuth2 SCA Approach to be used for authorisation of payment initiations and consents. In both services, PIS and AIS, OAuth2 will in this option be used in an integrated way, by using the following steps:

Integrated OAuth in the Use Case SCA for PIS:

1.) The payment data is posted to the corresponding payment initiation endpoint of the XS2A API.
2.) The OAuth2 protocol is used with the "Authorisation Code Grant" flow to get the consent on the payment authorised by the PSU, while using the "scope" attribute in OAuth2 to refer to the data from Step 1.).
3.) The corresponding payment is then automatically initiated by the ASPSP after a successful authorisation by the PSU.

Integrated OAuth in the Use Case SCA for AIS:

1.) The AIS consent data is posted to the consents endpoint of the XS2A API.
2.) The OAuth2 protocol is used with the "Authorisation Code Grant" flow to get the consent on the payment resp. the AIS access authorised by the PSU, while using the "scope" attribute in OAuth2 to refer to the data from Step 1.).
3.) The TPP can use the access token received during the OAuth2 protocol to access the /accounts endpoint for authorised account information for the validity period of the authorised consent resp. the validity period of the technical access token.

For Step 2.), details are described in Section 13.

When using OAuth2, the XS2A API calls will work with an access token instead of using the PSU credentials.

4.4 XS2A Interface API Structure

The XS2A Interface is resource oriented. Resources can be addressed under the API endpoints

```
https://{provider}/v1/{service}{?query-parameters}
```

using additional content parameters {parameters}

where

- {provider} is the host and path of the XS2A API, which is not further mentioned. The host or path may contain release version information of the ASPSP.
- v1 is denoting the final version 1.3 of the Berlin Group XS2A interface Implementation Guidelines.

  Remark for Future: The handling of implementation and specification release version information is planned to be adapted in a more standardized way in future versions of the specification.
- {service} has the values consents, payments, bulk-payments, periodic-payments, accounts, card-accounts, signing-baskets or funds-confirmations, eventually extended by more information on product types and request scope
- {?query-parameters} are parameters detailing GET based access methods, e.g. for filtering content data
- {parameters} are content attributes defined in JSON or XML encoding according to the following
  - XML encoding appears only when ISO 20022 pain.001 messages are transported when demanded by the ASPSP for the corresponding payment product
  - all other request bodies are encoded in JSON

The structure of the request/response is described according to the following categories

- Path: Attributes encoded in the Path, e.g. "payments/sepa-credit-transfers" for {resource}
- Query Parameters: Attributes added to the path after the "?" sign as process steering flags or filtering attributes for GET access methods. Query parameters of type Boolean shall always be used in a form query-parameter=true or query-parameter=false.
- Header: Attributes encoded in the HTTP header of request or response

- Request: Attributes within the content parameter set of the request
- Response: Attributes within the content parameter set of the response, defined in XML, text or JSON:
  - XML encoding appears only, when camt.052, camt.053 or camt.054 messages (reports, notifications or account statements) or pain.002 payment status messages are transported. pain.002 messages will only be delivered for the GET Status Request, and only in cases where the payment initiation was performed by using pain.001 messages.
  - Text encoding appears only, when MT940, MT941 or MT942 messages (reports, notifications or account statements) are transported.
  - All other response bodies are encoded in JSON.

The HTTP response codes which might be used in this XS2A interface are specified in Section 14.11. This is not repeated for every API call definition.

> **Remark:** For JSON based responses, this specification defines body attributes which are responded from ASPSP to TPP following POST or PUT API calls. The ASPSP is free to return the whole addressed resource within the response, following usual REST methodologies.

4.5 Multicurrency Accounts

Definition: A multicurrency account is an account which is a collection of different sub-accounts which are all addressed by the same account identifier like an IBAN by e.g. payment initiating parties. The sub-accounts are legally different accounts and they all differ in their currency, balances and transactions. An account identifier like an IBAN together with a currency always addresses uniquely a sub-account of a multicurrency account.

This specification supports to address multicurrency accounts either on collection or on sub-account level. The currency data attribute in the corresponding data structure "Account Reference" allows to build structures like

```
{"iban": "DE40100100103307118608"}
```

or

```
{"iban": "DE40100100103307118608",
 "currency": "EUR"}
```

If the underlying account is a multicurrency account, then

- the first reference is referring to the collection of all sub-accounts addressable by this IBAN, and
- the second reference is referring to the euro sub-account only.

This interface specification is acting on sub-accounts of multicurrency accounts in exactly the same way as on regular accounts. This applies to payment initiation as well as to account information.

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.

4.6 Authorisation Endpoints

The NextGenPSD2 API is supporting dedicated authorisation endpoints for payment initiation transactions and establish consent transactions in order to handle transaction authorisation by PSUs. These authorisation endpoints are supported from version 1.2 of this specification for supporting the following new features in a common structured way

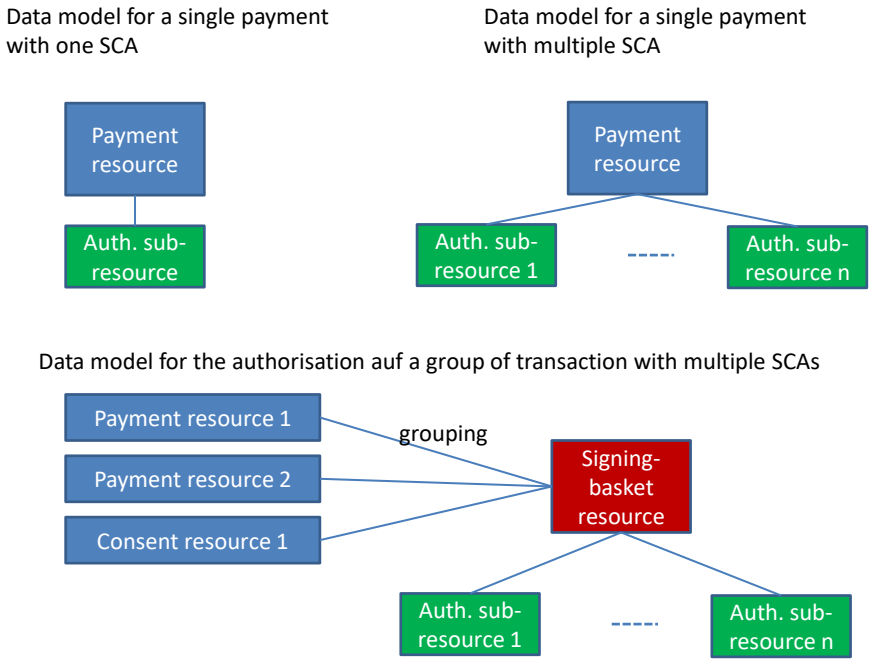
- multiple level SCA, where a transaction needs an authorisation by more than one PSU, e.g. in a corporate context,
- signing of a group of transactions with one SCA, as it is offered by ASPSPs today in online banking,
- signing of a group of transactions with multi-level SCA, where this group of transactions need an authorisation by more than one PSU, e.g. in a corporate context.

To support this, the resources resulting from the submission of payment data or consent data are separated from authorisation (sub-)resources. A payment which needs to be signed *n* times then will end up in a payment resource with *n* SCA (sub-)resources in a normal successful process.

Remark: This new resource structure also applies to the authorisation of individual transactions, which is a major change to the data model supported in version 1.0 and 1.1 of this specification. Nevertheless, the optimised integration of the authorisation process into the payment initiation or establish consent process will still be supported, cp. the paragraph at the end of this section.

The optional function of grouping several transactions for one common authorisation process is supported by the signing-baskets endpoint, which might be offered by the ASPSP. If this function is offered by an ASPSP, the TPP can first submit payment and consent data without starting the authorisation. After having grouped the related payment and consent resources by using a grouping command through the signing-baskets endpoint, the authorisation then can be started by authorising this basket content. This results in a basket resource with the corresponding authorisation sub-resource.

The following picture gives an overview on the abstract data model for the different scenarios:



Remark: When offering the signing basket function, the ASPSP might restrict the grouping e.g.

- to payments as such,
- to individual payments,
- to the same payment product.

This restriction on groupings will then be detailed in the ASPSPs documentation.

Note: The grouping of transaction is only a "signing vehicle", bundling authorisation processes for the grouped transactions. The authorisation rules for transactions can be very complex in a corporate context. The signing basket gets the status of being fully authorised as soon as all grouped transactions have been successfully authorised by the applied SCA mechanism. A transaction with less authorisation requirements might then be authorised earlier than the

whole signing basket and also already processed. In addition, single transactions of the signing basket could be authorised with additional SCAs directly on transaction level, depending on the implementations of the ASPSPs – the signing basket is a non-exclusive mechanism to bundle authorisations. Current implementations of this functionality differ in Europe, specifically in a corporate context. For this reason, more complex functionality as DELETE processes on partially authorised signing baskets are not supported yet.

Remark for Future: The upcoming versions of the specification might implement more advanced functionality of the signing basket function and cancellation processes around it.

Optimisation process for the submission of single payments

The general model introduced above requires the TPP to start two sub-processes when initiating a payment, creating a signing basket or submitting a consent. In a payment initiation of a sepa credit transfer this would result in

```
POST /payments/sepa-credit-transfers {payment data}
```

which is generating the payment resource and returns paymentId as a resource identification.

```
POST /payments/sepa-credit-transfers/paymentId/authorisations
```

is then starting the authorisation process with creating an authorisation sub-resource and returning an authorisationId for addressing this sub-resource in the following.

Applying this requirement to all authorisations of transactions e.g. in the Redirect SCA Approach would significantly augment the calls on the resulting API. For this reason, this specification still enables the ASPSP to directly start e.g. a Redirect SCA processing after the submission of a payment or a consent, if no other data from the TPP has to be submitted anyhow. In this case, the ASPSP will create the related authorisation sub-resources automatically and will give access to these sub-resources to the TPP by returning corresponding hyperlinks, cp. Section 4.15. As a consequence, the authorisation status would still result by submitting the command

```
GET /payments/sepa-credit-
transfers/paymentId/authorisations/authorisationId,
```

where the authorisation resource with identification authorisationId has been created by the ASPSP implicitly.

4.7 Payment Cancellation Endpoints

Starting from version 1.2, this specification is supporting the cancellation of payment initiations by PISPs. This process is divided into two steps

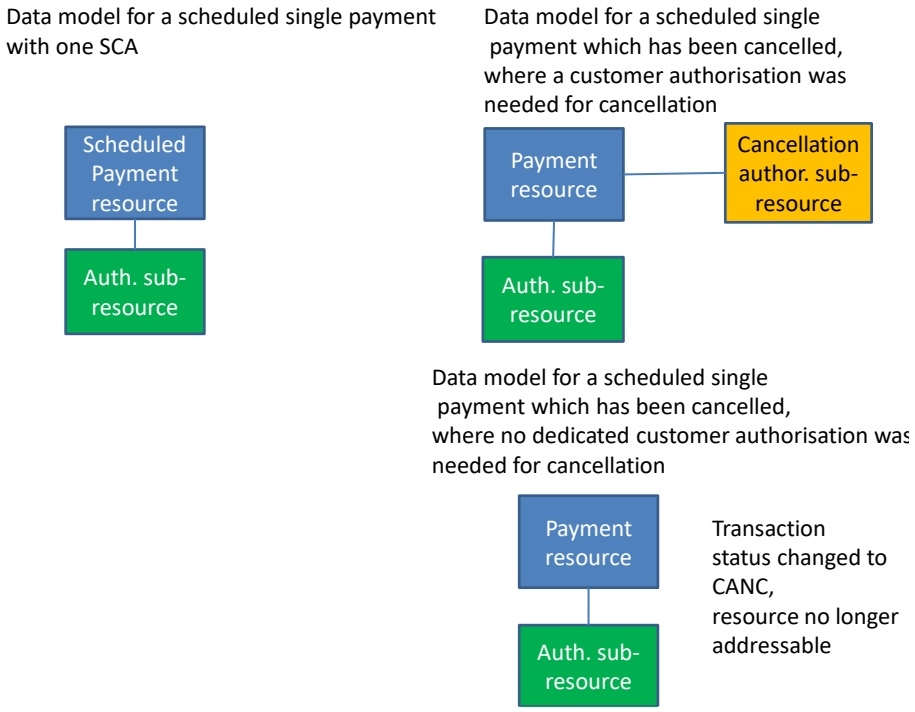
1. DELETE the corresponding resource.
2. Start an authorisation process for the cancellation by the PSU where needed by submitting a

   ```
   POST payments/sepa-credit-transfers/paymentId/cancellation-
                                authorisations
   ```

   command.

The second step might be omitted, where a dedicated authorisation of the cancellation is not foreseen by the ASPSP. The need to authorise the cancellation will be communicated by sending corresponding hyperlinks to the TPPs, cp. Section 4.15.

In the two-step approach, this cancellation process will be handled by cancellation-authorisation sub-resources in analogy to the actual authorisations. The authorisation sub resources will stay unchanged. The following picture shows the changes on resource level in case of a scheduled payment:



The corresponding original authorisation sub-resources stay unchanged.

For transactions, where a multilevel SCA is needed for authorisation, also a multilevel SCA might be needed for cancellation, depending on ASPSP role management. In equivalence to authorisation, the model would then be extended by more cancellation sub-resources.

4.8 Requirements on PSU Context Data

The following data elements are forwarding information about the PSU-TPP interface and are enhancing the risk management procedures of the ASPSP. It is strongly recommended to send

these data elements in all request messages within the payment initiation or consent initiation transaction flow, i.e. flows with a PSU authentication involved. The further definitions of request parameters within the related sections are not repeating the definition of these elements for the matter of better readability. The only exception is where conditions other than "optional" apply on specific request messages, e.g. for the PSU IP Address. More details are provided in the data overview in within Section 5.2 or Section 6.2.

Attribute	Format	Condition	Description
PSU-IP-Address	String	Optional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP.
PSU-IP-Port	String	Optional	The forwarded IP Port header field consists of the corresponding HTTP request IP Port field between PSU and TPP, if available.
PSU-Accept	String	Optional	The forwarded IP Accept header fields consist of the corresponding HTTP request Accept header fields between PSU and TPP, if available.
PSU-Accept-Charset	String	Optional	see above
PSU-Accept-Encoding	String	Optional	see above
PSU-Accept-Language	String	Optional	see above
PSU-User-Agent	String	Optional	The forwarded Agent header field of the HTTP request between PSU and TPP, if available.
PSU-Http-Method	String	Optional	HTTP method used at the PSU – TPP interface, if available. Valid values are: • GET • POST • PUT • PATCH • DELETE
PSU-Device-ID	String	Optional	UUID (Universally Unique Identifier) for a device, which is used by the PSU, if available.

Attribute	Format	Condition	Description
			UUID identifies either a device or a device dependant application installation. In case of an installation identification this ID need to be unaltered until removal from device.
PSU-Geo-Location	Geo Location	Optional	The forwarded Geo Location of the corresponding HTTP request between PSU and TPP if available.

Note: Information about the PSU/TPP interface might be used by the ASPSP as input for his fraud detection and risk management systems. Some ASPSPs use this information also to exclude some authentication methods (for example some ASPSPs do not allow to receive an OTP by SMS on the same smartphone used also for the transaction itself). In addition the ASPSP might need to receive specific device related information to be able to support an optimised app-2-app redirection procedure for the TPP. For these reasons it is highly recommended that a TPP includes all of this information into the related request messages. Missing information may result in an assessment of the user device as not useable for the authentication method or in a classification of the current transaction as a "higher risk transaction" e.g. due to session attacs. By this the probability of a rejection of that transaction due to the result of fraud detection and/or risk management might be increased, cp [XS2A-SecB] for details.

4.9 Requirements on TPP Identification

[PSD2] is mandating the identification of TPPs by PSD2 related eIDAS certificates. The specific certificates to be used within the PSD2 context are specified within [ETSI PSD2]. The requirements defined in [XS2A-OR] yield a TPP identification by the QWAC and/or QSEAL certificate used by the TPP.

The TPP is noted in the eIDAS certificate by its legal name. Still, the TPP might use brand names towards the PSU, which are differing from the legal name strongly. Thus, it might be beneficial for the TPP if the ASPSP is able to use also TPP brand names towards the PSU in all PSU related processes like SCA. Specific brand names of the TPP could be entered into the certificate field Organisation Unit (marked with the tag “OU”). ASPSP may ignore entries in this field.

Remark: The usage of the certificate field “OU” by the TPP will lead to the usage of several certificates if the TPP intends to separate different TPP brands in processing.

Note: The usage of more than one certificate by the TPP, differing e.g. by different OU entries does not mean that the consent management is treating these different certificates as different entities. By default of this framework, the legal owner of the certificate is the counterparty for access managements through consent tokens and not the brand cited in the OU field.

4.10 Requirements on TPP URIs

The TPP can provide several URIs to the ASPSP as parameters for succeeding protocol steps. For security reasons, it should be ensured that these URIs are secured by the TPP eIDAS QWAC used for identification of the TPP. The following applies:

URIs which are provided by TPPs in TPP-Redirect-URI or TPP-Nok-Redirect-URI should comply with the domain secured by the eIDAS QWAC certificate of the TPP in the field CN or SubjectAltName of the certificate. Please note that in case of example-TPP.com as certificate entry TPP-Redirect-URI like

- www.example-TPP.com/xs2a-client/v1/ASPSPidentifcation/mytransaction-id or
- redirections.example-TPP.com/xs2a-client/v1/ASPSPidentifcation/mytransaction-id

would be compliant.

Wildcard definitions shall be taken into account for compliance checks by the ASPSP.

Remark for Future: ASPSPs in future may reject requests, if the provided URIs do not comply. This is not yet valid for the current version of the specification.

Remark for Future: For migration reasons, this specification mandates the TPP to keep the TPP-Redirect-URI used within all authorisation processes for a specific transaction during the lifecycle of this transaction constant. This might be removed in the next version of the specification.

Remark for Future: The restrictions on URIs will apply to TPP-URIs used within future Push Services of the ASPSP.

4.11 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 /v1/consents/{consentId} applies only if the endpoint /v1/consents is supported at all.

Please note that 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.

4.11.1 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 5.3.1 and 5.3.2
payments/{payment-product}/{paymentId}	GET	Mandatory	Read the details of an initiated payment. Section 5.5
payments/{payment-product}/{paymentId}/status	GET	Mandatory	Read the transaction status of the payment Section 5.4

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 5.3.3
bulk-payments/{payment-product}/{paymentId}	GET	Mandatory	Read the details of an initiated bulk payment. Section 5.5
bulk-payments/{payment-product}/{paymentId}/status	GET	Mandatory	Read the transaction status of the bulk payment Section 5.4
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 5.3.4
periodic-payments/{payment-product}/{paymentId}	GET	Mandatory	Read the details of an initiated standing order for recurring/periodic payments. Section 5.5
periodic-payments/{payment-product}/{paymentId}/status	GET	Mandatory	Read the transaction status of the standing order for recurring/periodic payments. Section 5.4

Endpoints/Resources	Method	Condition	Description
{payment-service}/{payment-product}/{paymentId}/authorisations	POST	Mandatory	Create an authorisation sub-resource and start the authorisation process, might in addition transmit authentication and authorisation related data. This method is iterated n times for a n times SCA authorisation in a corporate context, each creating an own authorisation sub-endpoint for the corresponding PSU authorising the transaction. The ASPSP might make the usage of this access method unnecessary in case of only one SCA process needed, since the related authorisation resource might be automatically created by the ASPSP after the submission of the payment data with the first POST payments/{payment-product} call. Section 7.1
{payment-service}/{payment-product}/{paymentId}/authorisations	GET	Mandatory	Read a list of all authorisation sub-resources IDs which have been created. Section 7.4
{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}	PUT	Mandatory for Embedded SCA Approach, Conditional for other approaches	Update data on the authorisation resource if needed. It may authorise a payment within the Embedded SCA Approach where needed. Independently from the SCA Approach it supports e.g. the selection of the authentication method and a non-SCA PSU authentication. Section 7.2 and Section 7.3

Endpoints/Resources	Method	Condition	Description
{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}	GET	Mandatory	Read the SCA status of the authorisation. Section 7.5
{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 5.6
{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations	POST	Optional	Starts the authorisation of the cancellation of the addressed payment with resource identification paymentId if mandated by the ASPSP (i.e. the DELETE access method is not sufficient) and if applicable to the payment-service, 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). Section 7.1

Endpoints/Resources	Method	Condition	Description
{payment-service}{payment-product}/{paymentId}/cancellation-authorisations	GET	Optional	Retrieve a list of all created cancellation authorisation sub-resources. If the POST command on this endpoint is supported, then also this GET method needs to be supported. Section 5.7
{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations/{authorisationId}	PUT	Mandatory for Embedded SCA Approach, Conditional for other approaches	Update data on the cancellation authorisation resource if needed. It may authorise a cancellation of the payment within the Embedded SCA Approach where needed. Independently from the SCA Approach it supports e.g. the selection of the authentication method and a non-SCA PSU authentication. Section 7.2 and Section 7.3
{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations/{authorisationId}	GET	Mandatory	Read the SCA status of the cancellation authorisation. Section 7.5

4.11.2 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

Endpoints/Resources	Method	Condition	Description
			resources are provided if a related consent has been already granted. **Remark:** Note that the /consents endpoint optionally offers to grant an access on **all available** payment accounts of a PSU. In this case, this endpoint will deliver the information about all available payment accounts of the PSU at this ASPSP. Section 6.5.1
accounts?withBalance	GET	Optional	Read the identifiers of the available payment account together with booking balance information, depending on the consent granted Section 6.5.1
accounts/{account-id}	GET	Mandatory	Give detailed information about the addressed account. Section 6.5.2
accounts/{account-id}?withBalance	GET	Optional	Give detailed information about the addressed account together with balance information Section 6.5.2
accounts/{account-id}/balances	GET	Mandatory	Give detailed balance information about the addressed account Section 6.5.3
accounts/{account-id}/transactions	GET	Mandatory	Read transaction reports or 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.

Endpoints/Resources	Method	Condition	Description
			Section 6.5.4
accounts/{account-id}/transactions?withBalance	GET	Optional	Read transaction reports or transaction lists of a given account, depending on the steering parameter "bookingStatus" together with balances. Section 6.5.4
accounts/{account-id}/transactions/{transactionId}	GET	Optional	Read transaction details of an addressed transaction. Section 6.5.5

Remark: Note that the {account-id} parameters 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.

4.11.3 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 6.6.1
card-accounts/{account-id}	GET	Optional	Give detailed information about the addressed card account. Section 6.6.2 **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 6.6.3 **Remark for Future**: This endpoint might be made mandatory for future versions of the specification.
card-accounts/{account-id}/transactions	GET	Mandatory	Read transaction reports or transaction lists related to a given card account. For a given card account, additional parameters are

Endpoints/Resources	Method	Condition	Description
			e.g. the attributes "dateFrom" and "dateTo". Section 6.6.4

Remark: Note that the {card-account-id} parameters can be tokenised 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.

4.11.4 Consents Endpoint

Endpoints/Resources	Method	Condition	Description
consents	POST	Mandatory	Create a consent resource, defining access rights to dedicated accounts of a given PSU-ID. These accounts are addressed explicitly in the method as parameters as a core function. Section 6.3.1
consents	POST	Optional	As an option, an ASPSP might optionally accept a specific access right on the access on all psd2 related services for all available accounts. As another option an ASPSP might optionally also accept a command, where only access rights are inserted without mentioning the addressed account. The relation to accounts is then handled afterwards between PSU and ASPSP. This option is not supported for the Embedded SCA Approach. As a last option, an ASPSP might in addition accept a command with access rights • to see the list of available payment accounts or

Endpoints/Resources	Method	Condition	Description
			• to see the list of available payment accounts with balances. Section 6.3.1
consents/{consentId}	GET	Mandatory	Reads the exact definition of the given consent resource {consentId} including the validity status Section 6.3.3
	DELETE	Mandatory	Terminate the addressed consent. Section 6.4
consents/{consentId}/status	GET	Mandatory	Read the consent status of the addressed consent resource. Section 6.3.2
consents/{consentId}/authorisations	POST	Mandatory	Create an authorisation sub-resource and start the authorisation process, might in addition transmit authentication and authorisation related data. The ASPSP might make the usage of this access method unnecessary, since the related authorisation resource will be automatically created by the ASPSP after the submission of the consent data with the first POST consents call. Section 7.1
consents/{consentId}/authorisations	GET	Mandatory	Read a list of all authorisation sub-resources IDs which have been created. Section 7.4

Endpoints/Resources	Method	Condition	Description
consents/{consentId}/authorisations/{authorisationId}	PUT	Mandatory for Embedded SCA Approach, Conditional for other approaches	Update data on the authorisation resource if needed. It may authorise a consent within the Embedded SCA Approach where needed. Independently from the SCA Approach it supports e.g. the selection of the authentication method and a non-SCA PSU authentication. Section 7.2 and Section 7.3
consents/{consentId}/authorisations/{authorisationId}	GET	Mandatory	Read the SCA status of the authorisation. Section 7.5

4.11.5 Signing-baskets Endpoint

Endpoints/Resources	Method	Condition	Description
signing-baskets	POST	Optional	Create a signing basket resource for authorising several transactions with one SCA method. The resource identifications of these transactions are contained in the payload of this access method Section 8.1
signing-baskets/{basketId}	GET	Optional	Retrieve the signing basket content Section 8.2
	DELETE	Optional	Delete the signing basket structure as long as no authorisation has yet been applied. The underying transactions are not affected by this deletion. Section 8.5
signing-baskets/{basketId}/status	GET	Optional	Read the status of the signing basket

Endpoints/Resources	Method	Condition	Description
			Section 8.3
signing-baskets/{basketId}/authorisations	POST	Mandatory	Create an authorisation sub-resource and start the authorisation process, might in addition transmit authentication and authorisation related data. The ASPSP might make the usage of this access method unnecessary, since the related authorisation resource will be automatically created by the ASPSP after the submission of the basket data with the first POST consents call. Section 7.1
signing-baskets/{basketId}/authorisations/{authorisationId}	PUT	Mandatory for Embedded SCA Approach, Conditional for other approaches	Update data on the authorisation resource if needed. It may authorise all transactions in the addressed signing basket within the Embedded SCA Approach where needed. Independently from the SCA Approach it supports e.g. the selection of the authentication method and a non-SCA PSU authentication. Section 7.2 and Section 7.3
signing-baskets/{basketId}/authorisations/{authorisationId}	GET	Mandatory	Read the SCA status of the authorisation. Section 7.5

Remark: The signing basket as such is not deletable after a first authorisation has been applied. Nevertheless, single transactions might be cancelled on an individual basis on the XS2A interface.

4.11.6 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 10.2

Remark for Future: The PUT HTTP methods might be adapted to technical PATCH methods in a future version of the specification. A corresponding decision will reflect current market practices and the work in ISO TC68/SC9/WG2 on Financial API services.

4.12 HTTP Response Codes

The HTTP response code is communicating the success or failure of a TPP request message, cp. [RFC7231]. The 4XX HTTP response codes should only be given if the current request cannot be fulfilled, e.g. a payment initiation cannot be posted or account transactions cannot be retrieved. A request to get the status of an existing payment or a consent usually returns HTTP response code 200 since the actual request to retrieve the status succeeded, regardless if that payment or consent state is set to failure or not.

This specification supports the following HTTP response codes:

Status Code	Description
200 OK	PUT, GET Response Codes This return code is permitted if a request was repeated due to a time-out. The response in that might be either a 200 or 201 code depending on the ASPSP implementation. The POST for a Funds request will also return 200 since it does not create a new resource. DELETE Response Code where a payment resource has been cancelled successfully and no further cancellation authorisation is required.
201 Created	POST response code where Payment Initiation or Consent Request was correctly performed.

Status Code	Description
202 Accepted	DELETE response code, where a payment resource can be cancelled in general, but where a cancellation authorisation is needed in addition.
204 No Content	DELETE response code where a consent resource was successfully deleted. The code indicates that the request was performed, but no content was returned.
400 Bad Request	Validation error occurred. This code will cover malformed syntax in request or incorrect data in payload.
401 Unauthorized	The TPP or the PSU is not correctly authorized to perform the request. Retry the request with correct authentication information.
403 Forbidden	Returned if the resource that was referenced in the path exists but cannot be accessed by the TPP or the PSU. This code should only be used for non-sensitive id references as it will reveal that the resource exists even though it cannot be accessed.
404 Not found	Returned if the resource or endpoint that was referenced in the path does not exist or cannot be referenced by the TPP or the PSU. When in doubt if a specific id in the path is sensitive or not, use the HTTP response code 404 instead of the HTTP response code 403.
405 Method Not Allowed	This code is only sent when the HTTP method (PUT, POST, DELETE, GET etc.) is not supported on a specific endpoint. It has nothing to do with the consent, payment or account information data model. DELETE Response code in case of cancellation of a payment initiation, where the payment initiation cannot be cancelled due to legal or other operational reasons.
406 Not Acceptable	The ASPSP cannot generate the content that the TPP specified in the Accept header.
408 Request Timeout	The server is still working correctly, but an individual request has timed out.
409 Conflict	The request could not be completed due to a conflict with the current state of the target resource.
415 Unsupported Media Type	The TPP has supplied a media type which the ASPSP does not support.

Status Code	Description
429 Too Many Requests	The TPP has exceeded the number of requests allowed by the consent or by the RTS.
500 Internal Server Error	Internal server error occurred.
503 Service Unavailable	The ASPSP server is currently unavailable. Generally, this is a temporary state.

4.13 Additional Error Information

If necessary, the ASPSP **might** communicate additional error information to the TPP within a request/response dialogue which results in 4xx or 5xx HTTP response codes. This specification offers two possibilities for ASPSPs to communicate additional error information. The ASPSP might choose one of the solutions. Note that the major additional error information is the detailed error code which is of type "Message Code" as defined in Section 14.11 is used in both variants of additional error information.

In cases, where no message code is defined for an HTTP response code in Section 14.11, the additional error information is not used, since the messageCode is a mandatory subfield. In this case, the HTTP code gives sufficient information about the error situation.

4.13.1 NextGenPSD2 Specific Solution

The NextGenPSD2 XS2A specification offers a proprietary way to transport additional error information. In this solution, the additional error information is sent to the TPP using the data element tppMessageInformation with the attribute category set to "ERROR". The attribute "code" indicates the error, cp. Section 14.11 and if applicable the path of the element of the request message which provoked this error message. It will further offer a free text field to describe the error context or actions to be taken to the TPP.

In addition, the response message might optionally contain a _links section containing a hyperlink to tell the TPP the next step to avoid further errors, cp. Section 4.15. This applies especially in case of PSU authentication errors where a resubmission of credentials by the TPP might be needed after new entering of credentials by the PSU.

Response Code

The HTTP response code is 4xx or 5xx as defined in Section 4.12 for response codes in case of errors.

Response Header

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	The string application/json is used.

Response Body

Attribute	Type	Condition	Description
tppMessages	Array of TPP Message Information	Optional	Error information
_links	Links	Optional	Should refer to next steps if the problem can be resolved e.g. for re-submission of credentials.

Example 1 (Access token not correct):

```
{ "tppMessages": [{
        "category": "ERROR",
        "code": "TOKEN_INVALID",
        "text": "additional text information of the ASPSP up to 500
characters"
   }]
}
```

Example 2 (Password incorrect):

```
{ "tppMessages": [{
        "category": "ERROR",
        "code": "PSU_CREDENTIALS_INVALID",
        "text": "additional text information of the ASPSP up to 500
characters"
   }],
  "_links": {
        "updatePsuAuthentication": {"href": "/v1/payments/sepa-credit-
transfers/1234-wertiq-983/authorisations/123auth456"}
           }
}
```

4.13.2 Standardised Additional Error Information

In [RFC7807], a standardised definition of reporting error information is described. In the following, requirements of how to use this standardised error information reporting in the context of the NextGenPSD2 XS2A interface are defined.

Response Code

The HTTP response code is 4xx or 5xx as defined in Section 4.12 for response codes in case of errors.

Response Header

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	The string application/problem+json is used.

Response Body

Attribute	Type	Condition	Description
type	Max70Text	Mandatory	A URI reference [RFC3986] that identifies the problem type. **Remark for Future**: These URI will be provided by NextGenPSD2 in future.
title	Max70Text	Optional	Short human readable description of error type. Could be in local language. To be provided by ASPSPs.
detail	Max500Text	Optional	Detailed human readable text specific to this instance of the error. XPath might be used to point to the issue generating the error in addition. **Remark for Future**: In future, a dedicated field might be introduced for the XPath.
code	Message Code	Mandatory	Message code to explain the nature of the underlying error.
additionalErrors	Array of Error Information	Optional	Might be used if more than one error is to be communicated

Attribute	Type	Condition	Description
_links	Links	Optional	Should refer to next steps if the problem can be resolved e.g. for re-submission of credentials.

Example

```
  HTTP/1.1 401 Unauthorized
  Content-Type: application/problem+json
  Content-Language: en
   {
    "type": "https://berlingroup.com/error-codes/TOKEN_INVALID",
    "title": " The OAuth2 token is associated to the TPP but is not valid
for the addressed service/resource.",
   "detail": " additional text information of the ASPSP up to 500
characters ",
    "code": "TOKEN_INVALID",
     "additionalErrors": [ {
            "title": "The PSU-Corporate-ID cannot be matched by the
addressed ASPSP.",
            "detail": "additional text information of the ASPSP up to 500
characters",
            "code": "CORPORATE_ID_INVALID"
    },… ],
    "_links": { }
   }
```

4.14 Status Information

4.14.1 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 with in 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 including funds availability 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.

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.

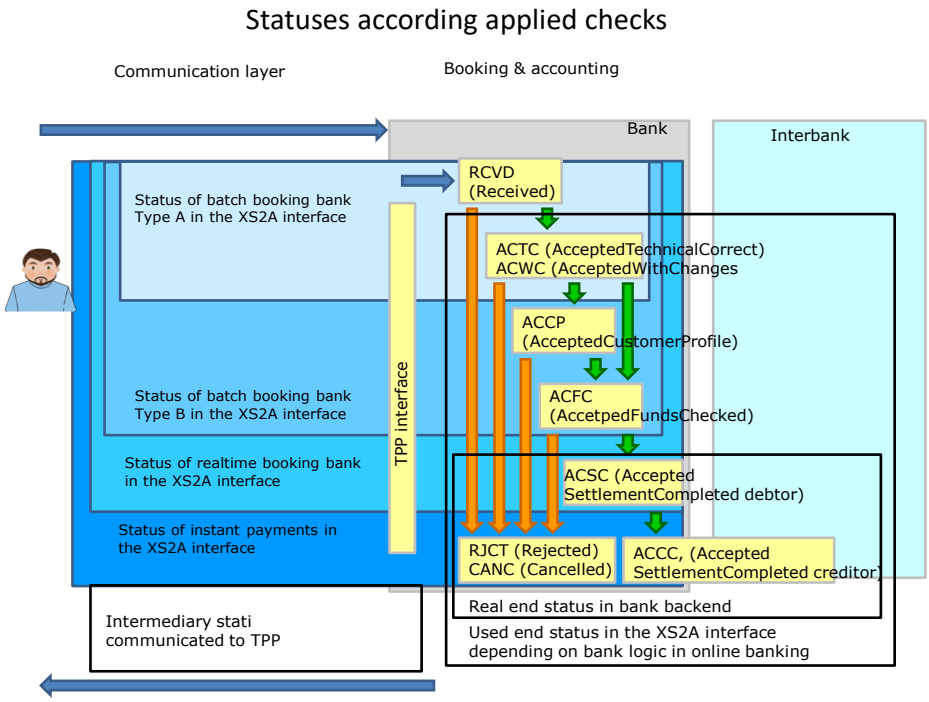
In addition, the ASPSP will inform the TPP about the status of the technical SCA process for a payment initiation within the GET SCA Status Response Message. For this status reporting the data element “scaStatus” is used.

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:



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.

Remark: This code is not yet part of the ISO20022 transaction status external reason code. The Berlin Group will raise a corresponding change request.

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 new status "PATC" for "PartiallyAcceptedTechnicalCorrect".

4.14.2 Status Information for the AIS within the Establish Consent Process

The status of the consent resource is changing during the initiation process as well as the transaction status of a payment initiation resource. In difference to the payment initiation process, there are only SCA checks on the consent resource and no feedback loop with the ASPSP backend. The data element for the status of the consent is defined as "consentStatus".

The only codes within the **initiation phase** supported for the consentStatus for this process are "received", “rejected” and “valid”. The current status of the consent resource is returned within all response messages during the authorisation process of the consent.

After a successful authorisation of a consent by a PSU, the consent resource might change its status during its lifecycle which needs to be transparent to the AIS. The following codes are supported during the **lifecycle phase** of the consent:

- "expired": The consent has been expired (e.g. after 90 days).
- "revokedByPsu": The consent has been revoked by the PSU.
- "terminatedByTpp": The AIS has terminated the consent.

The AIS can retrieve this status within the GET Status Response Message.

Note: The "expired" status will also apply to one off consents, once they are used or out dated.

Note: The "terminatedByTpp" status will also apply, when a recurring has been terminated in case of a side effect by the same TPP establishing a new consent for the same PSU.

In addition, the ASPSP informs the TPP about the status of the technical SCA process for establishing a consent within the GET SCA Status Response Message. For this status reporting the data element “scaStatus” will be used.

4.15 API Steering Process by Hyperlinks

The XS2A API requires for the payment initiation and account information service several requests from the TPP towards the ASPSP. With the Payment Initiation Request and the Account Information Consent Request, a resource presentation is generated by the ASPSP. The location header of the response will usually contain a link to the created resource.

In addition, the ASPSP can embed a hyperlink together with a "tag" for the semantics of this hyperlink into the response to these first requests and to all succeeding requests within the services. This hyperlink then can be either a relative link, which is recommend to save space, for the host starting e.g. with "/v1/payments/sepa-credit-transfers" or it can be a global link like https://www.testbank.com/psd2/authentication/v1/payments/transaction/asdf-asdf-asdf-1234.

The global links might be needed in some circumstances, e.g. a re-direct. The tag of the hyperlink transports the functionality of the resource addressed by the link, e.g. "authorise-transaction". This link indicates that results of a SCA method are to be posted to the resource addressed by this link to authorise e.g. a payment.

The steering hyperlinks are transported in the "_links" data element, cp. [HAL]. It may contain one or several hyperlinks.

The "_links" data element may contain more hyperlinks than specified in the related call. In this case, this will be documented in the ASPSP's PSD2 documentation or the hyperlinks can be ignored by the TPP.

In Section 14.6, the list of supported hyperlink types is defined.

Some hyperlinks might require additional data in the same response body which are then needed when following this hyperlink. The following table gives an overview on these specific steering hyperlinks to explain interconnection with the data elements.

Hyperlink	Additional Link Related Data	Description
startAuthorisationWith PsuAuthentication	(challengeData)	The link to an endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where PSU authentication data shall be uploaded with the corresponding call. Remark: In rare cases the ASPSP will ask only for some dedicated ciphers of the passwords. This information is then transported to the TPP by using the "challenge" data element, normally used only in SCA context.
startAuthorisationWith EncryptedPsuAuthentication	(challengeData)	Same as startAuthorisactionWith PsuAuthentication, but password is encrypted on application layer when uploaded.
updatePsuAuthentication	(challengeData)	The link to the payment initiation/consent resource, which needs to be updated by a PSU password and eventually the PSU identification if not delivered yet. **Remark**: In rare cases the ASPSP will ask only for some dedicated ciphers of the passwords. This information is then transported to the TPP by using the "challenge" data element, normally used only in SCA context.
updateEncryptedPsu Authentication	(challengeData)	Same as updatePsuAuthentication, but password is encrypted on application layer when uploaded.

Hyperlink	Additional Link Related Data	Description
startAuthorisationWith AuthenticationMethodSelection	scaMethods	This is a link to and endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where the selected SCA method shall be uploaded with the corresponding call.
selectAuthenticationMethod	scaMethods	This is a link to a resource, where the TPP can select the applicable strong customer authentication methods for the PSU, if there were several available authentication methods.
authoriseTransaction	challengeData, chosenScaMethod	A link to the resource, where a "Transaction Authorisation Request" can be sent to. This request transports the result of the SCA method performed by the customer, generating a response to the challenge data.
startAuthorisationWith TransactionAuthorisation	challengeData, chosenSCAMethod	A link to an endpoint, where an authorisation of a transaction or a cancellation can be started, and where the response data for the challenge is uploaded in the same call for the transaction authorisation or transaction cancellation at the same time in the Embedded SCA Approach.

4.16 Data Extensions

The ASPSP might add more data attributes to response messages. Such extensions then shall be documented in the ASPSP's documentation of its XS2A interface. These data attributes can be either ignored by the TPP or can be interpreted as defined by the above mentioned documentation.

The ASPSP might add additional optional data attributes to be submitted, e.g. for setting up additional services. In addition, an ASPSP can ask the TPP for a submission of proprietary

data in a second step via the "proprietaryData" hyperlink. This shall be published by the ASPSP in its documentation.

> **Remark**: Before defining these additional proprietary data elements, the ASPSP is requested to submit the attribute description to the Berlin Group NextGen Taskforce, where it will be decided on a standardised approach for the related data attributes.

5 Payment Initiation Service

Remark: The API design differs across the various SCA approaches (Embedded, Redirect, OAuth2 or Decoupled, cp. [XS2A-OR]), 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.

5.1 Payment Initiation Flows

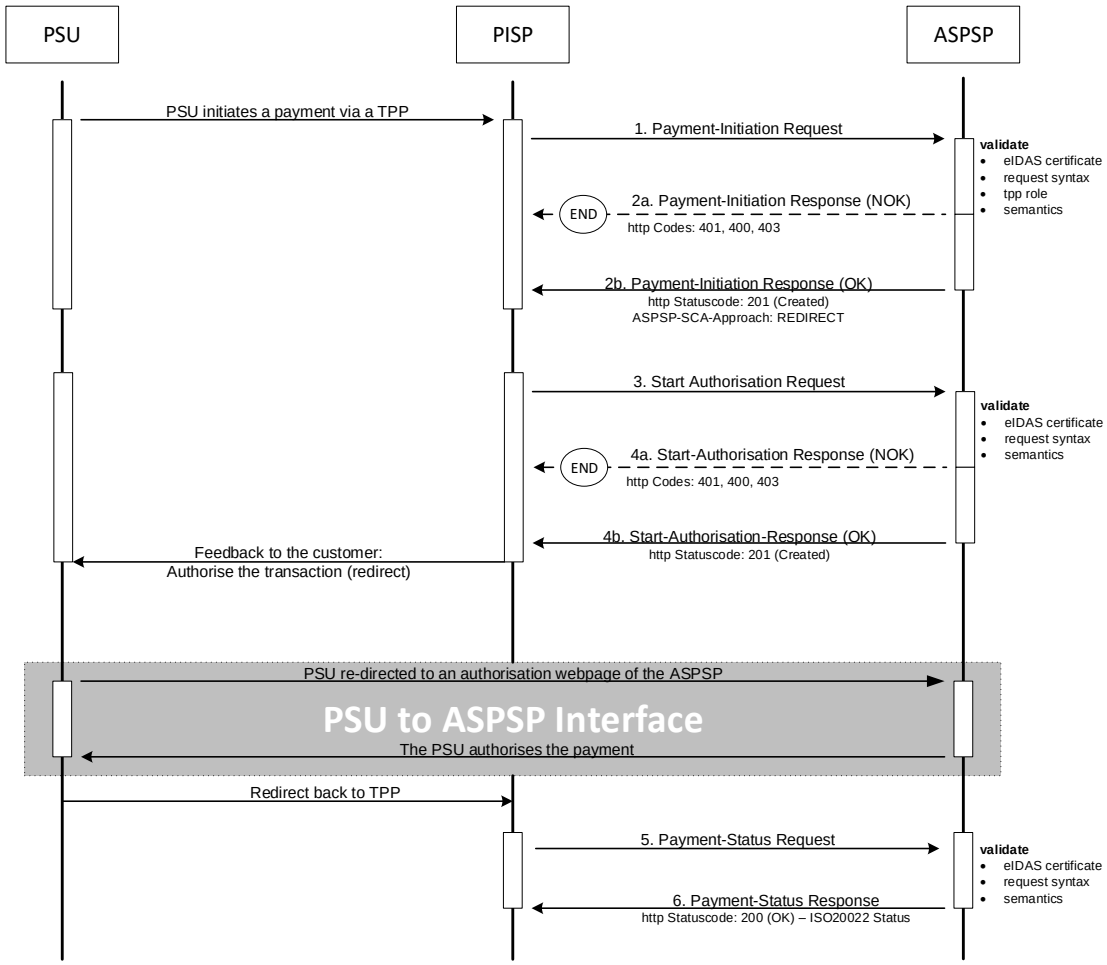
The payment initiation flow depends heavily on the SCA approach implemented by the ASPSP. The most complex flow is the flow for the Embedded SCA Approach, which further differs on whether there are various authentication methods available for the PSU. In the following, the different API flows are provided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

5.1.1 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

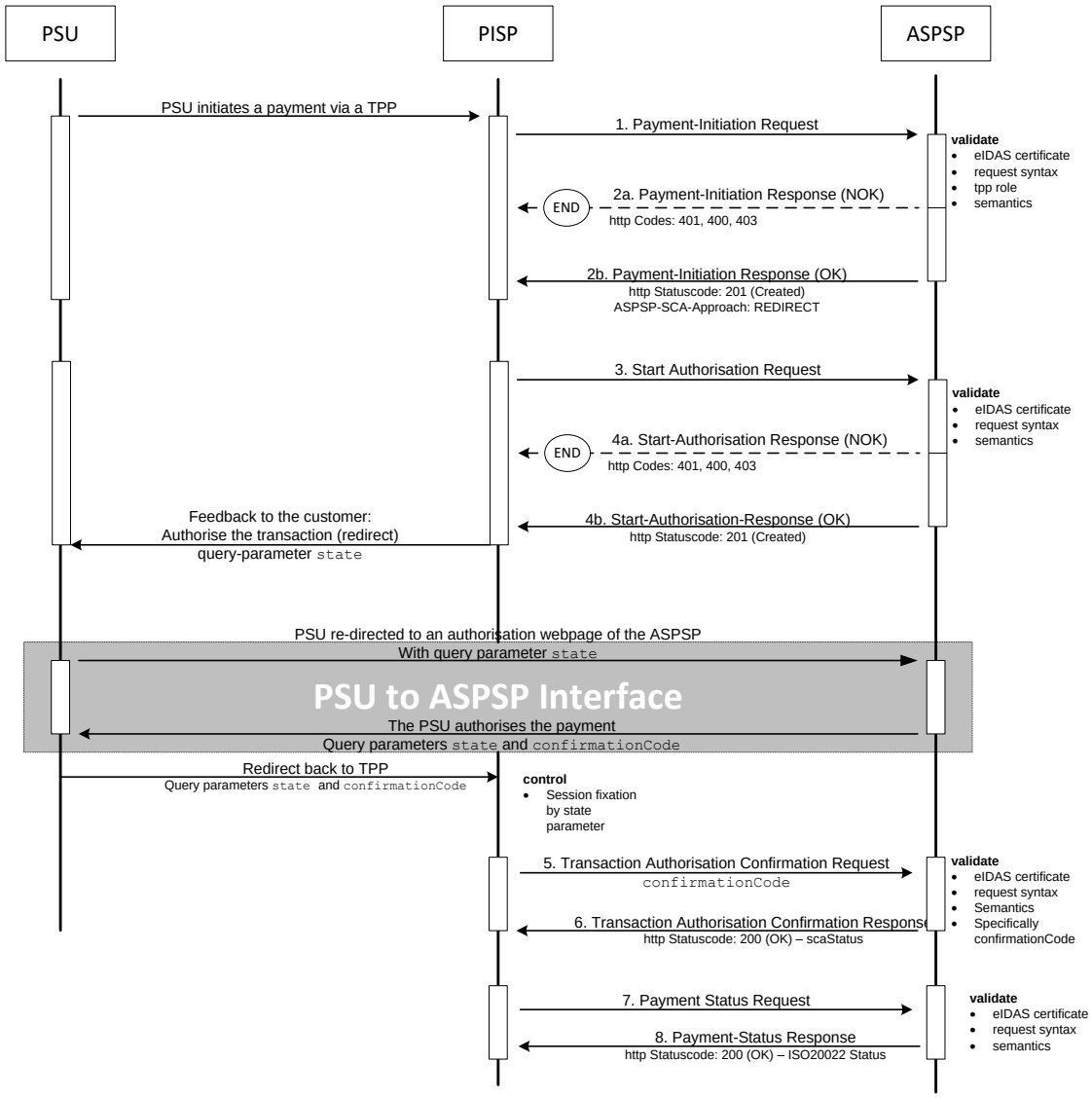
authorisation site. A status request might be requested by the TPP after the session is re-redirected to the TPP's system.



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

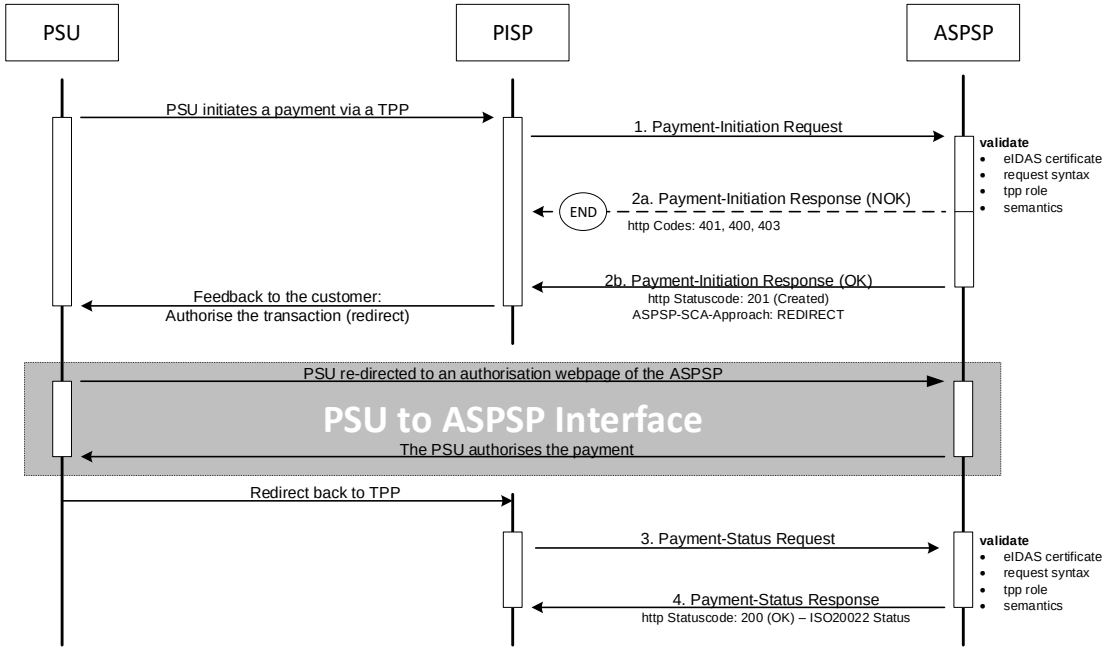
In addition to the scenario above, an authorisation confirmation request might be requested by the ASPSP from the TPP after the session is re-redirected to the TPP's system and after the

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.



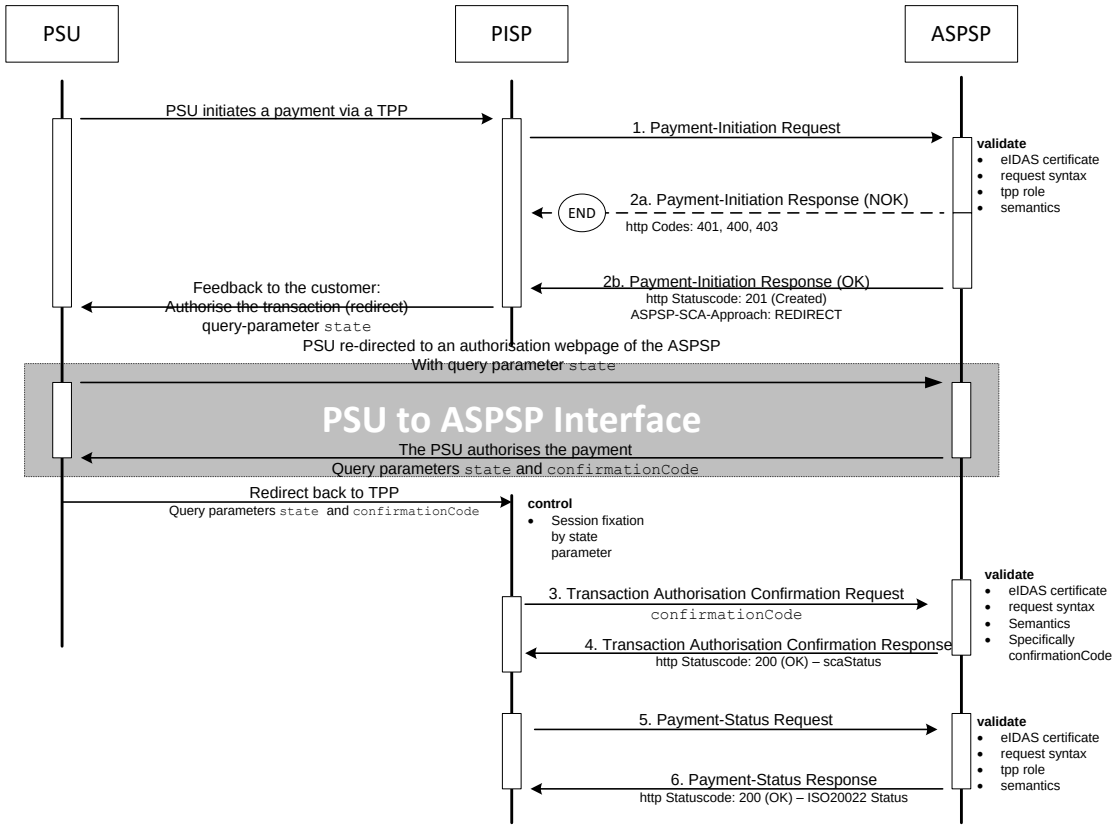
5.1.3 Redirect SCA Approach: Implicit Start of the Authorisation Process

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).



5.1.4 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.

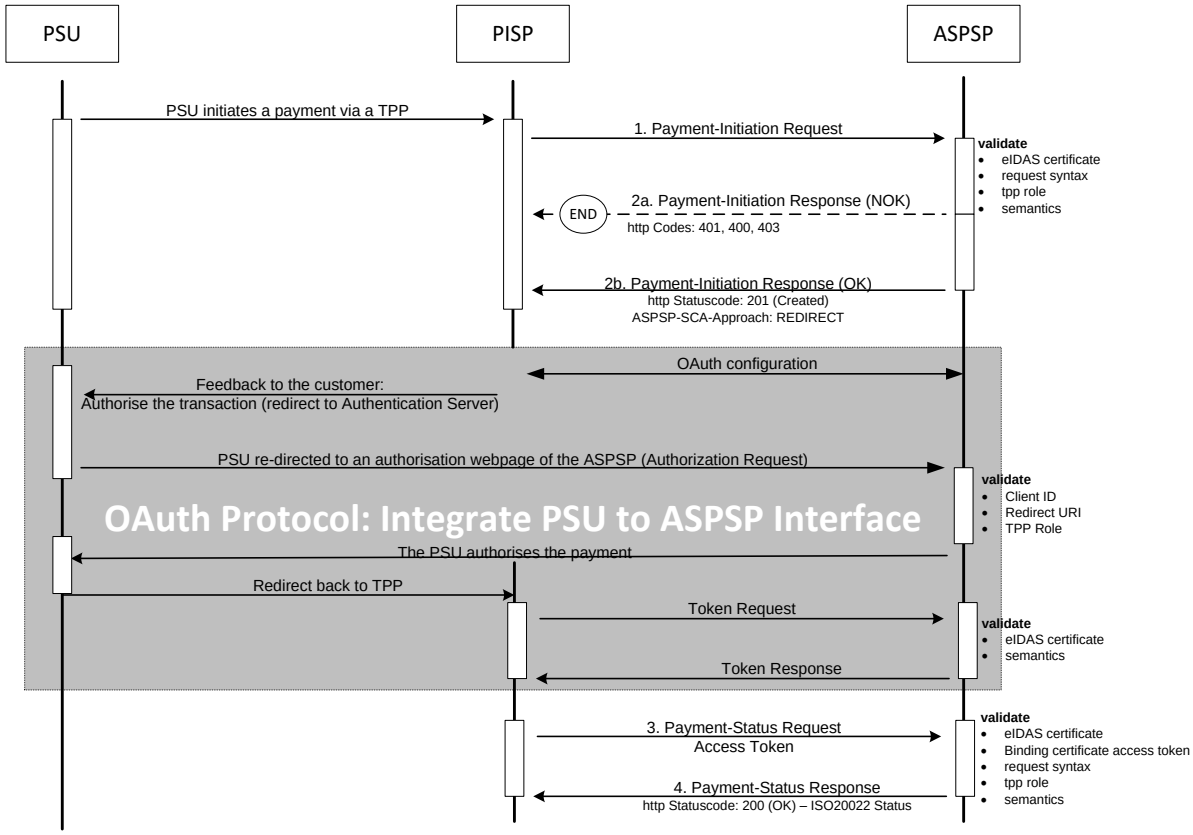


5.1.5 OAuth2 SCA Approach: Implicit Start of the Authorisation Process

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.

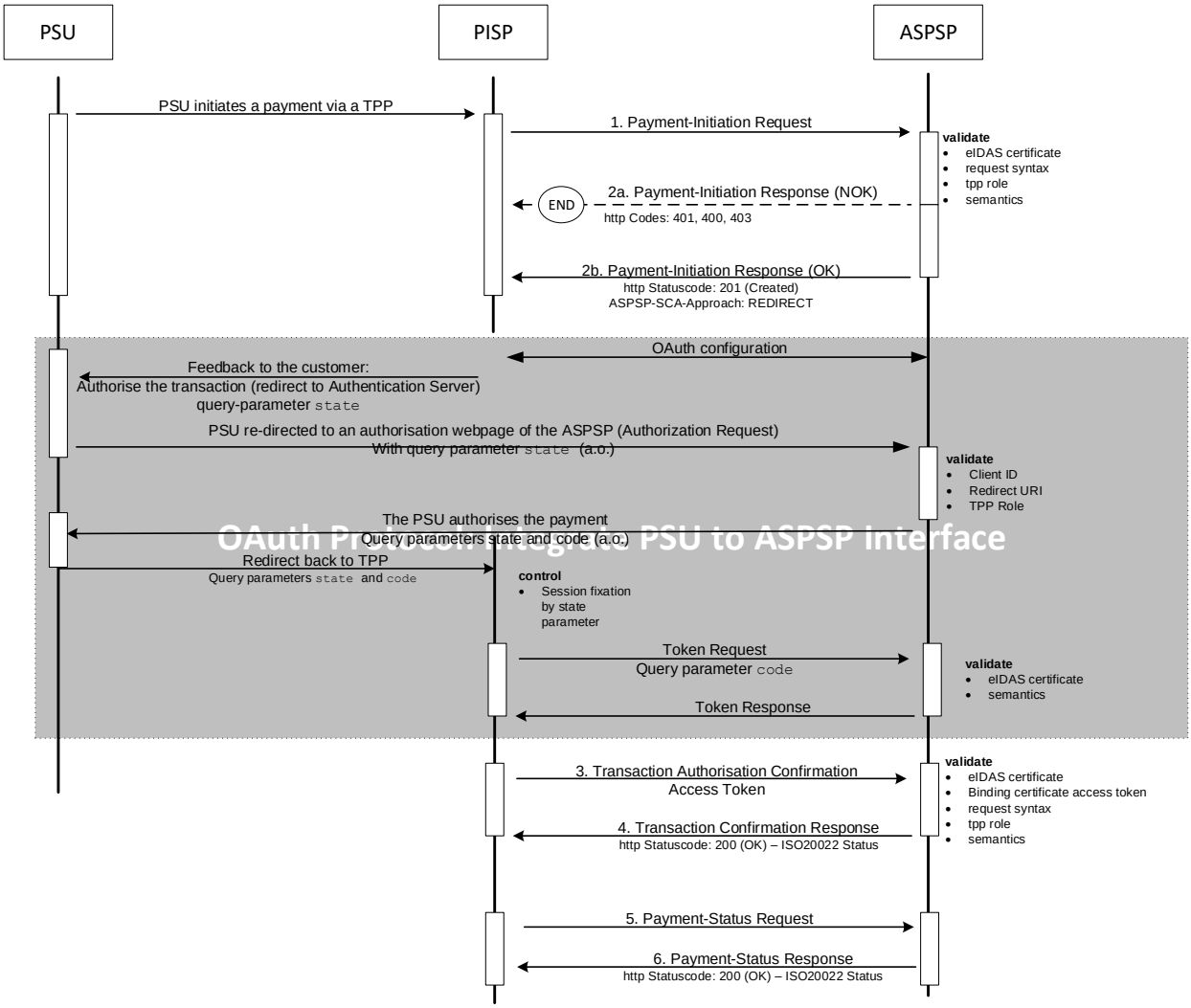


5.1.6 OAuth2 SCA Approach: Implicit Start of the Authorisation Process with Confirmation Code

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.

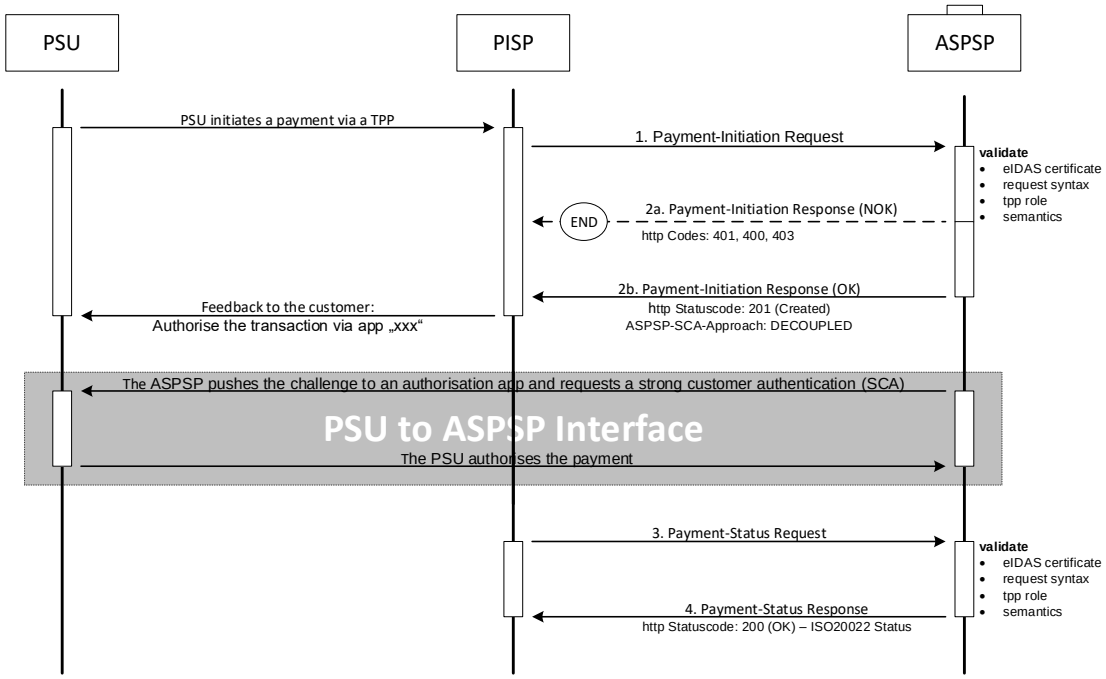


It is further recommended for ASPSPs and TPPs in this case to follow the Security Best Practice definitions as defined in [OA-SecTop]. This reference will also be added in the next version of the Implementation Guidelines.

5.1.7 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:



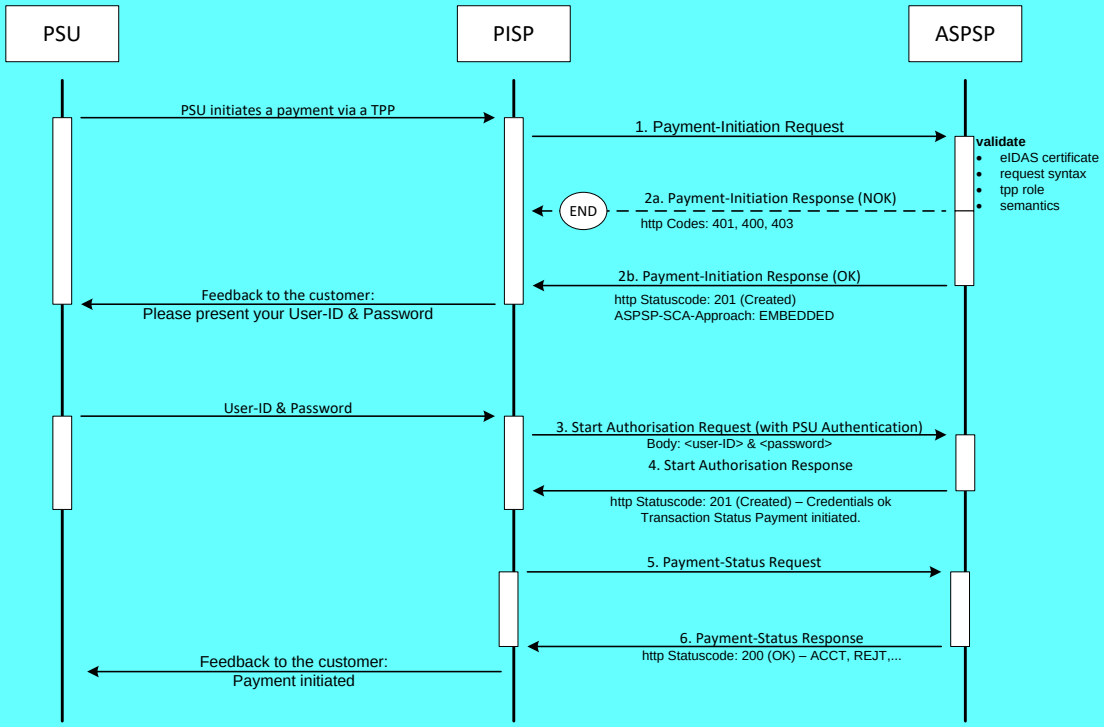
Remark: In Section 6.1.1.3, a version with the explicit start of the Authorisation Process is documented for the Establish Consent Request.

5.1.8 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

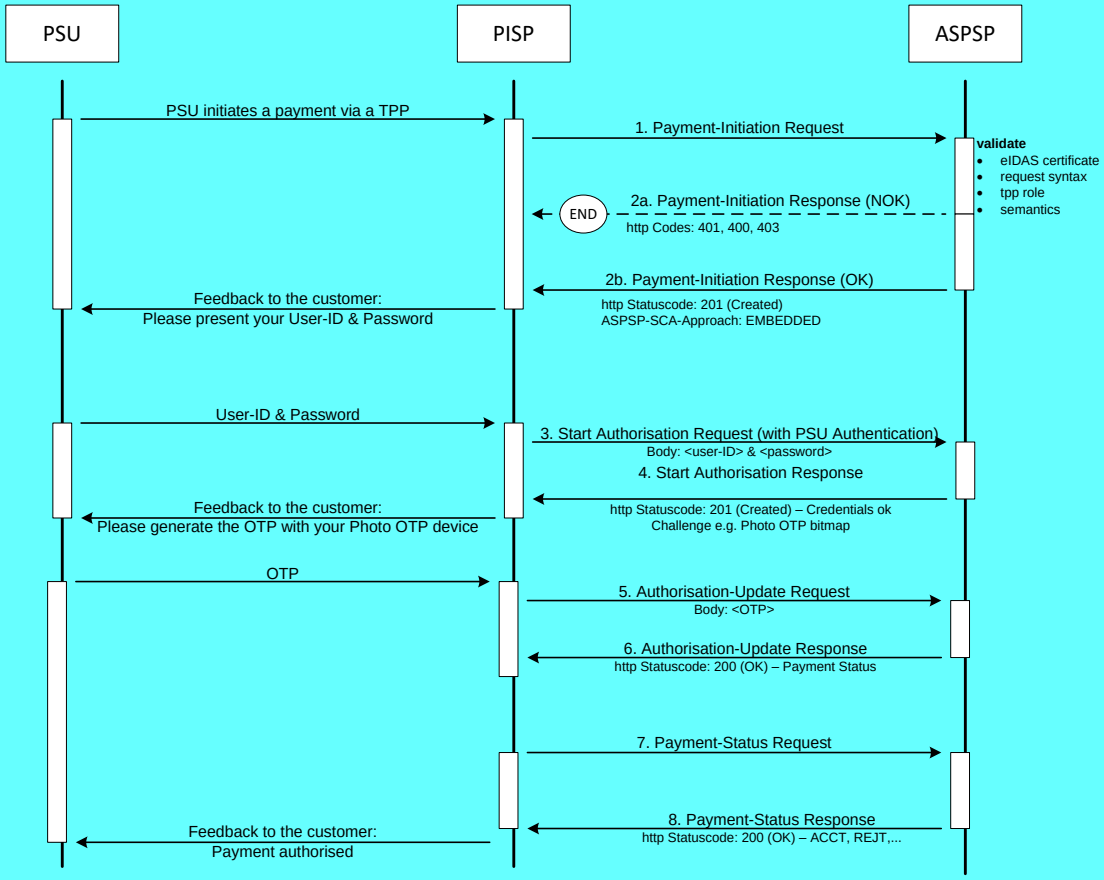
scenarios, where complexities like SCA processing and choosing an SCA method will be added.

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.



5.1.9 Embedded SCA Approach with only one SCA method available

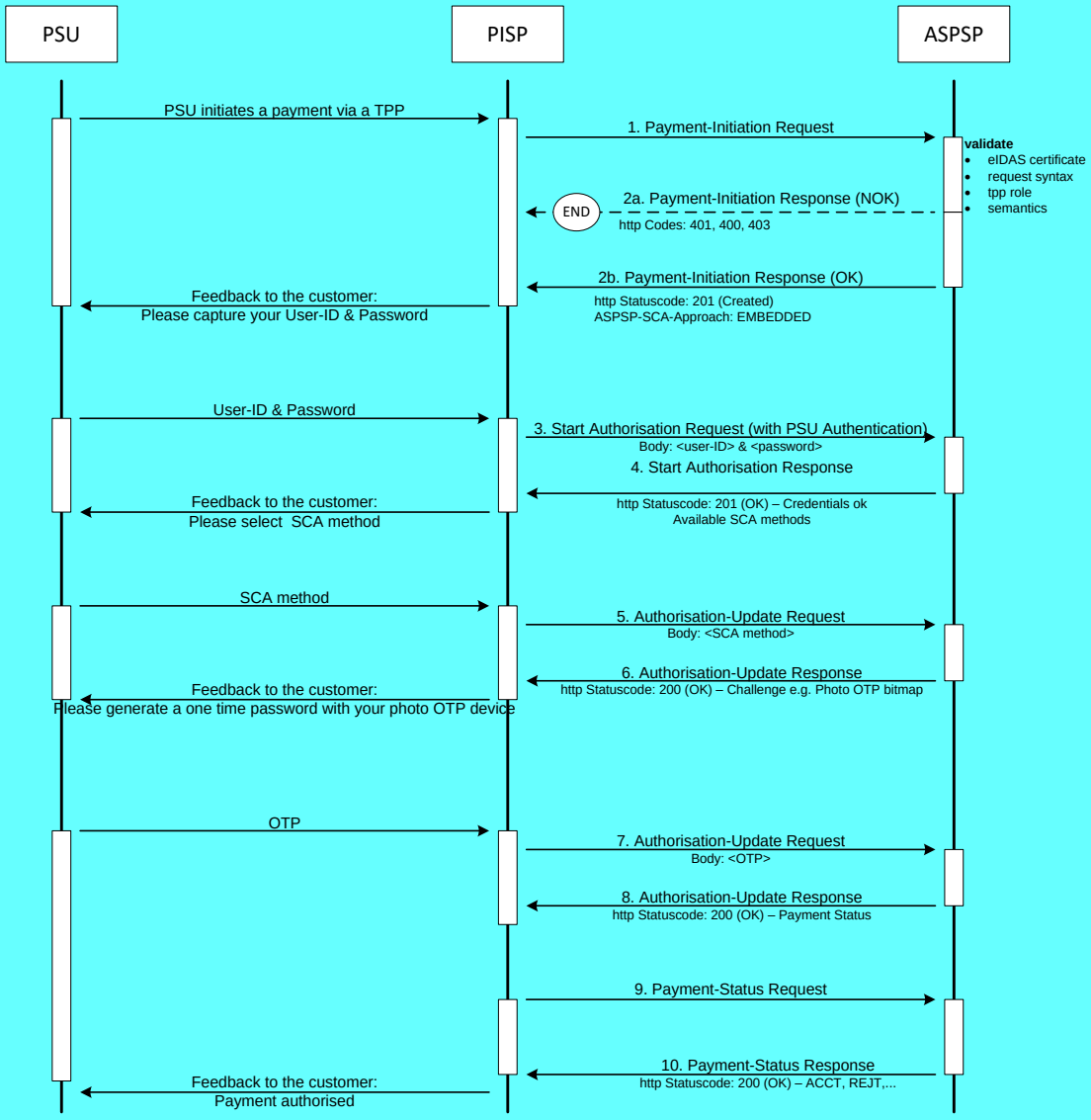
In case where only one SCA method is available, the "Authorise Transaction Request" is added to the flow, where the TPP is transmitting the authentication data of the customer, e.g. an OTP with included dynamic linking to the transaction details.



5.1.10 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.



5.1.11 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.

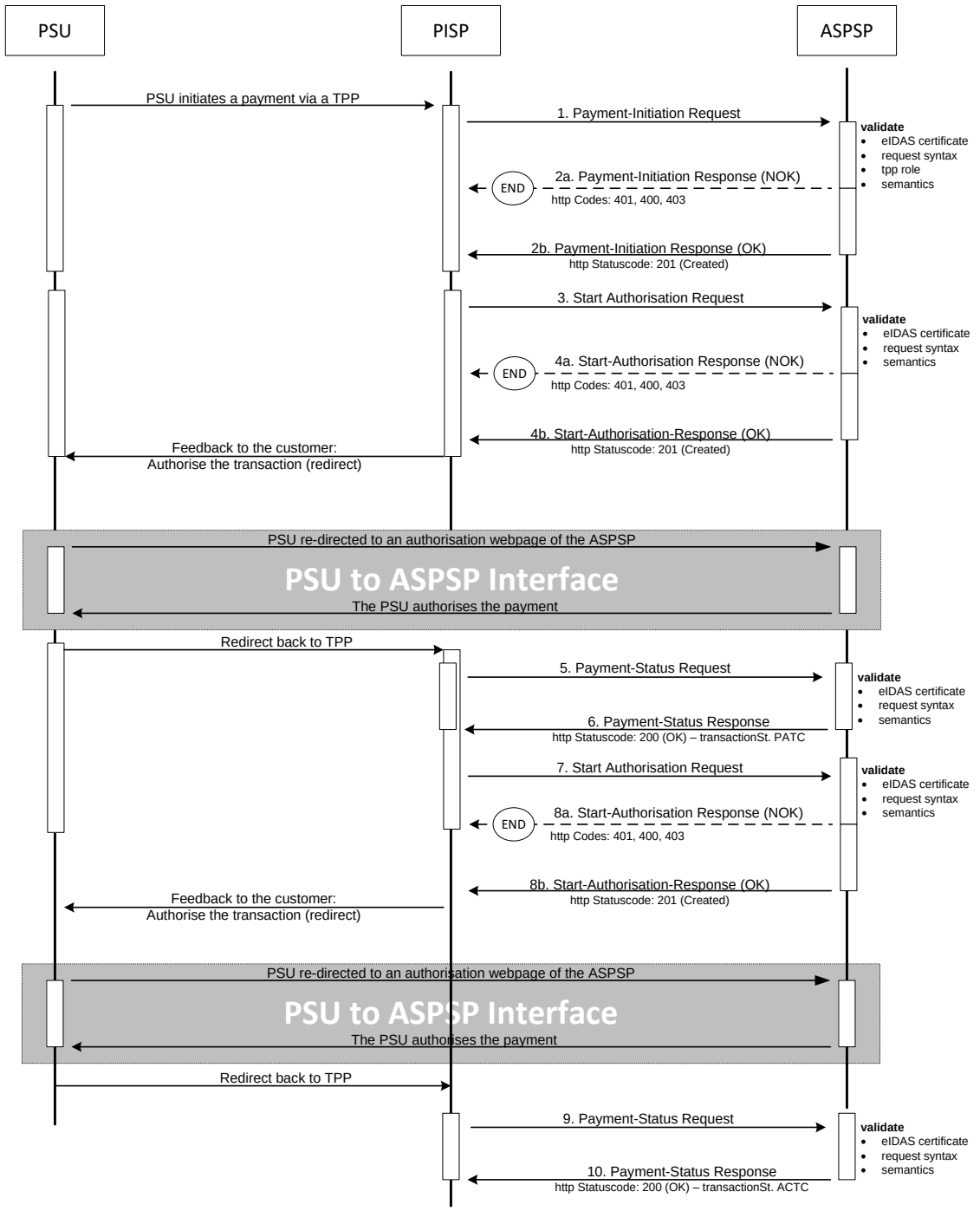
In case the ASPSP is starting the payment initiation flow with the Embedded SCA Approach the ASPSP will provide a list of available SCA methods to the PSU via the TPP. If the PSU chooses an authentication method which requires the Decoupled SCA Approach, then the ASPSP is branching into the transaction 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.

5.1.12 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.



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.

5.2 Data Overview Payment Initiation Service

The following table defines the technical description of the abstract data model as defined in [XS2A-OR] for 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 [XS2A-OR] 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.

  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 [XS2A-OR], 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 [XS2A-OR] 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 for data elements transported in the response body only apply in case of HTTP response code 2xx.** In case of HTTP response code 4xx or 5xx requirements as defined in Section 4.13 apply. In case of the Payment Initiation Response Message, where a payment initiation resource has only been created in case of a 2xx response code, e.g. no resource related information can be returned in case of HTTP response code 4xx or 5xx.

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.	Upd. Req.	Upd. Resp	Auth. Req.	Auth Resp.	Stat. Req.	Stat. Resp
TPP Registration Number						x	m		m		m		m	
TPP Name						x	m		m		m		m	
TPP Roles						x	m		m		m		m	
TPP National Competent Authority						x	m		m		m		m	
Request Identification	X-Request-ID			x			m	m	m	m	m	m	m	m
Resource ID	paymentId				x			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.	Upd. Req.	Upd. Resp	Auth. Req.	Auth Resp.	Stat. Req.	Stat. Resp
Resource ID[3]		x							m		m		m	
Transaction Fees	transactionFees				x			o						
Transaction Fee Indicator	transactionFeeIndicator				x			o						
Access Token (from optional OAuth2)	Authorization			x			c		c		c		c	
Further signature related data	Digest			x			c		c		c		c	
TPP Signing Certificate	TPP-Signature-Certificate			x			c		c		c		c	
TPP Electronic Signature	Signature			x			c		c		c		c	
Transaction Status	transactionStatus				x			m		m		m		m
Funds Availability Flag	fundsAvailable				x									c
PSU Message Information	psuMessage				x			o		o		o		o
TPP Message Information	tppMessages				x			o		o		o		o
PSU Identification	PSU-ID			x			c		c					
PSU Identification Type	PSU-ID-Type			x			c		c					
Corporate Identification	PSU-Corporate-ID			x			c		c		c		c	
Corporate ID Type	PSU-Corporate-ID-Type			x			c		c		c		c	
PSU Password	psuData.password				x				c					

[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.	Upd. Req.	Upd. Resp	Auth. Req.	Auth Resp.	Stat. Req.	Stat. Resp
Available SCA Methods	scaMethods				x			c		c				
Chosen SCA Method	chosenScaMethod				x				c					
PSU Authentication Data	scaAuthenticationData				x						m			
SCA Challenge Data	challengeData				x			c		c				
IP Address PSU	PSU-IP-Address			x			m		o		o		o	
IP Port PSU	PSU-IP-Port			x			o		o		o		o	
PSU User Agent	PSU-User-Agent[4]			x			o		o		o		o	
GEO Information	PSU-Geo-Location			x			o		o		o		o	
Redirect URL ASPSP	_links.scaRedirect				x			c						
ASPSP-SCA-Approach	ASPSP-SCA-Approach			x				c		c				
Further PSU related Information	PSU-Accept			x			o		o		o		o	
	PSU-Accept-Charset			x			o		o		o		o	
	PSU-Accept-Encoding			x			o		o		o		o	
	PSU-Accept-Language			x			o		o		o		o	
	PSU-Http-Method			x			o		o		o		o	
	PSU-Device-ID			x			o		o		o		o	
Redirect Preference	TPP-Redirect-Preferred			x			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.	Upd. Req.	Upd. Resp	Auth. Req.	Auth Resp.	Stat. Req.	Stat. Resp
Redirect URI TPP[5]	TPP-Redirect-URI			x			c							
	TPP-Nok-Redirect_URI			x			o							
Authorisation Preference	TPP-Explicit-Authorisation-Preferred			x			o							
Rejection Preference	TPP-Rejection-NoFunds-Preferred			x			o							
TPP Notification URI	TPP-Notification-URI			x			o							
TPP Notfication Content Preference	TPP-Notification-Content-Preferred			x			o							
TPP Brand Information	TPP-Brand-Logging-Information			x			o							
Payment Product	payment-product	x					m							

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

Remark: The request timestamp of every call is contained in the mandatory HTTP header "Date", cp. Section 14.34 for the formatting information. This timestamp is not contained in the data tables below because it is a mandatory HTTP header field anyhow and because incompatibilities could appear otherwise with future more formalised specification procedures.

Remark: The AIS and PIS service is sharing some sub processes which are once described in Section 7. So, for all Update Data Request/Response Definitions as well as for Authorise Transaction Request/Response Definitions, cp. Section 7.

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

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 Section 4.8. They are not mentioned anymore in the following detailed definitions for matter of better readability, as long as the usage is not mandated.

5.3 Payment Initiation Request

5.3.1 Payment Initiation with JSON encoding of the Payment Instruction

Call

```
POST /v1/payments/{payment-product}
```

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 [XS2A-DP]. 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.

Attribute	Type	Condition	Description
			This is the unique ID of TPP for the payment initiation regarding PSD2 article 47 and EBA RTS article 29.
PSU-ID	String	Conditional	Client ID of the PSU in the ASPSP client interface. Might be mandated in the ASPSP's documentation. It might be contained even if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in a preceding AIS service in the same session. In this case the ASPSP might check whether PSU-ID and token match, according to ASPSP documentation.
PSU-ID-Type	String	Conditional	Type of the PSU-ID; needed in scenarios where PSUs have several PSU-IDs as access possibility. In this case, the mean and use are then defined in the ASPSP's documentation.
PSU-Corporate-ID	String	Conditional	Identification of a Corporate in the Online Channels Might be mandated in the ASPSP's documentation. Only used in a corporate context.
PSU-Corporate-ID-Type	String	Conditional	This is describing the type of the identification needed by the ASPSP to identify the PSU-Corporate-ID content. Mean and use is defined in the ASPSP's documentation. Only used in a corporate context.
Authorization	String	Conditional	Bearer Token. Is contained only, if an OAuth2 based authentication was performed in a pre-step 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.

Attribute	Type	Condition	Description
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.
TPP-Redirect-Preferred	Boolean	Optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach. If it equals "false", the TPP prefers not to be redirected for SCA. The ASPSP will then choose between the Embedded or the Decoupled SCA approach, depending on the choice of the SCA procedure by the TPP/PSU. If the parameter is not used, the ASPSP will choose the SCA approach to be applied depending on the SCA method chosen by the TPP/PSU.
TPP-Redirect-URI	String	Conditional	URI of the TPP, where the transaction flow shall be redirected to after a Redirect. Mandated for the Redirect SCA Approach, specifically when TPP-Redirect-Preferred equals "true". See Section 4.10 for further requirements on this header. It is recommended to always use this header field. **Remark for Future**: This field might be changed to mandatory in the next version of the specification.
TPP-Nok-Redirect-URI	String	Optional	If this URI is contained, the TPP is asking to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the redirect SCA method. This might be ignored by the ASPSP. See Section 4.10 for further requirements on this header.

Attribute	Type	Condition	Description
TPP-Explicit-Authorisation-Preferred	Boolean	Optional	If it equals "true", the TPP prefers to start the authorisation process separately, e.g. because of the usage of a signing basket. This preference might be ignored by the ASPSP, if a signing basket is not supported as functionality. If it equals "false" or if the parameter is not used, there is no preference of the TPP. This especially indicates that the TPP assumes a direct authorisation of the transaction in the next step, without using a signing basket.
TPP-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.
TPP-Notification-URI	String	Optional	URI for the Endpoint of the TPP-API to which the status of the payment initiation should be sent. This header field **may by ignored** by the ASPSP, cp. also the extended service definition in [XS2A-RSNS].
TPP-Notification-Content-Preferred	String	Optional	The string has the form status=X1, …, Xn where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated. The usage of the constants supports the following semantics:

Attribute	Type	Condition	Description
			SCA: A notification on every change of the scaStatus attribute for all related authorisation processes is preferred by the TPP. PROCESS: A notification on all changes of consentStatus or transactionStatus attributes is preferred by the TPP. LAST: Only a notification on the last consentStatus or transactionStatus as available in the XS2A interface is preferred by the TPP. This header field may be ignored, if the ASPSP does not support resource notification services for the related TPP.
TPP-Brand-Logging-Information	String	Optional	This header might be used by TPPs to inform the ASPSP about the brand used by the TPP towards the PSU. This information is meant for logging entries to enhance communication between ASPSP and PSU or ASPSP and TPP. This header might 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, cp. Section 11.1.

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 are defined in Section 11 of this document. Further definitions might be given community or ASPSP specific. In [XS2A-DP], a list of community specific payment product definitions and links regarding community/ASPSP specific payment product definitions are given. 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.
ASPSP-SCA-Approach	String	Conditional	This data element must be contained, if the SCA Approach is already fixed. Possible values are: • EMBEDDED • DECOUPLED • REDIRECT The OAuth SCA approach will be subsumed by REDIRECT.
ASPSP-Notification-Support	Boolean	Conditional	true if the ASPSP supports resource status notification services. false if the ASPSP supports resource status notification in general, but not for the current request. Not used, if resource status notification services are generally not supported by the ASPSP. Shall be supported if the ASPSP supports resource status notification services, see more details in the extended service definition [XS2A-RSNS].

Attribute	Type	Condition	Description
ASPSP-Notification-Content	String	Conditional	The string has the form status=X1, …, Xn where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated. The usage of the constants supports the following semantics: SCA: Notification on every change of the scaStatus attribute for all related authorisation processes is provided by the ASPSP for the related resource. PROCESS: Notification on all changes of consentStatus or transactionStatus attributes is provided by the ASPSP for the related resource. LAST: Notification on the last consentStatus or transactionStatus as available in the XS2A interface is provided by the ASPSP for the related resource. This field must be provided if the ASPSP-Notification-Support =true. The ASPSP might consider the notification content as preferred by the TPP, but can also respond independently of the preferred request.

Response Body

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	Mandatory	The values defined in Section 14.13 might be used.
paymentId	String	Mandatory	resource identification of the generated payment initiation resource.

Attribute	Type	Condition	Description
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. 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

Attribute	Type	Condition	Description
			"startAuthorisationWithAuthenticationMethodSelection" contained in the response body. These methods shall be presented towards the PSU for selection by the TPP.
chosenScaMethod	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 "startAuthorisationWithPsuAuthentication" or "startAuthorisactionWithEncryptedPsuAuthentication" link.
_links	Links	Mandatory	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP. Type of links admitted in this response, (further links might be added for ASPSP defined extensions): "scaRedirect": In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser. "scaOAuth": In case of a SCA OAuth2 Approach, the ASPSP is transmitting the URI where the configuration of the Authorisation Server can be

Attribute	Type	Condition	Description
			retrieved. The configuration follows the OAuth 2.0 Authorisation Server Metadata specification. "confirmation": Might be added by the ASPSP if either the "scaRedirect" or "scaOAuth" hyperlink is returned in the same response message. This hyperlink defines the URL to the resource which needs to be updated with • a confirmation code as retrieved after the plain redirect authentication process with the ASPSP authentication server or • an access token as retrieved by submitting an authorization code after the integrated OAuth based authentication process with the ASPSP authentication server. "startAuthorisation": In case, where an explicit start of the transaction authorisation is needed, but no more data needs to be updated (no authentication method to be selected, no PSU identification nor PSU authentication data to be uploaded). "startAuthorisationWithPsuIdentification": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU identification data. "startAuthorisationWithPsuAuthentication": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU authentication data. "startAuthorisationWithEncryptedPsuAuthentication": Same as startAuthorisactionWithPsuAuthentication, but the

Attribute	Type	Condition	Description
			authentication data need to be encrypted on application level while uploading. "startAuthorisationWithAuthentication MethodSelection": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while selecting the authentication method. This link is contained under exactly the same conditions as the data element "scaMethods" "startAuthorisationWithTransactionAuthorisation": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while authorising the transaction e.g. by uploading an OTP received by SMS. "self": The link to the payment initiation resource created by this request. This link can be used to retrieve the resource data. "status": The link to retrieve the transaction status of the payment initiation. "scaStatus": The link to retrieve the scaStatus of the corresponding authorisation sub-resource. This link is only contained, if an authorisation sub-resource has been already created.
psuMessage	Max500Text	Optional	Text to be displayed to the PSU
tppMessages	Array of TPP Message Information	Optional	Messages to the TPP on operational issues.

Example

Request

```
POST https://api.testbank.com/v1/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

{
   "instructedAmount": {"currency": "EUR", "amount": "123.50"},
   "debtorAccount": {"iban": "DE40100100103307118608"},
   "creditorName": "Merchant123",
   "creditorAccount": {"iban": "DE02100100109307118603"},
   "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/v1/payments/sepa-credit-
transfers/1234-wertiq-983
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "1234-wertiq-983",
  "_links": {
        "scaRedirect": {"href": "https://www.testbank.com/asdfasdfasdf"},
        "self": {"href": "/v1/payments/sepa-credit-transfers/1234-wertiq-
983"},
        "status": {"href": "/v1/payments/sepa-credit-transfers/1234-wertiq-
983/status"},
        "scaStatus": {"href": "/v1/payments/sepa-credit-transfers/1234-
wertiq-983/authorisations/123auth456"}
  }
}
```

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/v1/payments/sepa-credit-
transfers/1234-wertiq-983
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "1234-wertiq-983",
  "_links": {
        "self": {"href": "/v1/payments/sepa-credit-transfers/1234-wertiq-
983"},
        "status": {"href": "/v1/payments/sepa-credit-transfers/1234-wertiq-
983/status"},
        "startAuthorisation": {"href": "/v1/payments/sepa-credit-
transfers/1234-wertiq-983/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/v1/payments/sepa-credit-
transfers/1234-wertiq-983
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "1234-wertiq-983",
  "_links": {
        "scaOAuth": {"href": "https://www.testbank.com/oauth/.well-
known/oauth-authorization-server"},
        "self": {"href": "/v1/payments/sepa-credit-transfers/1234-wertiq-
983"},
        "status": {"href": "/v1/payments/sepa-credit-transfers/1234-wertiq-
983/status"},
        "scaStatus": {"href": "/v1/payments/sepa-credit-transfers/1234-
wertiq-983/authorisations/123auth456"}
  }
}
```

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/v1/payments/sepa-credit-
transfers/1234-wertiq-983
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "1234-wertiq-983",
  "_links": {
     "startAuthorisationWithPsuIdentification": {"href":"/v1/payments/sepa-
credit-transfers/1234-wertiq-983/authorisations"},
     "self": {"href": "/v1/payments/sepa-credit-transfers/1234-wertiq-983"}
    }
}
```

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/v1/payments/sepa-credit-
transfers/1234-wertiq-983
Content-Type:           application/json

{
    "transactionStatus": "RCVD",
    "paymentId": "1234-wertiq-983",
    "_links": {
        "startAuthenticationWithPsuAuthentication": {"href":
"/v1/payments/sepa-credit-transfers/1234-wertiq-983/authorisations"},
         "self": {"href": "/v1/payments/sepa-credit-transfers/1234-wertiq-
983"}
              }
}
```

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

Call

```
POST /v1/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 5.3.3.2 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 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. [XS2A-DP].

Query Parameters

The same query parameter definition as in Section 5.3.1 applies.

Request Header

The same header as in Section 5.3.1, 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 5.3.1.

Example

Request

```
POST https://api.testbank.com/v1/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 5.3.1

5.3.3 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.

5.3.3.1 Bulk Payment Initiation with JSON encoding of the Payment Instruction

Call

```
POST /v1/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. For definitions of basic non euro generic products see [XS2A-DP]..

Attribute	Type	Description
		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 5.3.1 applies.

Request Headers

The same HTTP header definition as in Section 5.3.1 applies.

Request Body

The body definition with the JSON based SEPA bulk payments is contained in Section 11.3, further definitions for non SEPA payments in [XS2A-DP]..

Response

The responses definition is analogous to the initiation of single payments, cp. Section 5.3.1.

5.3.3.2 Bulk Payment Initiation with XML encoding of the Payment Instruction

Call

```
POST /v1/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

Attribute	Type	Description
		• 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 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, [XS2A-DP].

Query Parameters

The same query parameter definition as in Section 5.3.2 applies.

Request Headers

The same HTTP header definition as in Section 5.3.2 applies

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 5.3.2.

5.3.4 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.

5.3.4.1 Standing Orders for Recurring/Periodic Payments in JSON encoding

Call

```
POST /v1/periodic-payments/{payment-product}
```

Path Parameters

The same path parameter to determine the underlying payment type of the recurring payment as in Section 5.3.1 applies.

Query Parameters

The same query parameter definition as in Section 5.3.1 applies.

Request Header

For this initiation the same header as in Section 5.3.1 is used.

Request Body

First, any tag of the underlying payment as defined in Section 11.1 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

Tag	Type	Usage	Description
			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.

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:/v1/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
{
```

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

5.3.4.2 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 /v1/periodic-payments/{product-name}
```

Path Parameters

The same path parameter to determine the underlying payment type of the recurring payment as in Section 5.3.2 applies.

Query Parameters

The same query parameter and HTTP header definition as in Section 5.3.1 applies.

Request Header

The same header definitions as in Section 5.3.1 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; boundary=AaaBbbCcc

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 5.3.2.

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

Tag	Type	Usage	Description
			If not given, it is an infinite standing order.
frequency	Frequency Code	Mandatory	Frequency of the recurring payment resulting from this standing order.
dayOfExecution	DD	Conditional	"31" is ultimo

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 Management Information and XML Payment Information

```
POST https:/v1/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--
```

5.4 Get Transaction Status Request

Call

`GET /v1/{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”

Attribute	Type	Description
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 5.3.1, the status is returned in this JSON based encoding.
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".
psuMessage	Max500Text	Optional	

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

If the Payment Initiation Request is encoded in XML, cp. Section 5.3.2, 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.

Example

Example for JSON based endpoint

Request

```
GET https://api.testbank.com/v1/payments/1234-wertiq-983/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/v1/payments/pain.001-sepa-credit-
transfers/1234-wertiq-983/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>45724572567256897269O6</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>
```

5.5 Get Payment Request

```
GET /v1/{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”
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 Available 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.

5.6 Payment Cancellation Request

Call

```
DELETE /v1/{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.
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.

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 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.
TPP-Redirect-Preferred	Boolean	Optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach. If it equals "false", the TPP prefers not to be redirected for SCA. The ASPSP will then choose between the Embedded or the Decoupled SCA approach, depending on the choice of the SCA procedure by the TPP/PSU. If the parameter is not used, the ASPSP will choose the SCA approach to be applied depending on the SCA method chosen by the TPP/PSU.
TPP-Redirect-URI	String	Conditional	URI of the TPP, where the transaction flow shall be redirected to after a Redirect. Mandated for the Redirect SCA Approach, specifically when TPP-Redirect-Preferred equals "true". See Section 4.10 for further requirements on this header. It is recommended to always use this header field. **Remark for Future**: This field might be changed to mandatory in the next version of the specification.
TPP-Nok-Redirect-URI	String	Optional	If this URI is contained, the TPP is asking to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the redirect SCA method. This might be ignored by the ASPSP. See Section 4.10 for further requirements on this header.

Attribute	Type	Condition	Description
TPP-Explicit-Authorisation-Preferred	Boolean	Optional	If it equals "true", the TPP prefers to start the authorisation process separately, e.g. because of the usage of a signing basket. This preference might be ignored by the ASPSP, if a signing basket is not supported as functionality. If it equals "false" or if the parameter is not used, there is no preference of the TPP. This especially indicates that the TPP assumes a direct authorisation of the transaction in the next step, without using a signing basket.

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.
chosenScaMethod	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 PsuAuthentication" or "startAuthorisactionWith EncryptedPsuAuthentication" link.
_links	Links	Conditional	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP.

Attribute	Type	Condition	Description
			Type of links admitted in this response, (further links might be added for ASPSP defined extensions):
			"startAuthorisation": In case, where just the authorisation process of the cancellation needs to be started, but no additional data needs to be updated for time being (no authentication method to be selected, no PSU identification nor PSU authentication data to be uploaded).
			"startAuthorisationWithPsuIdentification": In case where a PSU Identification needs to be updated when starting the cancellation authorisation: The link to the cancellation-authorisations end-point, where the cancellation sub-resource has to be generated while uploading the PSU identification data.
			"startAuthorisationWithPsuAuthentication": In case of a yet to be created authorisation sub-resource: The link to the cancellation-authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU authentication data.
			"startAuthorisationWithEncryptedPsuAuthentication": Same as startAuthorisactionWithPsu Authentication where the authentication data need to be encrypted on application layer in uploading.
			"startAuthorisationWithAuthentication MethodSelection": The link to the authorisation end-point, where the cancellation-authorisation sub-resource has to be generated while selecting the authentication method. This link is contained under exactly the

Attribute	Type	Condition	Description
			same conditions as the data element "scaMethods"

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

Request

```
DELETE https://api.testbank.com/v1/payments/sepa-credit-
transfers/123456scheduled789

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/v1/payments/sepa-credit-
transfers/123456scheduled789

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": "/v1/payments/sepa-credit-
transfers/123456scheduled789"},
        "status": {"href": "/v1/payments/sepa-credit-
transfers/123456scheduled789/status"},
        "startAuthorisation": {"href": "/v1/payments/sepa-credit-
transfers/123456scheduled789/cancellation-authorisations"}
  }
}
```

5.7 Get Cancellation Authorisation Sub-Resources Request

Call in context of a Payment Cancellation Request

```
GET /v1/{payment-service}/{payment-product}/{paymentId}/cancellation-
authorisations
```

Will deliver an array of resource identifications to all generated cancellation authorisation sub-resources.

Path Parameters

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 initiation resource.

Query Parameters

No specific 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.
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 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
authorisationIds	Array of String	Mandatory	An array of all authorisationIds connected to the cancellation of this payment resource.

Example

Request

```
GET https://api.testbank.com/v1/payments/sepa-credit-transfers/1234-wertiq-983/cancellation-authorisations
Accept:                 application/json
```

```
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7723
Date:                   Sun, 06 Aug 2017 15:04:07 GMT
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7723
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
Content-Type:           application/json

{
  "authorisationIds": ["123auth456"]
}
```

5.8 Multilevel SCA for Payments

The Payment Initiation Requests defined in this section are independent from the need of one or several SCA processes, i.e. independent from the number of authorisations needed for the execution of payments. In contrast, the Initiation Response messages defined above in this section are specific to the processing of one SCA.. In the following the background is explained on diverging requirements on the Payment Initiation Response messages.

For payment initiation with multilevel SCA, this specification requires an explicit start of the authorisation, i.e. links directly associated with SCA processing like "scaRedirect" or "scaOAuth" cannot be contained in the response message of a Payment Initiation Request for a payment, where multiple authorisations are needed. Also if any data is needed for the next action, like selecting an SCA method is not supported in the response, since all starts of the multiple authorisations are fully equal. In these cases, first an authorisation sub-resource has to be generated following the "startAuthorisation" link.

Response Body for Payment Initiation Messages with Multilevel SCA

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	Mandatory	The values defined in Section 14.13 might be used.
paymentId	String	Mandatory	resource identification of the generated payment initiation resource.
transactionFees	Amount	Optional	Can be used by the ASPSP to transport transaction fees relevant for the underlying payments.

Attribute	Type	Condition	Description
transactionFeeIndicator	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.
_links	Links	Mandatory	"startAuthorisation": In case, where an explicit start of the transaction authorisation is needed, but no more data needs to be updated (no authentication method to be selected, no PSU identification nor PSU authentication data to be uploaded). "startAuthorisationWithPsuIdentification": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU identification data. "startAuthorisationWithPsuAuthentication": The link to the authorisation end-point, where an authorisation sub-resource has to be generated while uploading the PSU authentication data. "startAuthorisationWithEncryptedPsuAuthentication": The link to the authorisation end-point, where an authorisation sub-resource has to be generated while uploading the encrypted PSU authentication data. "self": The link to the payment initiation resource created by this request. This link can be used to retrieve the resource data. "status": The link to retrieve the transaction status of the payment initiation.

Attribute	Type	Condition	Description
psuMessage	Max500Text	Optional	Text to be displayed to the PSU
tppMessages	Array of TPP Message Information	Optional	Messages to the TPP on operational issues.

Remark: In difference to the Payment Initiation Flow with one SCA, optimisation processes with implicitly generating authorisation sub-resources are not supported for Multiple SCA to keep the several authorisation processes of different PSUs for the same payment identical, so that the start of the authorisation process is context free. That is, the only steering hyperlinks returned to the TPP after a payment initiations are "start authorisation" hyperlinks with information in addition about mandatory data to be uploaded with the Start Authorisation Request (PSU Identification or PSU Authentication data). It is not possible to upload with the first command the selected authentication method or OTP Response data because this would require to transport the selected authentication methods or challenge data before.

5.9 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 4.5. This is independent of the coding in JSON or XML.

6 Account Information Service

Supported Sub-Services

This specification foresees different types of account information services:

- Transaction reports 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 available 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 available and accessible accounts is as follows:

Definition: The list of **available** accounts of an ASPSP related to a PSU is the list of accounts of a PSU which are open for access through the XS2A interface 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 available 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. In this case, the available 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 Account Information Consent

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

- 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 the 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.

- 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 reports as Account Information type additionally
    - the addressed account identification and
    - the period of the transaction report
    - 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 reports.

  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.

Note: The several Read account data transactions are own transactions following [XS2A-OR], thus a transaction identification will only be used several times in case of pagination while reading transaction lists/account statements.

Consent Models

This specification supports three different consent models, cp. also [XS2A-OR]:

- Detailed Consent

  The Consent Management is handled between TPP and PSU. The TPP is submitting then the detailed consent information – PSU identification, services and account numbers affected – to the ASPSP for authorisation by the PSU. The ASPSP is displaying the consent details to the PSU when performing the SCA.

- Global Consent

  The Consent Management is handled between TPP and PSU. The TPP is submitting then a global consent information, which is only the PSU identification, to the ASPSP for authorisation by the PSU. The ASPSP is displaying only the general access to the PSU's account to the PSU when performing the SCA.

- Bank Offered Consent

  The TPP is asking the ASPSP to deal with the Consent Management. The ASPSP might ask the PSU for a detailed consent modelling or just for a global consent on all AIS services. This is authorised by the PSU with an SCA. The detailed consent information can be retrieved by the TPP in a following step by reading the corresponding consent object.

Account Owner Name Delivery: Potential Impact on Consent Model

The following rules and requirements for the support of this service apply.

- An ASPSP may deliver the account owner name without any extension to the consent model defined above.

or

- An ASPSP may require an explicit consent extension by the PSU to deliver the account owner name.

If an ASPSP offers the Detailed and the Global Consent Model, then the ASPSP is mandated to offer the extension for both models if it is offered for one of these models.

The offer of the consent extension model for the consent for the available accounts is independent from the above requirement, since it also depends on the fact whether the account owner name is delivered in the payment account overview.

The provision of this service by an ASPSP might depend on the fact that the account owner name is also delivered in online channels of the ASPSP.

6.1 Account Information Service Flows

As for the payment initiation, please note that the following flows do not cover all possible variances and are exemplary flows. Especially the flows for

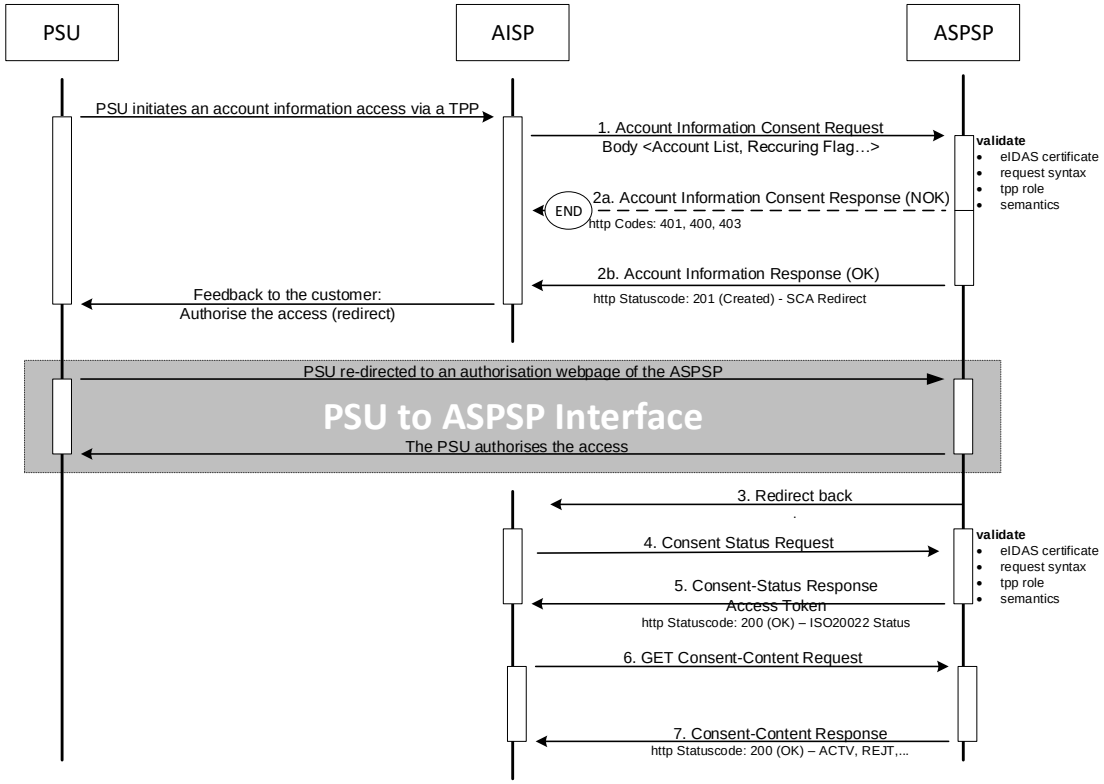
- Redirect and OAuth2 SCA Approach with an explicit start of the Authorisation Process or
- Flows with integrating an explicit confirmation of an authorisation resource

are not shown, since they are following exactly the flow logic as described the Payment Initiation Flows, cp. Section 5.1.

6.1.1 Account Information Consent Flow

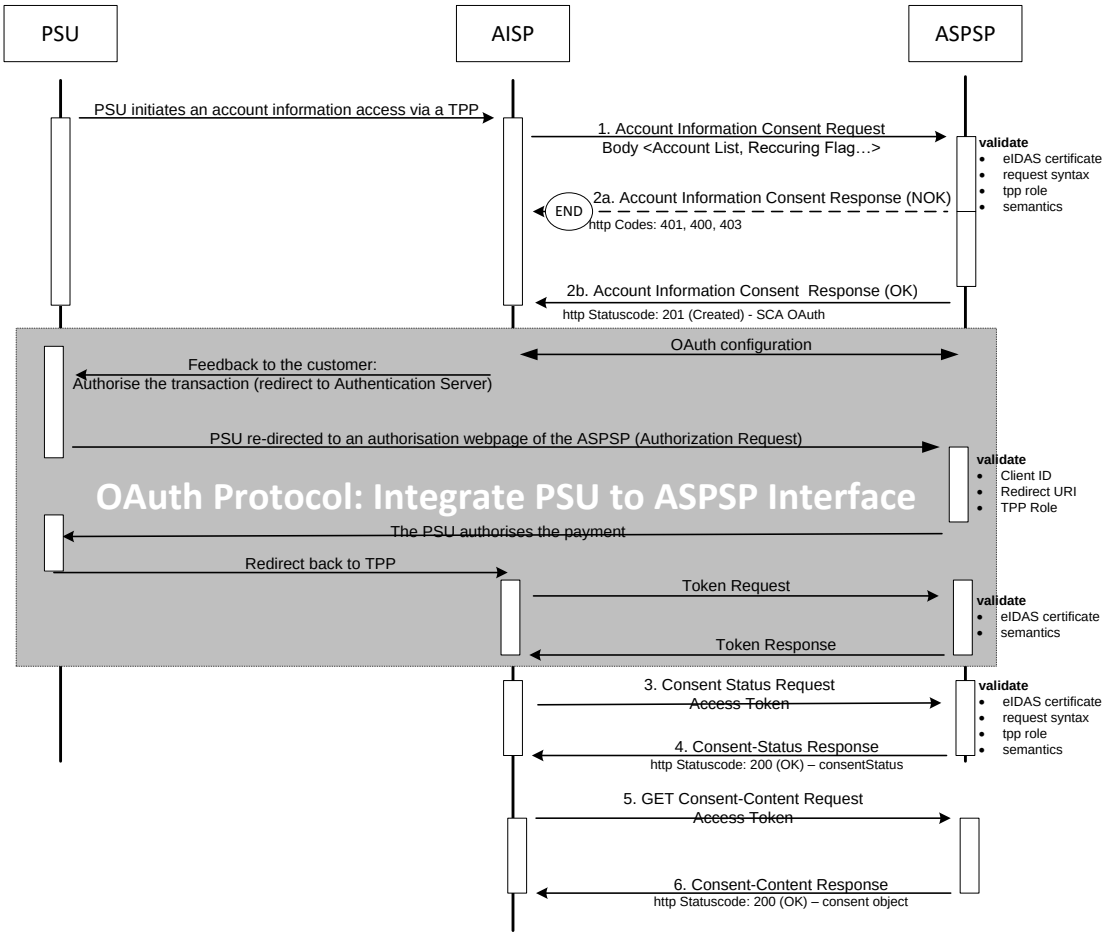
6.1.1.1 Redirect SCA Approach: Implicit Start of the Authorisaction Process

If the ASPSP supports the Redirect SCA Approach, the message flow within the Account Information Consent sub-service is simple. The Account Information Consent Request is followed by a redirection to the ASPSP SCA authorisation site. A status or content request on the created consent resource might be requested by the TPP after the session is re-redirected to the TPP's system.



6.1.1.2 OAuth2 SCA Approach: Implicit Start of the Authorisation Process

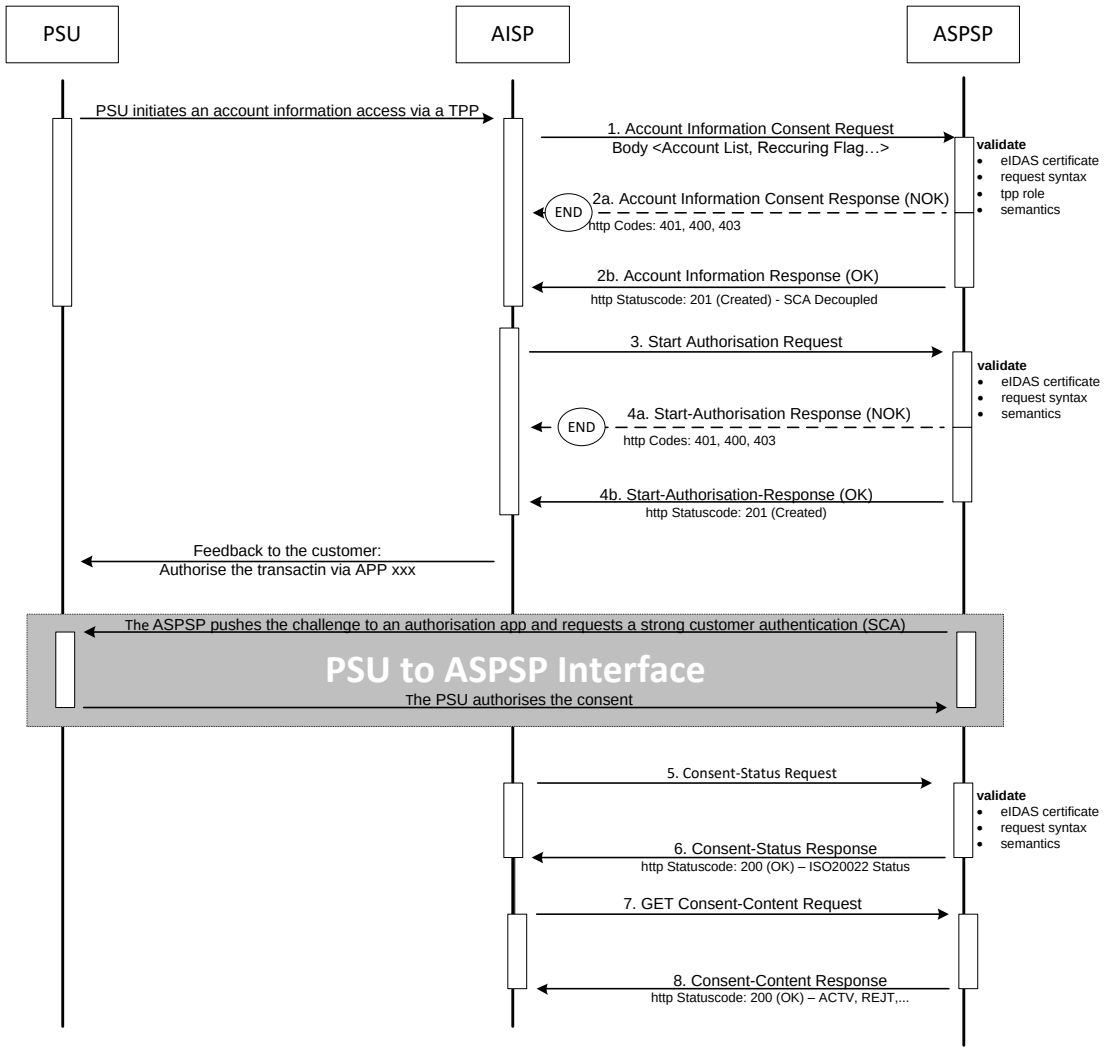
If the ASPSP supports the OAuth2 SCA Approach, the flow is very similar to the Redirect SCA Approach. Instead of redirecting the PSU directly to an authentication server, the OAuth2 protocol is used for the transaction authorisation process. In the following, a flow is shown, where the Authorisation Process in the NextGenPSD2 API has been implicitly started, cp. 5.1.5.



6.1.1.3 Decoupled SCA Approach: Explicit 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 account access consent e.g. via a dedicated mobile app. 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 account access".

After the SCA between ASPSP and PSU, the TPP then needs to ask for the result of the transaction.



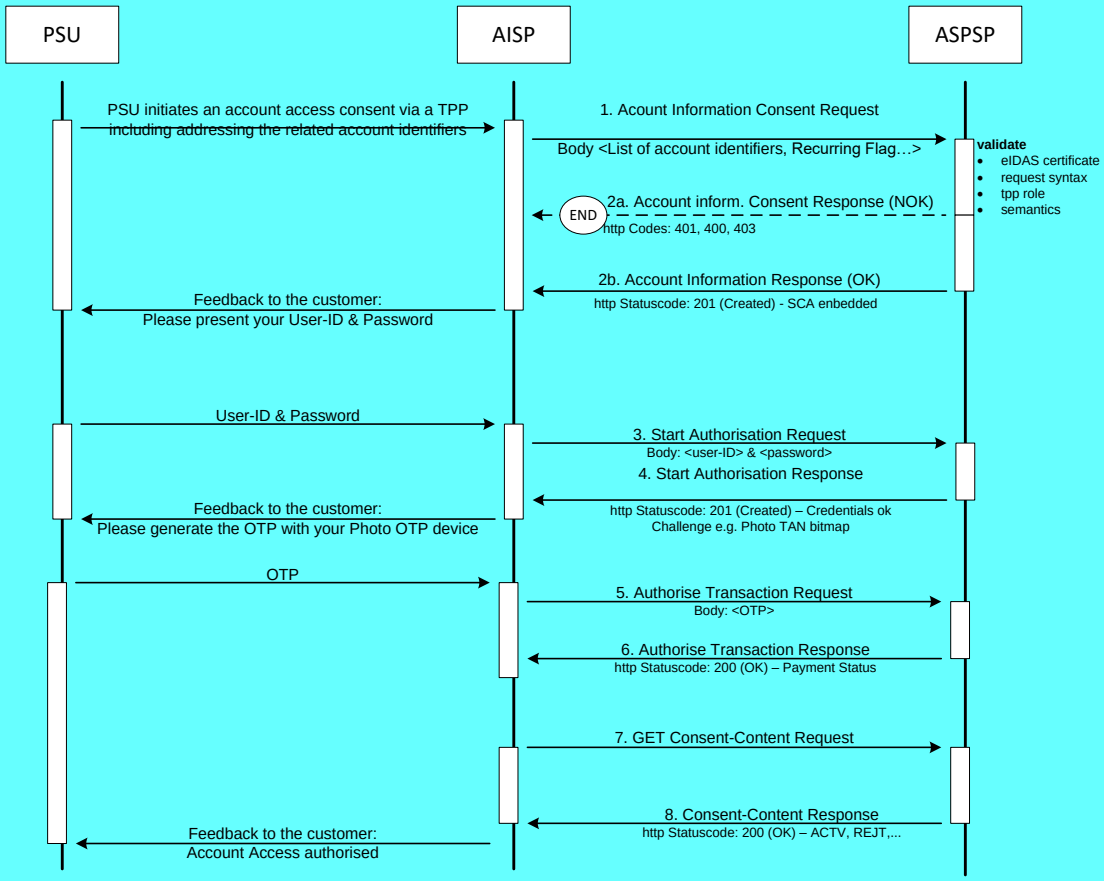
6.1.1.4 Embedded SCA Approach with only one SCA method available

In the following, several exemplary flows are shown, where the ASPSP has chosen to process the SCA methods for the consent approval 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.

Remark: In case where OAuth2 is requested by the ASPSP as a pre-step to replace the PSU- and password by an access token, the sequence of the PSU authentication

with the first authentication factor is omitted. This applies for all examples for the Embedded SCA Approach.

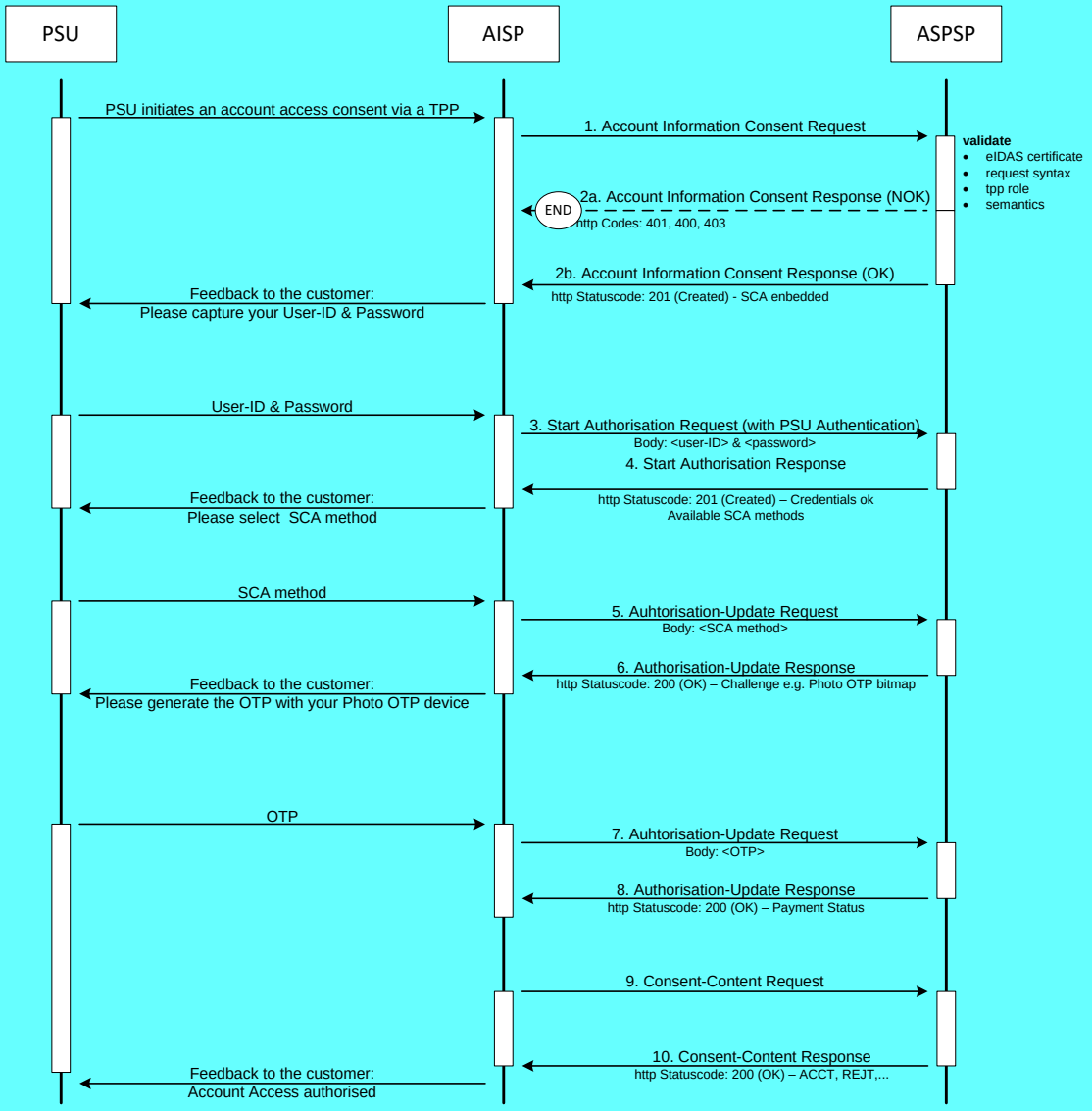
In case where only one SCA method is available, the "Authorise Transaction Request" is added to the flow, where the TPP is transmitting the authentication data of the customer, e.g. an OTP with included dynamic linking to the transaction details.



6.1.1.5 Embedded SCA Approach with Selection of a SCA method

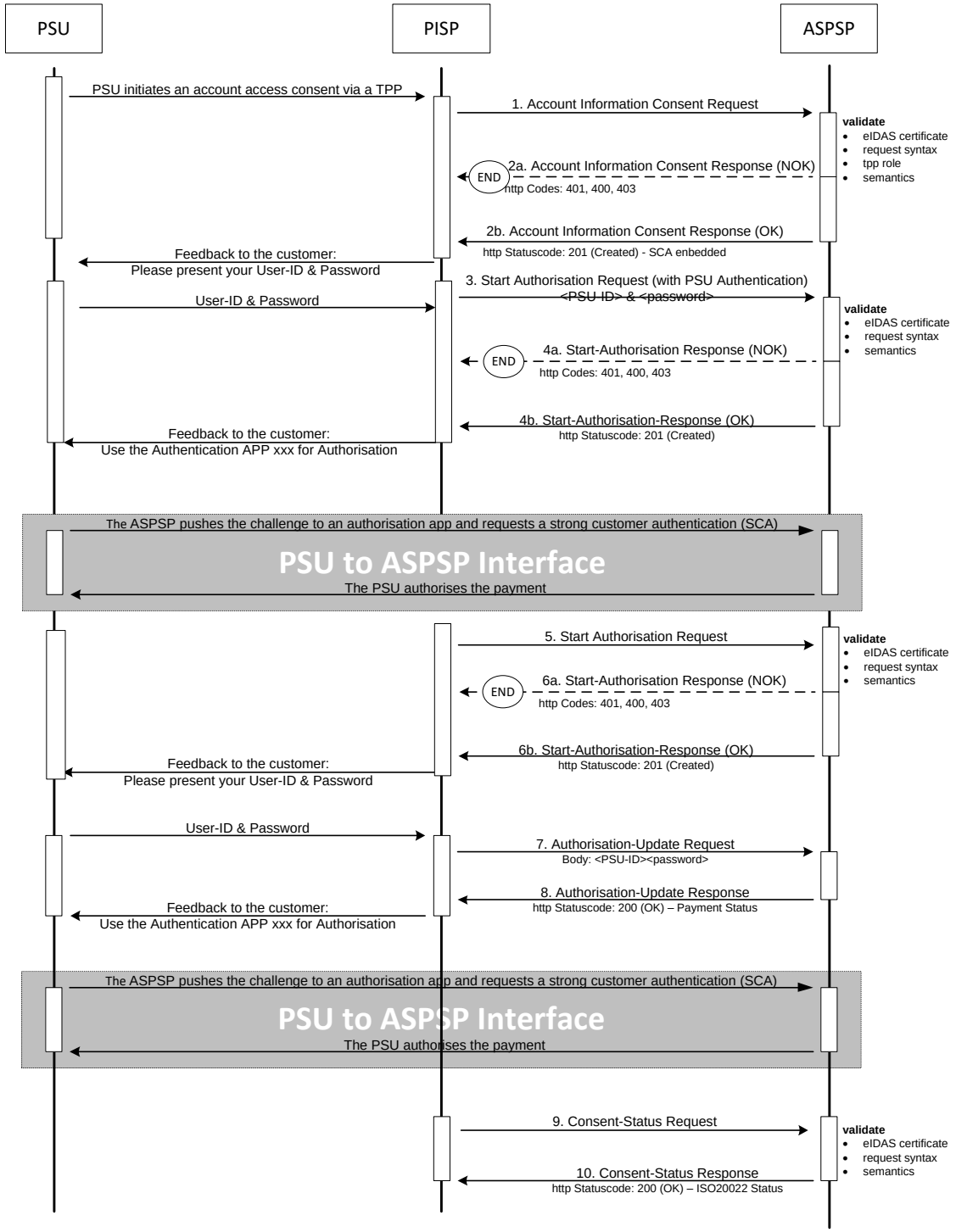
In the following flow, there is a selection of an SCA method added in case of the ASPSP supporting several SCA methods for the corresponding PSU. The ASPSP transmits first the

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.



6.1.1.6 Multilevel SCA Approach: Example Decoupled SCA Approach

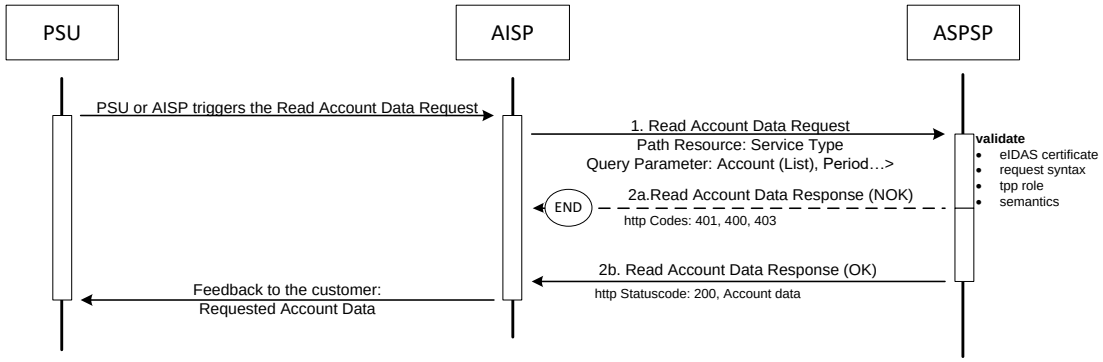
The multilevel SCA Approach flows for the Establish Consent Requests will follow exactly the same pattern as for the Payment Initiation, cp. Section 5.1.12. Whereas the Redirect SCA Approach was used there as an example, the following flow will give an example for the Decoupled SCA Approach:



Note, that in this example the ASPSP asks in Step 6b the TPP to add PSU-ID and password, since it was not uploaded together with the Start Authorisation Process.

6.1.2 Read Account Data Flow

The Read Account Data flow is independent from the corresponding Consent Management flow. It is a simple Request/Response process as follows:



6.2 Data Overview Account Information Service

The following table defines the technical description of the abstract data model as defined [XS2A OR] 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 [XS2A OR] 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.
- The "Usage" column gives an overview on the usage of data elements in the different API Calls. Within [XS2A-OR], 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:
  - Establish Consent Request, which shall be the first API Call for every transaction within XS2A Account Information service.
  - 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 of a second factor is needed.
  - The Read Data Request is the request to retrieve Account Information data, which is addressed to different endpoints with different parameters.
  - The Status Request is used 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 [XS2A-OR] 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. **These requirements for data elements transported in the response body only apply in case of HTTP response code 2xx. In case of HTTP response code 4xx or 5xx requirements as defined in Section 4.13 apply.** In case of the Establish Consent Response Message, where a consent resource has only been created in case of a 2xx response code, e.g. no resource related information can be returned if the HTTP response code equals 4xx or 5xx.

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

Data element	**Attribute encoding**	**Location**					**Usage**									
		Path	Query Param.	Header	Body	Certificate	Establ.. Cons.Req.	Establ. Cons. Resp.	Upd. Data Req.	Upd. Data Resp	Auth. Req.	Auth Resp.	Status Req.	Status Resp.	Read Data Req.	Read Data Resp
Provider Identification		x					m		m		m		m		m	
TPP Registration Number						x	m		m		m		m		m	
TPP Name						x	m		m		m		m		m	
TPP Role						x	m		m		m		m		m	
TPP National Competent Authority						x	m		m		m		m		m	
Request Identification	X-Request-ID			x			m	m	m	m	m	m	m	m	m	m
Resource ID	consentId				x			m								
Resource ID[6]		x							m		m		m			
Resource-ID[7]	Consent-ID			x											m	

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

[7] Please note that the consent identification is addressed by different syntax depending of where it is transported.

Data element	Attribute encoding	Location					Usage									
		Path	Query Param.	Header	Body	Certificate	Establ.. Cons.Req.	Establ. Cons. Resp.	Upd. Data Req.	Upd. Data Resp	Auth. Req.	Auth Resp.	Status Req.	Status Resp.	Read Data Req.	Read Data Resp
Access Token (from optional OAuth2)	Authorization			x			c		c		c		c		c	
TPP Signing Certificate Data	TPP-Signature-Certificate			x			c		c		c		c		c	
TPP Signing Electronic Signature	Signature			x			c		c		c		c		c	
Further signature related data	Digest			x			c		c		c		c		c	
ASPSP-SCA-Approach	ASPSP-SCA-Approach			x				c		c						
Transaction Status	consentStatus				x			m		m		m		m		
SCA Status	scaStatus				x									o		
PSU Message Information	psuMessage				x			o		o		o		o		o
TPP Message Information	tppMessages				x			o		o		o		o		o
PSU Identification	PSU-ID			x			c		c							
PSU Identification Type	PSU-ID-Type			x			c		c							
Corporate Identification	PSU-Corporate-ID			x			c		c		c		c			
Corporate Type	PSU-Corporate-ID-Type						c		c		c		c			
PSU Password	psuData.password				x				c							
Available SCA Methods	scaMethods				x			c		c						
Chosen SCA Method	chosenScaMethod				x				c							

Data element	Attribute encoding	Location					Usage									
		Path	Query Param.	Header	Body	Certificate	Establ.. Cons.Req.	Establ. Cons. Resp.	Upd. Data Req.	Upd. Data Resp	Auth. Req.	Auth Resp.	Status Req.	Status Resp.	Read Data Req.	Read Data Resp
PSU Authentication Data	psuData.authentication				x						m					
SCA Challenge Data	challengeData				x			c		c						
IP Address PSU	PSU-IP-Address			x			m		o		o		o		c	
PSU IP Port	PSU-IP-Port			x			o		o		o		o		o	
Further PSU related Information	PSU-Accept			x			o		o		o		o		o	
	PSU-Accept-Charset			x			o		o		o		o		o	
	PSU-Accept-Encoding			x			o		o		o		o		o	
	PSU-Accept-Language			x			o		o		o		o		o	
	PSU-Http-Method			x			o		o		o		o		o	
	PSU-Device-ID			x			o		o		o		o		o	
PSU User Agent	PSU-User-Agent			x			o		o		o		o		o	
GEO Information	PSU-Geo-Location			x			o		o		o		o		o	
Redirect URL ASPSP	_links.scaRedirect				x			c								
Redirect Preference	TPP-Redirect-Preferred			x			o									
Redirect URL TPP	TPP-Redirect-URI			x			c									
Authorisation Preference	TPP-Explicit-Authorisation-Preferred			x			o									
TPP Notification URI	TPP-Notification-URI			x			o									

Data element	Attribute encoding	Location					Usage									
		Path	Query Param.	Header	Body	Certificate	Establ.. Cons.Req.	Establ. Cons. Resp.	Upd. Data Req.	Upd. Data Resp	Auth. Req.	Auth Resp.	Status Req.	Status Resp.	Read Data Req.	Read Data Resp
TPP Notification Content Preference	TPP-Notification-Content-Preferred			x			o									
TPP Brand Information	TPP-Brand-Logging-Information			x			o									
PSU Account	account				x										c	
PSU Account List	access				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	
Validity Period	validUntil				x		m									
Frequency	frequencyPerDay				x		m									
Recurring Indicator	recurringIndicator				x		m									
Combined service	combinedServiceIndicator				x		m									

Remark: The upper table refers to the "Acccount Information Consent Request" referring dedicated accounts, cp. Section 6.3.1.1.

The XS2A Interface calls which represent the messages defined in [XS2A-OR] for the Payment Consent Request will be defined in the following sections.

Remark: The AIS and PIS services are sharing some sub processes which are once described in Section 7. So, for all Update Data Request/Response Definitions as well as for Authorise Transaction Request/Response Definitions, cp. Section 7.

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 Section 4.8. They are not mentioned anymore in the following detailed definitions for matter of better readability, as long as the usage is not mandated.

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.

Multicurrency Accounts in Submission of Consents

Multicurrency accounts are addressed by just using the external account identifier in the submission of a consent on dedicated accounts, without specifying a currency. Asking for the consent to retrieve account information data of a multicurrency accounts implies getting it for all sub-accounts.

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

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.

Multicurrency Accounts in Reading Transactions

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.

6.3 Establish Account Information Consent

In this section, the Establish Account Information Consent process is defined for the XS2A Interface.

6.3.1 Account Information Consent Request

6.3.1.1 Consent Request on Dedicated Accounts

Call

```
POST /v1/consents
```

Creates an account information consent resource at the ASPSP regarding access to accounts specified in this request.

Side Effects

When this Consent Request is a request where the "recurringIndicator" equals true, and if it exists already a former consent for recurring access on account information for the addressed PSU and potentially addressed corporate identification submitted by this TPP, then the former consent automatically expires as soon as the new consent request is authorised by the PSU.

There are no expiration side effects foreseen for Consent Requests where the "recurringIndicator" equals false.

Query Parameters

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.
PSU-ID	String	Conditional	Client ID of the PSU in the ASPSP client interface. Might be mandated in the ASPSP's documentation. It might be contained even if an OAuth2 based authentication was performed in a pre-step In this case the ASPSP might check whether PSU-ID and token match, according to ASPSP documentation."
PSU-ID-Type	String	Conditional	Type of the PSU-ID, needed in scenarios where PSUs have several PSU-IDs as access possibility.

Attribute	Type	Condition	Description
PSU-Corporate-ID	String	Conditional	Might be mandated in the ASPSP's documentation. Only used in a corporate context.
PSU-Corporate-ID-Type	String	Conditional	Might be mandated in the ASPSPs documentation. Only used in a corporate context.
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.
Authorization	String	Conditional	If OAuth2 has been chosen as pre-step to authenticate the PSU.
TPP-Redirect-Preferred	Boolean	Optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach. If it equals "false", the TPP prefers not to be redirected for SCA. The ASPSP will then choose between the Embedded or the Decoupled SCA approach, depending on the choice of the SCA procedure by the TPP/PSU. If the parameter is not used, the ASPSP will choose the SCA approach to be applied depending on the SCA method chosen by the TPP/PSU.

Attribute	Type	Condition	Description
TPP-Redirect-URI	String	Conditional	URI of the TPP, where the transaction flow shall be redirected to after a Redirect. Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true". See Section 4.10 for further requirements on this header. It is recommended to always use this header field. **Remark for Future**: This field might be changed to mandatory in the next version of the specification.
TPP-Nok-Redirect-URI	String	Optional	
TPP-Explicit-Authorisation-Preferred	Boolean	Optional	If it equals "true", the TPP prefers to start the authorisation process separately, e.g. because of the usage of a signing basket. This preference might be ignored by the ASPSP, if a signing basket is not supported as functionality. If it equals "false" or if the parameter is not used, there is no preference of the TPP. This especially indicates that the TPP assumes a direct authorisation of the transaction in the next step, without using a signing basket.
TPP-Notification-URI	String	Optional	URI for the Endpoint of the TPP-API to which the status of the consent should be sent. This header field **may by ignored** by the ASPSP, cp. also the extended service definition in [XS2A-RSNS].

Attribute	Type	Condition	Description
TPP-Notification-Content-Preferred	String	Optional	The string has the form status=X1, …, Xn where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated. The usage of the constants supports the following semantics: SCA: A notification on every change of the scaStatus attribute for all related authorisation processes is preferred by the TPP. PROCESS: A notification on all changes of consentStatus or transactionStatus attributes is preferred by the TPP. LAST: Only a notification on the last consentStatus or transactionStatus as available in the XS2A interface is preferred by the TPP. This header field may be ignored, if the ASPSP does not support resource notification services for the related TPP.
TPP-Brand-Logging-Information	String	Optional	This header might be used by TPPs to inform the ASPSP about the brand used by the TPP towards the PSU. This information is meant for logging entries to enhance communication between ASPSP and PSU or ASPSP and TPP. The ASPSP might ignore this field.

Request Body

Attribute	Type	Condition	Description
access	Account Access	Mandatory	Requested access services.
recurringIndicator	Boolean	Mandatory	true, if the consent is for recurring access to the account data false, if the consent is for one access to the account data
validUntil	ISODate	Mandatory	This parameter is defining a valid until date (including the mentioned date) for the requested consent. The content is the local ASPSP date in ISODate Format, e.g. 2017-10-30. Future dates might get adjusted by ASPSP. If a maximal available date is requested, a date in far future is to be used: "9999-12-31". In both cases, the consent object to be retrieved by the GET Consent Request will contain the adjusted date.
frequencyPerDay	Integer	Mandatory	This field indicates the requested maximum frequency for an access without PSU involvement per day. For a one-off access, this attribute is set to "1". The frequency needs to be greater equal to one. If not otherwise agreed bilaterally between TPP and ASPSP, the frequency is less equal to 4.

Attribute	Type	Condition	Description
			Remark for Future: Additional conditions might be added later to deal with the situation where the PSU is consenting towards the TPP for account access only where the PSU is actively asking.
combinedService Indicator	Boolean	Mandatory	If true indicates that a payment initiation service will be addressed in the same "session", cp. Section 9.

Note: All permitted major "access" attributes ("accounts", "balances" and "transactions") used in this message shall carry a non-empty array of account references, indicating the accounts where the type of access is requested. It can contain references regarding current account and/or card accounts. Please note that a "transactions" or "balances" access right also gives access to the generic /accounts endpoints, i.e. is implicitly supporting also the "accounts" access.

Note: The “access” attribute “additionalInformation” contains further sub-attributes. The additionalInformation attribute may only be used together with one of the major “access” attributes, see above. There is no requirement whether the related sub-attributes of “additionalInformation” carry also non-empty attributes as well where applicable. In case of an empty array in such a sub-attribute, the semantic is that the TPP is asking for the additionalInformation for all accounts which are addressed in at least one of the major “access” attributes.

Note: Even if the ASPSP is not requiring an explicit consent for an additionalInformation, e.g. the account owner name, the ASPSP should ignore a related consent request extension of the TPP, i.e. not reject the related consent request. This also applies in case the requested access is not offered (e.g. account owner name).

This specification mandates the ASPSP to support all POST consent requests with dedicated accounts, i.e. POST requests with the above mentioned sub-attributes, where at least one sub-attribute is contained, and where all contained sub-attributes carry a non-empty array of account references. This results in a consent on dedicated accounts. For this Consent Request on Dedicated Accounts, no assumptions are made for the SCA Approach by this specification.

Optionally, the ASPSP can support also Consent Requests, where the above mentioned sub-attributes "accounts", "balances" and "transactions" only carry an empty array or where the

sub-attributes "availableAccounts", “availableAccountsWithBalance”or "allPsd2" are used – all of them with the value "allAccounts" or “allAccountsWithOwnerName”, cp. 6.3.1.2,

Response Code

HTTP Response Code equals 201.

Response Header

Attribute	Type	Condition	Description
Location	String	Mandatory	Location of the created resource.
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
ASPSP-SCA-Approach	String	Conditional	Possible values are: • EMBEDDED • DECOUPLED • REDIRECT OAuth will be subsumed by the constant value REDIRECT
ASPSP-Notification-Support	Boolean	Conditional	true if the ASPSP supports resource status notification services. false if the ASPSP supports resource status notification in general, but not for the current request. Not used, if resource status notification services are generally not supported by the ASPSP. Shall be supported if the ASPSP supports resource status notification services, see more details in the extended service definition [XS2A-RSNS].
ASPSP-Notification-Content	String	Conditional	The string has the form status=X1, …, Xn

Attribute	Type	Condition	Description
			where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated. The usage of the constants supports the following semantics: SCA: Notification on every change of the scaStatus attribute for all related authorisation processes is provided by the ASPSP for the related resource. PROCESS: Notification on all changes of consentStatus or transactionStatus attributes is provided by the ASPSP for the related resource. LAST: Notification on the last consentStatus or transactionStatus as available in the XS2A interface is provided by the ASPSP for the related resource. This field must be provided if the ASPSP-Notification-Support =true. The ASPSP might consider the notification content as preferred by the TPP, but can also respond independently of the preferred request.

Response Body

Attribute	Type	Condition	Description
consentStatus	Consent Status	Mandatory	authentication status of the consent
consentId	String	Mandatory	Identification of the consent resource as it is used in the API structure

Attribute	Type	Condition	Description
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 "selectAuthenticationMethods" contained in the response body. These methods shall be presented towards the PSU for selection by the TPP.
chosenScaMethod	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 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 startAuthorisationWithPsuAuthentication or startAuthorisationWithEncryptedPsuAuthentication link.
_links	Links	Mandatory	A list of hyperlinks to be recognised by the TPP. Type of links admitted in this response (which might be extended by single ASPSPs as indicated in its XS2A documentation): "scaRedirect": In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser. "scaOAuth": In case of an OAuth2 based Redirect Approach, the ASPSP is transmitting the link where the configuration of the OAuth2

Attribute	Type	Condition	Description
			Server is defined. The configuration follows the OAuth 2.0 Authorisation Server Metadata specification. "confirmation": Might be added by the ASPSP if either the "scaRedirect" or "scaOAuth" hyperlink is returned in the same response message. This hyperlink defines the URL to the resource which needs to be updated with • a confirmation code as retrieved after the plain redirect authentication process with the ASPSP authentication server or • an access token as retrieved by submitting an authorization code after the integrated OAuth based authentication process with the ASPSP authentication server. "startAuthorisation": In case, where an explicit start of the transaction authorisation is needed, but no more data needs to be updated (no authentication method to be selected, no PSU identification nor PSU authentication data to be uploaded). "startAuthorisationWithPsuIdentification": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU identification data. "startAuthorisationWithPsuAuthentication": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU authentication data. "startAuthorisationWithEncryptedPsu Authentication":

Attribute	Type	Condition	Description
			The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the encrypted PSU authentication data. "startAuthorisationWithAuthentication MethodSelection": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while selecting the authentication method. This link is contained under exactly the same conditions as the data element "scaMethods" "startAuthorisationWithTransactionAuthorisation": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while authorising the transaction e.g. by uploading an OTP received by SMS. "self": The link to the Establish Account Information Consent resource created by this request. This link can be used to retrieve the resource data. "status": The link to retrieve the transaction status of the consent request. "scaStatus": The link to retrieve the scaStatus of the corresponding authorisation sub-resource. This link is only contained, if an authorisation sub-resource has been already created.
psuMessage	Max500Text	Optional	Text to be displayed to the PSU, e.g. in a Decoupled SCA Approach

Example

Request

```
POST https://api.testbank.com/v1/consents
Content-Type:          application/json
```

```
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7756
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 15:05:37 GMT

{
  "access": {
      "balances": [
          { "iban": "DE40100100103307118608" },
          { "iban": "DE02100100109307118603",
            "currency": "USD"
          },
          { "iban": "DE67100100101306118605" }
        ],
      "transactions": [
          { "iban": "DE40100100103307118608" },
          { "maskedPan": "123456xxxxxx1234" }
        ]
    },
  "recurringIndicator": true,
  "validUntil": "2017-11-01",
  "frequencyPerDay": 4

}
```

Response in case of a redirect

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     REDIRECT
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Location:               "v1/consents/1234-wertiq-983"
Content-Type:           application/json

{
  "consentStatus": "received",
  "consentId": "1234-wertiq-983",
  "_links": {
    "scaRedirect": {"href": "https://www.testbank.com/authentication/1234-
wertiq-983"},
    "status": {"href": "/v1/consents/1234-wertiq-983/status"},
    "scaStatus": {"href": "v1/consents/1234-wertiq-983/
authorisations/123auth567"}
  }
}
```

Response in case of a redirect with a dedicated start of the authorisation process

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     REDIRECT
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Location:               "v1/consents/1234-wertiq-983"
Content-Type:           application/json

{
  "consentStatus": "received",
  "consentId": "1234-wertiq-983",
  "_links": {
    "startAuthorisation": {"href": "v1/consents/1234-wertiq-
983/authorisations"}
  }
}
```

Response *in case of the OAuth2 approach with an implicit generated authorisation resource*

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     REDIRECT
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Location:               "v1/consents/1234-wertiq-983"
Content-Type:           application/json

{
  "consentStatus": "received",
  "consentId": "1234-wertiq-983",
  "_links": {
    "self": {"href": "/v1/consents/1234-wertiq-983"},
    "scaStatus": {"href": "v1/consents/1234-wertiq-
983/authorisations/123auth567"},
    "scaOAuth": {"href": "https://www.testbank.com/oauth/.well-known/oauth-
authorization-server"}
  }
}
```

Response in case of the decoupled approach

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     DECOUPLED
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Location:               "v1/consents/1234-wertiq-983"
Content-Type:           application/json

{
  "consentStatus": "received",
  "consentId": "1234-wertiq-983",
  "_links": {
    "startAuthorisationWithPsuIdentification": {"href": "/v1/consents/1234-
wertiq-983/authorisations"}
  }
}
```

Response in case of the embedded approach

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     EMBEDDED
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Location:               "v1/consents/1234-wertiq-983"
Content-Type:           application/json

{
  "consentStatus": "received",
  "consentId": "1234-wertiq-983",
  "_links": {
    "startAuthorisationWithPsuAuthentication": {"href": "/v1/consents/1234-
wertiq-983/authorisations"}
  }
}
```

Example for Consent Request with dedicated request for account owner name

Request

```
POST https://api.testbank.com/v1/consents
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7756
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 15:05:37 GMT

{
  "access": {
      "balances": [
          { "iban": "DE40100100103307118608" },
          { "iban": "DE02100100109307118603",
            "currency": "USD"
          },
          { "iban": "DE67100100101306118605" }
        ],
      "transactions": [
          { "iban": "DE40100100103307118608" },
          { "maskedPan": "123456xxxxxx1234" }
        ],
       "additionalInformation" :
          {"ownerName": [{ "iban": "DE40100100103307118608" }]
          }
    },
  "recurringIndicator": false,
  "validUntil": "2017-11-01",
  "frequencyPerDay": 1,
  "combinedServiceIndicator": false
}
```

6.3.1.2 Consent Request on Account List or without Indication of Accounts

Consent Request on Account List of Available Accounts

This function is supported by the same call as the Consent Request on Dedicated Accounts. The only difference is that the call only contains the "availableAccounts" or “availableAccountsWithBalance” sub attribute within the "access" attribute with value "allAccounts".

In this case the call creates an account information consent resource at the ASPSP to return a list of all **available** accounts, resp. all available accounts with its balances. For the first of these specific Consent Requests, no assumptions are made for the SCA Approach by this specification, since there are no balances or transaction information contained and this is then not unambigously required by [EBA-RTS]. It is up to the ASPSP to implement the appropriate requirements on customer authentication.

Consent Request without Indication of Accounts – Bank Offered Consent

This function is supported by the same call as the Consent Request on Dedicated Accounts. The only difference is that the call contains the "accounts", "balances" and/or "transactions" sub attribute within the "access" attribute all with an empty array.

The ASPSP will then agree bilaterally directly with the PSU on which accounts the requested access consent should be supported. The result can be retrieved by the TPP by using the GET Consent Request method, cp. 6.3.3. For this function the Embedded SCA Approach is not supported.

Consent Request for Access to all Accounts for all PSD2 defined AIS – Global Consent

This function is supported by the same call as the Consent Request on Dedicated Accounts. The only difference is that the call contains the "allPsd2" sub attribute within the "access" attribute with the value "allAccounts".

If this function is supported, it will imply a consent on all available accounts of the PSU on all PSD2 related account information services. For this specific Consent Request, no assumptions are made for the SCA Approach by this specification.

Example Consent on Account List of Available Accounts

Request

```
POST https://api.testbank.com/v1/consents
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7756
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 15:05:37 GMT

{"access":
      {"availableAccounts": "allAccounts"},
 "recurringIndicator": false,
 "validUntil": "2017-08-06",
 "frequencyPerDay": 1
}
```

Example Consent without dedicated Account

Request

```
POST https://api.testbank.com/v1/consents
Content-Type            application/json
X-Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7756
```

```
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 15:05:37 GMT

{"access":
      {"balances": [],
       "transactions": []},
 "recurringIndicator": true,
 "validUntil": "2017-11-01",
 "frequencyPerDay": 4
}
```

6.3.2 Get Consent Status Request

Call

```
GET /v1/consents/{consentId}/status
```

Can check the status of an account information consent resource.

Path Parameters

Attribute	Type	Description
consentId	String	The consent identification assigned to the created resource.

Query Parameters

No specific 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.
Authorization	String	Conditional	Is contained only, if an OAuth2 based SCA was performed in the corresponding consent transaction or if OAuth2 has been used in a pre-step.

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
consentStatus	Consent Status	Mandatory	This is the overall lifecycle status of the consent.
psuMessage	Max500Text	Optional	

Example

Request

```
GET https://api.testbank.com/v1/consents/qwer3456tzui7890/status
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 15:05:46 GMT
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json
```

```
{
  "consentStatus": "valid"
}
```

6.3.3 Get Consent Request

Call

```
GET /v1/consents/{consentId}
```

Returns the content of an account information consent object. This is returning the data for the TPP especially in cases, where the consent was directly managed between ASPSP and PSU e.g. in a re-direct SCA Approach.

Path Parameters

Attribute	Type	Description
consentId	String	ID of the corresponding consent object as returned by an Account Information Consent Request

Query Parameters

No specific query parameter.

Request Header

The same as defined in Section 6.3.2.

Request Body

No request body.

Response Code

HTTP Response Code equals 200.

Response Header

The same as defined in Section 6.3.2.

Response Body

Attribute	Type	Condition	Description
access	Account Access	Mandatory	
recurringIndicator	Boolean	Mandatory	
validUntil	ISODate	Mandatory	
frequencyPerDay	Integer	Mandatory	
lastActionDate	ISODate	Mandatory	This date is containing the date of the last action on the consent object either through the XS2A interface or the PSU/ASPSP interface having an impact on the status.
consentStatus	Consent Status	Mandatory	The status of the consent resource.
_links	Links	Optional	Type of links recommended for this response is "account" and/or "cardAccount", depending on the nature of the consent.

Example

Request

```
GET https://api.testbank.com/v1/consents/qwer3456tzui7890
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json

{
  "access":
```

```
      {"balances":
          [{"iban": "DE23100100101234567 89"}],
      "transactions":
          [{"iban": "DE23100100101234567 89"},
           {"pan": "123456xxxxxx3457"}]
      },
 "recurringIndicator": true,
 "validUntil": "2017-11-01",
 "frequencyPerDay": 4,
 "consentStatus": "valid",
 "_links": {"account": {"href": "/v1/accounts"}}
}
```

Remark: This specification supports no detailed links to AIS service endpoints corresponding to this account. This is due to the fact, that the /accounts endpoint will deliver all detailed information, including the hyperlinks e.g. to the balances or transactions of certain accounts. Still due to the guiding principles, the ASPSP may deliver more links in addition, which then will be documented in the ASPSPs XS2A API documentation.

6.3.4 Multilevel SCA for Establish Consent

The Establish Account Information Consent Request messages defined in this section are independent from the need of one or several SCA processes, i.e. independent from the number of authorisations needed for establishing the consent. In contrast, the Establish Account Information Consent Response messages defined above in this section are specific to the processing of one SCA. In the following the background is explained on diverging requirements on the Establish Account Information Consent Response messages.

For establish account information consent with multilevel SCA, this specification requires an explicit start of the authorisation, i.e. links directly associated with SCA processing like "scaRedirect" or "scaOAuth" cannot be contained in the response message of a Establish Account Information Consent Request for a consent, where multiple authorisations are needed. Also if any data is needed for the next action, like selecting an SCA method is not supported in the response, since all starts of the multiple authorisations are fully equal. In these cases, first an authorisation sub-resource has to be generated following the "startAuthorisation" link.

Response Body for Establish Account Information Messages with Multilevel SCA

Attribute	Type	Condition	Description
consentStatus	Consent Status	Mandatory	The values defined in Section 14.14 might be used.

Attribute	Type	Condition	Description
consentId	String	Mandatory	resource identification of the generated payment initiation resource.
_links	Links	Mandatory	"startAuthorisation": In case, where an explicit start of the transaction authorisation is needed, but no more data needs to be updated (no authentication method to be selected, no PSU identification nor PSU authentication data to be uploaded). "startAuthorisationWithPsuIdentification": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU identification data. "startAuthorisationWithPsuAuthentication": The link to the authorisation end-point, where an authorisation sub-resource has to be generated while uploading the PSU authentication data. "startAuthorisationWithEncryptedPsuAuthentication": The link to the authorisation end-point, where an authorisation sub-resource has to be generated while uploading the encrypted PSU authentication data. "self": The link to the consent resource created by this request. This link can be used to retrieve the resource data. "status": The link to retrieve the status of the consent.
psuMessage	Max500Text	Optional	Text to be displayed to the PSU
tppMessages	Array of TPP Message Information	Optional	Messages to the TPP on operational issues.

Remark: In difference to the Establish Account Information Consent Flow with one SCA, optimisation processes with implicitly generating authorisation sub-resources are not

supported for Multiple SCA to keep the several authorisation processes of different PSUs for the same consent identical, so that the start of the authorisation process is context free. That is, the only steering hyperlinks returned to the TPP after starting establishing a consent are "start authorisation" hyperlinks with information in addition about mandatory data to be uploaded with the Start Authorisation Request (PSU Identification or PSU Authentication data). It is not possible to upload with the first command the selected authentication method or OTP Response data because this would require to transport the selected authentication methods or challenge data before.

6.4 Delete an Account Information Consent Object

The TPP can delete an account information consent object if needed with the following call:

Call

```
DELETE /v1/consents/{consentId}
```

Deletes a given consent.

Path Parameters

Attribute	Type	Description
consentId	String	Contains the resource-ID of the consent to be deleted.

Query Parameters

No specific 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.
Authorization	String	Conditional	Is contained only, if an OAuth2 based SCA was performed in the corresponding consent transaction or if OAuth2 has been used in a pre-step.

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

No Response Body

Example

Request

```
DELETE https://api.testbank.com/v1/consents/qwer3456tzui7890
X-Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date                    Sun, 13 Aug 2017 17:05:37 GMT
```

Response

```
HTTP/1.x 204 No Content
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
```

6.5 Read Account Data Requests

6.5.1 Read Account List

Call

```
GET /v1/accounts {query-parameters}
```

Reads a list of bank 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 available accounts ("availableAccounts" access rights respectively "availableAccountsWithBalance", cp. Section 6.3.1.2), 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 "accounts", "balances", "transactions" or "allPsd2" are not sufficient, then more details can be retrieved by addressing the /accounts/{account-id} endpoint, cp. Section 6.5.2.

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	String	Mandatory	Shall be contained since "Establish Consent Transaction" was 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 if

Attribute	Type	Condition	Description
			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	

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": "DE23100100101234567 89",
       "currency": "EUR",
       "product": "Girokonto",
```

```
       "cashAccountType": "CACC",
       "name": "Main Account",
       "_links": {
            "balances": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/balances"},
            "transactions": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/transactions"}}
      },
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e81g",
       "iban": "DE2310010010123456788",
       "currency": "USD",
       "product": "Fremdwährungskonto",
       "cashAccountType": "CACC",
       "name": "US Dollar Account",
       "_links": {
            "balances": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e81g/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": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/balances"},
            "transactions": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/transactions"}}
      },
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e81g",
       "iban": "DE2310010010123456788",
       "currency": "USD",
       "product": "Fremdwährungskonto",
       "cashAccountType": "CACC",
       "name": "US Dollar Account",
```

```
       "_links": {
            "balances": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e81g/balances"},
            "transactions": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e81g/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": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e333/balances"},
            "transactions": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e333/transactions"}}
      },
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e80f",
       "iban": "DE2310010010123456788",
       "currency": "EUR",
       "product": "Girokonto",
       "cashAccountType": "CACC",
       "name": "Main Account",
       "_links": {
            "balances": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/balances"},
            "transactions": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/transactions"}}
      },
      {"resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e81g",
       "iban": "DE2310010010123456788",
       "currency": "USD",
       "product": "Fremdwährungskonto",
       "cashAccountType": "CACC",
       "name": "US Dollar Account",
```

```
       "_links": {
            "balances": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e81g/balances"},
            "transactions": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e81g/transactions"} }
       }
]}
```

6.5.2 Read Account Details

Call

```
GET /v1/accounts/{account-id} {query-parameters}
```

Reads details about an 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.

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	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
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 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	String	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": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/balances"},
            "transactions": {"href": "/v1/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": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/balances"},
            "transactions": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/transactions"}}
      }
}
```

6.5.3 Read Balance

Call

```
GET /v1/accounts/{account-id}/balances
```

Reads account data from a given 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 Account List" call, cp. Section 6.5.1.

The account-id is constant at least throughout the lifecycle of a given consent.

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

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	optional	Identifier of the addressed account. Remark for Future: It is recommended to use this data element. The condition might change to "mandatory" in a next version of the specification.
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
      }
```

```
    ]
}
```

6.5.4 Read Transaction List

Call

```
GET /v1/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.

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 6.5.1.

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 the PATH might be already given in detail 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

Attribute	Type	Condition	Description
			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 in this API nor other channels of the ASPSP.
dateTo	ISODate	Optional	End date (inclusive the data dateTo) of the transaction list, default is "now" 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.
entryReferenceFrom	String	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 identification entryReferenceFrom alternatively to the above defined period. This is a 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.
bookingStatus	String	Mandatory	Permitted codes are "booked", "pending", "both" and “information”. "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. If supported, "both" means to request transaction reports of transaction of

Attribute	Type	Condition	Description
			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.
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.
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.

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

Attribute	Type	Condition	Description
			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.
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**: Content types might be extended in the next version of the specification. This shall enable the TPP to address different camt.05x versions or different MT94x versions in a corporate context. The TPP then could e.g. say: "I prefer MT942, but take MT940 if MT942 is not available."

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
```

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	optional	Identifier of the addressed account. Remark for Future: It is recommended to use this data element. The condition might change to "mandatory" in a next version of the specification.
transactions	Account Report	Optional	JSON based account report. This account report contains transactions resulting from the query parameters.

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 report might be downloaded from in case where transaction reports 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/v1/accounts/qwer3456tzui7890/transactions?dateFrom
=2017-07-01&dateTo=2017-07-30&bookingStatus=both
Accept: application/json, text/plain;q=0.9, application/xml;q=0.8
```

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",
       "creditorName": "John Miles",
       "creditorAccount": {"iban": "DE67100100101306118605"},
       "transactionAmount": {"currency": "EUR", "amount": "256.67"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": "Example 1"
```

```
      },{
       "transactionId": "1234568",
       "debtorName": "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",
       "creditorName": "Claude Renault",
       "creditorAccount": {"iban": "FR7612345987650123456789014"},
       "transactionAmount": {"currency": "EUR", "amount": "-100.03"},
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": "Example 3"
      }],
   "_links":
      {"account": {"href":"/v1/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/v1/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",
       "creditorName": "John Miles",
       "creditorAccount": {"iban": "DE67100100101306118605"},
       "transactionAmount": {"currency": "EUR", "amount": "-256.67"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": "Example 1"
      },{
       "transactionId": "1234568",
       "debtorName": "Paul Simpson",
       "debtorAccount": {"iban": "NL76RABO0359400371"},
       "transactionAmount": {"currency": "EUR", "amount": "343.01"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": "Example 2"
      },{
       "transactionId": "1234569",
       "debtorName": "Pepe Martin",
       "debtorAccount": {"iban": "SE9412309876543211234567"},
       "transactionAmount": {"currency": "USD", "amount": "100"},
       "bookingDate": "2017-10-25",
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": "Example 3"
      }],
   "pending":
     [{
       "transactionId": "1234570",
       "creditorName": "Claude Renault",
       "creditorAccount": {"iban": "FR7612345987650123456789014"},
       "transactionAmount": {"currency": "EUR", "amount": "-100.03"},
       "valueDate": "2017-10-26",
       "remittanceInformationUnstructured": "Example 4"
      }],
   "_links":
     {"account": {"href": "/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f"}}
   }
}
```

Examples for AIS for standing orders

Request

```
GET https://api.testbank.com/v1/accounts/qwer3456tzui7890/transactions?
bookingStatus=information
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 06 Aug 2017 15:05:45 GMT
Accept:                 application/json
```

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":
     [{
       "creditorName": "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"
      }]
    }
}
```

6.5.5 Read Transaction Details

Call

```
GET /v1/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 the PATH might be already given in detail 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/v1/accounts/qwer3456tzui7890/transactions/1234567
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 06 Aug 2017 15:05:46 GMT
```

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",
       "creditorName": "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.

6.6 Read Card Account Data Requests

6.6.1 Read Card Account List

Call

```
GET /v1/card-accounts
```

Reads a list of card 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 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.
Consent-ID	String	Mandatory	Identification of the corresponding 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

Attribute	Type	Condition	Description
			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	

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": "/v1/card-accounts/3d9a81b3-a47d-4130-8765-
a9c0ff861b99/transactions"
        }
      }
    }
  ]
}
```


6.6.2 Read Card Account Details

Call

```
GET /v1/card-accounts/{account-id}
```

Reads details about a card 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",
      "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": "/v1/card-accounts/3d9a81b3-a47d-4130-8765-
a9c0ff861b99/transactions"
        }
      }
    }
}
```


6.6.3 Read Card Account Balance

Call

```
GET /v1/card-accounts/{account-id}/balances
```

Reads balance data from a given card 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 6.6.1.

The account-id is constant at least throughout the lifecycle of a given consent.

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.
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	optional	Identifier of the addressed card account. **Remark for Future**: It is recommended to use this data element. The condition might change to "mandatory" in a next version of the specification.
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"},
  "balances":[
    {
      "balanceAmount": { "currency": "EUR", "amount": "14355.78"},
      "balanceType": "interimBooked"
    },{
      "balanceAmount": { "currency": "EUR", "amount": "4175.86"},
      "balanceType": "nonInvoiced",
    }
  ]
}
```

6.6.4 Read Card Account Transaction List

Call

GET /v1/card-accounts/{account-id}/transactions {query-parameters}

Reads account data from a given card 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 6.6.1.

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 the PATH might be already given in detail 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 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 "now" if not given.
bookingStatus	String	Mandatory	Permitted codes are "booked", "pending" and "both"

Attribute	Type	Condition	Description
			"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.

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	optional	Identifier of the addressed card account. **Remark for Future**: It is recommended to use this data element. The condition might change to "mandatory" in a next version of the specification.
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 report might be downloaded from in case where transaction reports have a huge size.

Example

```
GET https://api.testbank.com/v1/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
```

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"
  },
  "cardTransactions": {
    "booked": [
      {
        "cardTransactionId": "201710020036959",
        "transactionAmount": { "currency": "EUR", "amount": "256.67" },
        "transactionDate": "2017-10-25",
        "bookingDate": "2017-10-26",
        "originalAmount": { "currency": "SEK", "amount": "2499" },
        "cardAcceptorAddress": {
          "city" : "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": {
          "city" : "STOCKHOLM",
          "country" : "SE"
        },
        "maskedPan": "525412******8999",
        "proprietaryBankTransactionCode" : "PURCHASE",
        "invoiced": false,
        "transactionDetails": "ICA SUPERMARKET SKOGHA"
```

```
      }
    ],
    "pending": [ ],
    "_links": {
      "cardAccount": {
        "href": "/v1/card-accounts/3d9a81b3-a47d-4130-8765-a9c0ff861b99"
      }
    }
  }
}
```

7 Processes used commonly in AIS and PIS Services

Processes on starting authorisations, update PSU identification or PSU authentication data and explicit authorisation of transactions by using SCA are very similar in PIS and AIS services. The API calls supporting these processes are described in the following independently from the service/endpoint. For reasons of clarity, the endpoints are defined always for the Payment Initiation Service, the Payment Cancellation, the Signing Basket function and the Account Information Service separately. These processes usually are used following a hyperlink of the ASPSP. The usage is defined at the beginning of the following sections.

7.1 Start Authorisation Process

Usage

The start authorisation process is a process which is needed for creating a new authorisation or cancellation sub-resource. This applies in the following scenarios:

- The ASPSP has indicated with an "startAuthorisation" hyperlink in the pre-ceeding Payment Initiation Response that an explicit start of the authorisation process is needed by the TPP. The "startAuthorisation" hyperlink can transport more information about data which needs to be uploaded by using the extended forms
  - "startAuthorisationWithPsuIdentfication",
  - "startAuthorisationWithPsuAuthentication",
  - "startAuthorisationWithEncryptedPsuAuthentication",
  - "startAuthorisationWithAuthentciationMethodSelection"
- The related payment initiation cannot yet be executed since a multilevel SCA is mandated.
- The ASPSP has indicated with an "startAuthorisation" hyperlink in the pre-ceeding Payment Cancellation Response that an explicit start of the authorisation process is needed by the TPP. The "startAuthorisation" hyperlink can transport more information about data which needs to be uploaded by using the extended forms as indicated above.
- The related payment cancellation request cannot be applied yet since a multilevel SCA is mandate for executing the cancellation.
- The signing basket needs to be authorised yet.

Call in the context of a Payment Initiation Request

```
POST /v1/{payment-service}/{payment-product}/{paymentId}/authorisations
```

Starts the authorisation process for a payment initiation.

Call in the context of a Payment Cancellation Request

```
POST /v1/{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations
```

Starts the authorisation process for a payment cancellation where needed.

Call in context of an Account Information Consent Request

```
POST /v1/consents/{consentId}/authorisations
```

Starts an authorisation process for establishing account information consent data on the server.

Call in the context of a Signing Basket Authorisation Request

```
POST /v1/signing-baskets/{basketId}/authorisations
```

Starts the authorisation process for all transactions contained in the related signing basket.

Path Parameters

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, basketId or consentId	String	Resource identification of the related payment initiation, signing basket or consent resource.

Query Parameters

No specific 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-ID	String	Optional	Client ID of the PSU in the ASPSP client interface. Shall be transmitted if this Request is indicated by "startAuthorisationWithPsuIdentification" or "startAuthorisationWithPsuAuthentication" or "startAuthorisationWithEncryptedPsuAuthentication" and this field has not yet been transmitted before.
PSU-ID-Type	String	Optional	Type of the PSU-ID, needed in scenarios where PSUs have several PSU-IDs as access possibility. Shall be transmitted in this case, if this Request is indicated by "startAuthorisationWithPsuIdentification" or "startAuthorisationWithPsuAuthentication" or "startAuthorisationWithEncryptedPsuAuthentication" and this field has not yet been transmitted before.
PSU-Corporate-ID	String	Optional	Identification of a Corporate in the Online Channels. Shall be transmitted if this Request is indicated by "startAuthorisationWithPsuIdentification" or "startAuthorisationWithPsuAuthentication" or "startAuthorisationWithEncryptedPsuAuthentication" and this field has not yet been transmitted before, and only where generally needed in a corporate context.
PSU-Corporate-ID-Type	String	Optional	This is describing the type of the identification needed by the ASPSP to identify the PSU-Corporate-ID content. Shall be transmitted if this Request is indicated by "startAuthorisationWithPsuIdentification". or "startAuthorisationWithPsuAuthentication" or "startAuthorisationWithEncryptedPsuAuthentication" and this field has not yet been transmitted before. Mean and use is defined in the ASPSP's documentation. Only used in a corporate context.

Attribute	Type	Condition	Description
Authorization	String	Conditional	Bearer Token. Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in an preceding AIS service in the same session.
TPP-Redirect-Preferred	Boolean	Optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach. If it equals "false", the TPP prefers not to be redirected for SCA. The ASPSP will then choose between the Embedded or the Decoupled SCA approach, depending on the choice of the SCA procedure by the TPP/PSU. If the parameter is not used, the ASPSP will choose the SCA approach to be applied depending on the SCA method chosen by the TPP/PSU.
TPP-Redirect-URI	String	Conditional	URI of the TPP, where the transaction flow shall be redirected to after a Redirect. Mandated for the Redirect SCA Approach, specifically when TPP-Redirect-Preferred equals "true". See Section 4.10 for further requirements on this header. This field may be ignored by the ASPSP for migration reasons. For this reason, the same TPP-Redirect-URI as used when creating the related resource shall be provided by the TPP. This specifically applies to the authorisation of a payment cancellation, where the same TPP-Redirect-URI as for the corresponding payment initiation shall be used. This applies also to multilevel SCA, where the TPP-Redirect-URI for all authorisation processes for one transaction shall be equal. It is recommended to always use this header field. **Remark for Future**: This field might be changed to mandatory in the next version of the specification. **Remark for Future:** The condition on keeping the TPP-Redirect-URI equal during a transaction

Attribute	Type	Condition	Description
			lifecycle might be removed in the next version of the specification.
TPP-Nok-Redirect-URI	String	Optional	If this URI is contained, the TPP is asking to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the redirect SCA method. This may be ignored by the ASPSP. See Section 4.10 for further requirements on this header. The same condition as for TPP-Redirect-URI on keeping the URI equal during a transaction lifecycle applies also to this header.

Request Body

No request body.

Note: If the hyperlinks in the following extended forms are used in the response message before, additional conditions on request body parameters apply as indicated in the following:

- "startAuthorisationWithPsuIdentification": Cp. Section 7.2.1
- "startAuthorisationWithPsuAuthentication": Cp. Section 7.2.2
- "startAuthorisationWithEncryptedPsuAuthentication": Cp. Section 7.2.2.
- "startAuthorsiationWithAuthenticationMethodSelection": Cp. Section 7.2.3.

The differences in the calls then are only whether to use a POST command to create the authorisation sub-resource and update the specified data at the same time or to use a PUT command to update the specified data to an already created sub-resource.

Response Code

HTTP response code equals 201.

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.

Attribute	Type	Condition	Description
ASPSP-SCA-Approach	String	Conditional	Possible values are: • EMBEDDED • DECOUPLED • REDIRECT OAuth will be subsumed by the value REDIRECT

Response Body

Attribute	Type	Condition	Description
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.
currencyConversion Fees	Amount	Optional	Might be used by the ASPSP to transport specific currency conversion fees related to the initiated credit transfer.
estimatedTotalAmount	Amount	Optional	The amount which is estimated to be debted from the debtor account. Note: This amount includes fees.
estimatedInterbank SettlementAmount	Amount	Optional	The estimated amount to be transferred to the payee.
scaStatus	SCA Status	Mandatory	
authorisationId	String	Mandatory	Unique resource identification of the created authorisation sub-resource.

Attribute	Type	Condition	Description
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 "selectAuthenticationMethod" 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 "updatePsuAuthentication" or "updateEncryptedPsuAuthentication" link.

Attribute	Type	Condition	Description
_links	Links	Mandatory	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP. Type of links admitted in this response, (further links might be added for ASPSP defined extensions): "scaRedirect": In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser.
			"scaOAuth": In case of a SCA OAuth2 Approach, the ASPSP is transmitting the URI where the configuration of the Authorisation Server can be retrieved. The configuration follows the OAuth 2.0 Authorisation Server Metadata specification. "confirmation": Might be added by the ASPSP if either the "scaRedirect" or "scaOAuth" hyperlink is returned in the same response message. This hyperlink defines the URL to the resource which needs to be updated with • a confirmation code as retrieved after the plain redirect authentication process with the ASPSP authentication server or • an access token as retrieved by submitting an authorization code after the integrated OAuth based authentication process with the ASPSP authentication server.

Attribute	Type	Condition	Description
			"updatePsuIdentification": The link to the authorisation or cancellation authorisation sub-resource, where PSU identification data needs to be uploaded. "udpatePsuAuthentication": The link to the authorisation or cancellation authorisation sub-resource, where PSU authentication data needs to be uploaded. "udpateEncryptedPsuAuthentication": The link to the authorisation or cancellation authorisation sub-resource, where encrypted PSU authentication data needs to be uploaded "selectAuthenticationMethod": The link to the authorisation or cancellation authorisation sub-resource, where the selected authentication method needs to be uploaded. This link is contained under exactly the same conditions as the data element "scaMethods" "authoriseTransaction": The link to the authorisation or cancellation authorisation sub-resource, where the authorisation data has to be uploaded, e.g. the TOP received by SMS. "scaStatus": The link to retrieve the scaStatus of the corresponding authorisation sub-resource.
psuMessage	Max500Text	Optional	

Note: If the hyperlinks in the following extended forms are used in the response message before, additional response parameters apply as indicated in the following:

- In case of "startAuthorisationWithPsuIdentification": Cp. Section 7.2.1
- In case of: "startAuthorisationWithPsuAuthentication": Cp. Section 7.2.2
- In case of: "startAuthorisationWithEncryptedPsuAuthentication": Cp. Section 7.2.2
- In case of: "startAuthorisationWithAuthenticationMethodSelection": Cp. Section 7.2.3.

Example

Request

```
POST https://api.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-ID:                 PSU-1234
```

Response

```
HTTP/1.x 201 CREATED
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     DECOUPLED
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Location:               https://www.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations/123auth456
Content-Type:           application/json
{
 "scaStatus": "received",
 "authorisationId": "123auth456",
 "psuMessage": "Please use your BankApp for transaction Authorisation.",
 "_links":{
   "scaStatus":  {"href":"/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/
authorisations/123auth456"}
   }
}
```

7.2 Update PSU Data

There are several possible Update PSU Data requests needed, which depends on the SCA Approach:

- Redirect SCA Approach: A specific Update PSU Data Request is applicable for
  - the selection of authentication methods, before choosing the actual SCA approach.
- Decoupled SCA Approach: A specific Update PSU Data Request is only applicable for
  - adding the PSU Identification, if not provided yet in the Payment Initiation Request or the Account Information Consent Request, or if no OAuth2 access token is used, or
  - the selection of authentication methods.
- Embedded SCA Approach: The Update PSU Data Request might be used
  - to add credentials as a first factor authentication data of the PSU and
  - to select the authentication method.

The SCA Approach might depend on the chosen SCA method. For that reason, the following possible Update PSU Data request can apply to all SCA approaches:

- Select an SCA method in case of several SCA methods are available for the customer.

These different Update PSU Data Requests are differentiated in the following sub sections.

7.2.1 Update PSU Data (Identification)

This call is used, when in the preceding call the hyperlink of type "updatePsuIdentification" was contained, e.g. in case of a Decoupled Approach in the response and is now followed by the TPP.

Call in the context of a Payment Initiation Request

```
PUT /v1/{payment-service}/{payment-
product}/{paymentId}/authorisations/{authorisationId}
```

Updates the payment initiation authorisation sub-resource data on the server by PSU data, if requested by the ASPSP.

Call in the context of a Payment Cancellation Request

```
PUT /v1/{payment-service}/{payment-product}/{paymentId}/cancellation-
authorisations/{authorisationId}
```

Updates the payment initiation cancellation authorisation sub-resource data on the server by PSU data, if requested by the ASPSP.

Call in case of an Account Information Consent Request

```
PUT /v1/consents/{consentId}/authorisations/{authorsationId}
```

Updates the account information consent authorisation data on the server by PSU data, if requested by the ASPSP.

Call in the context of a Signing Basket Authorisation Request

```
PUT /v1/signing-baskets/{basketId}/authorisations/{authorisationId}
```

Updates the signing basket authorisation data on the server by PSU data, if requested by the ASPSP.

Path Parameters

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, basketId or consentId	String	Resource identification of the related payment initiation, signing basket or consent resource.
authorisationId	String	Resource identifciation of the related Payment Initiation, Payment cancellation, Signing Basket or Consent authorisation sub-resource.

Query Parameters

No specific 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-ID	String	Conditional	Contained if not yet contained in a pre-ceeding request, and mandated by the ASPSP in the related response
PSU-ID-Type	String	Conditional	Type of the PSU-ID, needed in scenarios where PSUs have several PSU-IDs as access possibility.
PSU-Corporate-ID	String	Conditional	Contained if not yet contained in a pre-ceeding request, and mandated by the ASPSP in the related response. This field is relevant only in a corporate context.
PSU-Corporate-ID-Type	String	Conditional	Might be mandated by the ASPSP in addition if the PSU-Corporate-ID is contained.

Request Body

No request body.

Response Code

HTTP response code is 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.
ASPSP-SCA-Approach	String	Conditional	Possible values are: • EMBEDDED • DECOUPLED • REDIRECT OAuth will be subsumed by the value REDIRECT

Response Body

Attribute	Type	Condition	Description
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.
currencyConversion Fees	Amount	Optional	Might be used by the ASPSP to transport specific currency conversion fees related to the initiated credit transfer.
estimatedTotalAmount	Amount	Optional	The amount which is estimated to be debted from the debtor account. **Note:** This amount includes fees.
estimatedInterbank SettlementAmount	Amount	Optional	The estimated amount to be transferred to the payee.
scaMethods	Array of authentication objects	Conditional	Might be contained, if several authentication methods are available. (name, type)
_links	Links	Mandatory	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP. Type of links admitted in this response, (further links might be added for ASPSP defined extensions): "scaStatus": The link to retrieve the scaStatus of the corresponding authorisation sub-resource.

Attribute	Type	Condition	Description
			"selectAuthenticationMethod": This is a link to a resource, where the TPP can select the applicable second factor authentication methods for the PSU, if there are several available authentication methods and if the PSU is already sufficiently authenticated.. If this link is contained, then there is also the data element "scaMethods" contained in the response body
scaStatus	SCA Status	Mandatory	
psuMessage	Max500Text	Optional	

Example

Request

```
PUT https://api.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations/123auth456
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-ID:                 PSU-1234
```

Response

```
HTTP/1.x 200 OK
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     DECOUPLED
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json
{
 "scaStatus": "psuIdentified",
 "psuMessage": "Please use your BankApp for transaction Authorisation.",
 "_links":{
   "scaStatus":  {"href":"/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/
authorisations/123auth456"}
   }
}
```

7.2.2 Update PSU Data (Authentication) in the Decoupled or Embedded Approach

This call is used, when in the preceding call the hyperlink of type "updatePsuAuthentication", "updateEncryptedPsuAuthentication", "updateAdditionalPsuAuthentication" or "updateAdditionalEncryptedPsuAuthentication" was contained in the response and is followed by the TPP.[8]

Call in context of a Payment Initiation

```
PUT /v1/{payment-service}/{payment-
product}/{paymentId}/authorisations/{authorisationId}
```

Updates the payment initiation authorisation sub-resource data on the server by PSU credential data, if requested by the ASPSP

Call in context of a Payment Cancellation

```
PUT /v1/{payment-service}/{payment-product}/{paymentId}/cancellation-
authorisations/{authorisationId}
```

Updates the payment cancellation authorisation sub-resource data on the server by PSU credentials, if requested by the ASPSP

Call in context of an Account Information Consent Request

```
PUT /v1/consents/{consentId}/authorisations/{authorisationId}
```

Updates the account information consent authorisation sub-resource data on the server by PSU credential data, if requested by the ASPSP

Call in the context of a Signing Basket Authorisation Request

PUT /v1/signing-baskets/{basketId}/authorisations/{authorisationId}

Updates the signing basket authorisation data on the server by PSU credentials, if requested by the ASPSP.

Remark for Future: The next version of the specification might allow ASPSPs to mandate a payload encryption to protect the password contained in the payload.

[8] The next release of this specification might support encryption methods for transmission of the PSU password between TPP and ASPSP on application level.

Path Parameters

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, basketId or consentId	String	Resource identification of the related payment initiation, signing basket or consent resource.
authorisationId	String	Resource identifciation of the related Payment Initiation, Payment Cancellation, Signing Basket or Consent authorisation sub-resource.

Query Parameters

No specific 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-ID	String	Conditional	Contained if not yet contained in a pre-ceeding request, and mandated by the ASPSP in the related response
PSU-ID-Type	String	Conditional	Contained if not yet contained in a pre-ceeding request, and mandated by the ASPSP in the related response
PSU-Corporate-ID	String	Conditional	Contained if not yet contained in a pre-ceeding request, and mandated by the ASPSP in the related response. This field is relevant only in a corporate context.

Attribute	Type	Condition	Description
PSU-Corporate-ID-Type	String	Conditional	Contained if not yet contained in a pre-ceeding request, and mandated by the ASPSP documentation. Might be mandated by the ASPSP in addition if the PSU-Corporate-ID is contained.

Request Body

Attribute	Type	Condition	Description
psuData	PSU Data	Mandatory	The password, encryptedPassword, additionalPassword, or additionalEncryptedPassword subfield is used, depending whether the password or the additional password needs to be sent and depending on encryption requirements of the ASPSP as indicated in the corresponding hyperlink contained in the preceding response message of the ASPSP. **Remark for Future**: More details on the encrypted password transport will be published by a future bulletin.

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.

Attribute	Type	Condition	Description
ASPSP-SCA-Approach	String	Conditional	Possible values are: • EMBEDDED • DECOUPLED • REDIRECT OAuth will be subsumed by the value REDIRECT

Response Body

Attribute	Type	Condition	Description
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.
currencyConversionFees	Amount	Optional	Might be used by the ASPSP to transport specific currency conversion fees related to the initiated credit transfer.
estimatedTotalAmount	Amount	Optional	The amount which is estimated to be debted from the debtor account. Note: This amount includes fees.
estimatedInterbankSettlementAmount	Amount	Optional	The estimated amount to be transferred to the payee.
chosenScaMethod	Authentication object	Conditional	A definition of the provided SCA method is contained, if only one authentication method is available, and if the Embedded SCA approach is chosen by the ASPSP.
challengeData	Challenge	Conditional	Challenge data might be contained, if only one authentication method is available, and if the Embedded SCA approach is chosen by the ASPSP.

Attribute	Type	Condition	Description
scaMethods	Array of authentication objects	Conditional	Might be contained, if several authentication methods are available. (name, type)
_links	Links	Conditional	A list of hyperlinks to be recognised by the TPP. Might be contained, if several authentication methods are available for the PSU. Type of links admitted in this response: "updateAdditionalPsuAuthentication" The link to the payment initiation or account information resource, which needs to be updated by an additional PSU password. This link is only contained in rare cases, where such additional passwords are needed for PSU authentications. "updateAdditionalEncryptedPsuAuthentication" The link to the payment initiation or account information resource, which needs to be updated by an additional encrypted PSU password. This link is only contained in rare cases, where such additional passwords are needed for PSU authentications. "selectAuthenticationMethod": This is a link to a resource, where the TPP can select the applicable second factor authentication methods for the PSU, if there were several available authentication methods. This link is only contained, if the PSU is already identified or authenticated with the first relevant factor or alternatively an access token, if SCA is required and if the PSU has a choice between different authentication methods. If this link is contained, then there is also the data element "scaMethods" contained in the response body "authoriseTransaction": The link to the resource, where the "Transaction

Attribute	Type	Condition	Description
			Authorisation Request" is sent to. This is the link to the resource which will authorise the transaction by checking the SCA authentication data within the Embedded SCA approach. "scaStatus": The link to retrieve the scaStatus of the corresponding authorisation sub-resource.
scaStatus	SCA Status	Mandatory	
psuMessage	Max500Text	Optional	

NOTE: In case of an incorrect password, the TPP needs to ask the PSU for re-entering the password. The newly entered password needs to be updated to the same path. It is recommended that the ASPSP is informing the TPP about this by adding a _links section in the additional error information and presenting a corresponding updatePsuAuthentication or updateEncryptedPsuAuthentication hyperlink.

Example

Request in case of Embedded Approach

```
PUT https://api.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations/123auth456
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-ID:                 PSU-1234
{
"psuData": {
  "password": "start12"
  }
}
```

Response in case of the embedded approach

```
HTTP/1.x 200 OK
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     EMBEDDED
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json

{
  "scaStatus": "psuAuthenticated",
  "_links":{
   "authoriseTransaction": {"href": "/v1/payments/sepa-credit-
transfers/1234-wertiq-983/authorisations/123auth456"}
  }
}
```

7.2.3 Update PSU Data (Select Authentication Method)

This call is used, when in the preceding call the hyperlink of type "selectAuthenticationMethod" was contained in the response and was followed by the TPP.

Call in context of a Payment Initiation Request

```
PUT /v1/{payment-service}/{payment-
product}/{paymentId}/authorisations/{authorisationId}
```

Updates the payment initiation sub-resource data on the server by PSU data, if requested by the ASPSP.

Call in context of a Payment Cancellation Request

```
PUT /v1/{payment-service}/{payment-product}/{paymentId}/cancellation-
authorisations/{authorisationId}
```

Updates the payment cancellation sub-resource data on the server by PSU data, if requested by the ASPSP.

Call in context of an Account Information Consent Request

```
PUT /v1/consents/{consentId}/authorisations/{authorisationId}
```

Updates the account information consent authorisation data on the server by PSU data, if requested by the ASPSP

Call in the context of a Signing Basket Authorisation Request

```
PUT /v1/signing-baskets/{basketId}/authorisations/{authorisationId}
```

Updates the signing basket authorisation data on the server by PSU data, if requested by the ASPSP.

Path Parameters

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.
payment -product	String	Only in case of an Update Data Request in a Payment Initiation context.
paymentId, basketId or consentId	String	Resource identification of the related payment initiation, signing basket or consent resource.
authorisationId	String	Resource identiiciation of the related Payment Initiation, Payment Cancellation, Signing Basket or Consent authorisation sub-resource.

Query Parameters

No specific query parameters.

Response Code

The 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.

Request Body

Attribute	Type	Condition	Description

authentication MethodId	String	Mandatory	The authentication method ID as provided by the ASPSP.

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.
ASPSP-SCA-Approach	String	Optional	Possible values are: • EMBEDDED • DECOUPLED • REDIRECT OAuth will be subsumed by the constant REDIRECT

Response Body

Attribute	Type	Condition	Description
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.
currencyConversion Fees	Amount	Optional	Might be used by the ASPSP to transport specific currency conversion fees related to the initiated credit transfer.
estimatedTotalAmount	Amount	Optional	The amount which is estimated to be debted from the debtor account. **Note:** This amount includes fees.

Attribute	Type	Condition	Description
estimatedInterbank SettlementAmount	Amount	Optional	The estimated amount to be transferred to the payee.
chosenSca Method	Authentication object	Conditional	A definition of the provided SCA method is contained, if the Embedded SCA approach is chosen by the ASPSP.
challengeData	Challenge	Conditional	Challenge data might be contained, if the Embedded SCA approach is chosen by the ASPSP.

_links	Links	Conditional	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP. **Remark**: This method can be applied before or after PSU identification. This leads to many possible hyperlink responses. Type of links admitted in this response, (further links might be added for ASPSP defined extensions): "scaRedirect": In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser. "scaOAuth": In case of a SCA OAuth2 Approach, the ASPSP is transmitting the URI where the configuration of the Authorisation Server can be retrieved. The configuration follows the OAuth 2.0 Authorisation Server Metadata specification. "confirmation": Might be added by the ASPSP if either the "scaRedirect" or "scaOAuth" hyperlink is returned in the same response message. This hyperlink defines the URL to the resource which needs to be updated with • a confirmation code as retrieved after the plain redirect authentication process with the ASPSP authentication server or • an access token as retrieved by submitting an authorization code after the integrated

Attribute	Type	Condition	Description
			OAuth based authentication process with the ASPSP authentication server.
			"updatePsuIdentification": The link to the authorisation or cancellation authorisation sub-resource, where PSU identification data needs to be uploaded. "updatePsuAuthentication": The link to the authorisation or cancellation authorisation sub-resource, where PSU authentication data needs to be uploaded. "updateEncryptedPsuAuthentication": The link to the authorisation or cancellation authorisation sub-resource, where encrypted PSU authentication data needs to be uploaded. "authoriseTransaction": The link to the authorisation or cancellation authorisation sub-resource, where the authorisation data has to be uploaded, e.g. the TOP received by SMS. "scaStatus": The link to retrieve the scaStatus of the corresponding authorisation sub-resource.
scaStatus	Sca Status	Mandatory	
psuMessage	Max500Text	Optional	

Example

Request in case of Embedded Approach

```
PUT https://api.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations/123auth456
X-Request-ID:           asdfoeljkasdfoelkjasdf-123479093

{
authenticationMethodId: "myAuthenticationID"
}
```

Response in case of the embedded approach

```
HTTP/1.x 200 OK
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     EMBEDDED
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json

{
  "scaStatus": "scaMethodSelected",
  "chosenScaMethod": {
      "authenticationType": "SMS_OTP",
      "authenticationMethodId": "myAuthenticationID"},
  "challengeData": {
      "otpMaxLength": "6",
      "otpFormat": "integer"},
  "_links": {
     "authoriseTransaction": {"href": "/v1/payments/sepa-credit-
transfers/1234-wertiq-983/authorisations/123auth456"}
  }
}
```

7.3 Transaction Authorisation

This call is only used in case of an Embedded SCA Approach.

Call in context of a Payment Initiation Request

```
PUT /v1/payments/{payment-
product}/{paymentId}/authorisations/{authorisationId}
```

Transmit response data to the challenge for SCA checks by the ASPSP.

Call in context of a Payment Cancellation Request

```
PUT /v1/payments/{payment-product}/{paymentId}/cancellation-
authorisations/{authorisationId}
```

Transmit response data to the challenge for SCA checks by the ASPSP.

Call in context of an Account Information Consent Request

```
PUT /v1/consents/{consentId}/authorisation/{authorisationId}
```

Transfers response data to the challenge for SCA checks by the ASPSP.

Call in the context of a Signing Basket Authorisation Request

```
PUT /v1/signing-baskets/{basketId}/authorisations/{authorisationId}
```

Transfers response data to the challenge for SCA checks by the ASPSP.

Path Parameters

Attribute	Type	Description
payment -product	String	The related payment product of the payment initiation to be authorized.
paymentId, basketId or consentId	String	Resource identification of the related payment initiation, signing basket or consent resource.
authorisationId	String	Resource identification of the related Payment Initiation, Payment Cancellation, Signing Basket or Consent authorisation sub-resource.

Query Parameter

No specific 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.

Attribute	Type	Condition	Description
Authorization	String	Conditional	Is contained only, if the optional Oauth Pre-Step was performed.

Request Body

Attribute	Type	Condition	Description
scaAuthenticationData	String	Mandatory	SCA authentication data, depending on the chosen authentication method. If the data is binary, then it is base64 encoded.

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
scaStatus	SCA Status	Mandatory	

NOTE: In case of incorrect scaAuthenticationData, the TPP needs to ask the PSU for re-entering the authentication data by repeating the SCA method first. Depending on the implementation of the corresponding SCA method, the TPP needs

- either to re-start the full authorisation process by generating a new authorisation sub-resource, e.g. in case of an SMS OTP,
- or to submit newly generated authentication data generated on a customer device to the same path as the first time, and where no new challenge data from the ASPSP is needed, e.g. in case of a CHIP OTP.

The ASPSP is informing the TPP about this by adding a _links section in the additional error information and presenting a corresponding startAuthorisation, or transactionAuthorisation hyperlink.

Example

Request

```
PUT https://api.testbank.com/v1/payments/sepa-credit-transfers/
qwer3456tzui7890/authorisations/123auth456
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
{
 "scaAuthenticationData": "123456"
}
```

Response in case of the embedded approach

Response Code 200

Response Body

```
{
  "scaStatus": "finalised",
  "_links":{
   "scaStatus":  {"href":"/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations/123auth456"}
   }

}
```

7.4 Get Authorisation Sub-Resources Request

Call in context of a Payment Initiation Request

```
GET /v1/{payment-service}/{payment-product}/{paymentId}/authorisations
```

Will deliver an array of resource identifications of all generated authorisation sub-resources.

Call in context of an Account Information Consent Request

```
GET /v1/consents/{consentId}/authorisations
```

Will deliver an array of resource identifications of all generated authorisation sub-resources.

Call in the context of a Signing Basket Authorisation Request

```
GET /v1/signing-baskets/{basketId}/authorisations
```

Will deliver an array of resource identifications of all generated authorisation sub-resources.

Path Parameters

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, basketId or consentId	String	Resource identification of the related payment initiation, signing basket or consent resource.

Query Parameters

No specific 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.
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 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
authorisationIds	Array of String	Mandatory	An array of all authorisationIds connected to this payment, signing basket or consent resource.

Example

Request

```
GET https://api.testbank.com/v1/payments/sepa-credit-transfers/1234-wertiq-
983/authorisations
Accept:                 application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7723
Date:                   Sun, 06 Aug 2017 15:04:07 GMT
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7723
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
Content-Type:           application/json

{
  "authorisationIds": ["123auth456"]
}
```

7.5 Get SCA Status Request

Call in context of a Payment Initiation Request

```
GET /v1/{payment-service}/{payment-
product}/{paymentId}/authorisations/{authorisationId}
```

Checks the SCA status of an authorisation sub-resource.

Call in context of a Payment Cancellation Request

```
GET /v1/{payment-service}/{payment-product}/{paymentId}/cancellation-
authorisations/{authorisationId}
```

Checks the SCA status of a cancellation authorisation sub-resource.

Call in context of an Account Information Consent Request

```
GET /v1/consents/{consentId}/authorisations/{authorisationId}
```

Checks the SCA status of a authorisation sub-resource.

Call in the context of a Signing Basket Authorisation Request

```
GET /v1/signing-baskets/{basketId}/authorisations/{authorisationId}
```

Checks the SCA status of a authorisation sub-resource.

Path Parameters

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, basketId or consentId	String	Resource identification of the related payment initiation, signing basket or consent resource.

Attribute	Type	Description
authorisationId	String	Resource identification of the related Payment Initiation, Payment Cancellation, Signing Basket or Consent authorisation sub-resource.

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
scaStatus	SCA Status	Mandatory	This data element is containing information about the status of the SCA method applied.

Example

Request

```
GET https://api.testbank.com/v1/payments/sepa-credit-transfers/1234-wertiq-
983/authorisations/123auth456
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

{
  "scaStatus": "finalised"
}
```

7.6 Confirmation Request

This request is used, when in the preceding response the hyperlink of type "confirmation" was contained and if a redirection authentication method has been applied. Before the call can be submitted by the TPP, an authorization code, respectively a confirmation code needs to be retrieved by the TPP after the SCA processing in a redirect to the ASPSP authentication server.

In case of the integrated OAuth SCA Approach, the overall procedure to receive the authorization code and the access token succedingly is described in Section 13.

In case of the Redirect SCA Approach, the procedure to retrieve the confirmation code is described in the following sub sections. The actual Confirmation Request Message is

described in Section 7.6.4 for both the integrated OAuth2 SCA approach and the Redirect SCA Approach.

7.6.1 Retrieving the Confirmation Code in Redirect SCA approach

The TPP needs to fix the session of the PSU on the TPP browser with a nonce, where part of it is a unique state parameter.

In preparation of sending the authorization request, the TPP shall

- create a one-time use XSRF token to be conveyed to the ASPSP in the “state” parameter and,
- bind this value to the current session in the user agent.

Note: In case of the integrated OAuth SCA Approach, the TPP has to generate in addition a nonce for the challenge parameter. This has also to be bound to the session of the user agent.

7.6.2 Requirements on HTTP request of PSU browser

The TPP needs to forward the state parameter as query parameter to the PSU, which will lead to a GET HTTP request of the PSU browser as required as follows:

Query Parameter PSU Authorisation Request (GET command)

Attribute	Type	Condition	Description
state	string	mandated	state parameter as defined by the TPP as a unique parameter and bound to the PSU/TPP session.

Example

```
GET ASPSP-Redirect-URI?state=1234567er
```

After the customer authentication has taken place on the ASPSP server, the ASPSP responds with the same state parameter and a unique confirmationCode bound to the authorisation resource as query parameters. The confirmationCode will only be contained if SCA has been successfully performed.

Query Parameter PSU Authorisation Response (GET command response)

Attribute	Type	Condition	Description
state	string	mandated	state parameter as used in the corresponding request.

Attribute	Type	Condition	Description
code	string	conditional	unique authorisation code of the ASPSP, bound to the related transaction, in case of Integrated OAuth SCA Approach.
confirmationCode	string	conditional	unique authorisation code of the ASPSP, bound to the related transaction, in case of Redirect SCA Approach.

Example in case of Redirect SCA Approach

```
http 302?state=1234567er&confirmationCode=2256ffgh
```

7.6.3 Confirmation Call Pre-Condition

When retrieving the GET command from the PSU browser, the TPP must check whether the state parameter is linked to the current session.The “state” value is linked to the current session in the user agent. If the check is positive then the TPP further processes

- within context of the Integrated OAuth SCA Approach with retrieving the access Bearer token as described in Section 13 of this document and then proceed as described in Section 7.6.4.
- within context of the Redirect SCA Approach directly as described in Section 7.6.4.

If the check fails, the transaction must be stopped by the TPP.

7.6.4 Authorisation Confirmation Call

Call in the context of a Payment Initiation Request

```
PUT /v1/{payment-service}/{payment-
product}/{paymentId}/authorisations/{authorisationId}
```

Updates the payment initiation authorisation sub-resource data on the server by an authorization code, if requested by the ASPSP.

Call in the context of a Payment Cancellation Request

```
PUT /v1/{payment-service}/{payment-product}/{paymentId}/cancellation-
authorisations/{authorisationId}
```

Updates the payment initiation cancellation authorisation sub-resource data on the server by an authorization code, if requested by the ASPSP.

Call in case of an Account Information Consent Request

```
PUT /v1/consents/{consentId}/authorisations/{authorsationId}
```

Updates the account information consent authorisation data on the server by an authorization code, if requested by the ASPSP.

Call in the context of a Signing Basket Authorisation Request

```
PUT /v1/signing-baskets/{basketId}/authorisations/{authorisationId}
```

Updates the signing basket authorisation data on the server by an authorisation code, if requested by the ASPSP.

Path Parameters

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, basketId or consentId	String	Resource identification of the related payment initiation, signing basket or consent resource.
authorisationId	String	Resource identification of the related Payment Initiation, Payment Cancellation, Signing Basket or Consent authorisation sub-resource.

Query Parameters

No specific 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.

Attribute	Type	Condition	Description
Authorization	String	Conditional	Authorization Bearer Token as retrieved by the TPP in case the integrated OAuthSCA Approach as described in Section 13.

Request Body

Attribute	Type	Condition	Description
confirmationCode	String	Conditional	Confirmation Code as retrieved by the TPP from the redirect based SCA process as described in Section 7.6.1 ff.

Response Code

HTTP response code is 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
scaStatus	SCA Status	Mandatory	Value "finalised" if the transaction authorisation and confirmation was successful. Value "failed" if the transaction authorisation or confirmation was not successful.

Attribute	Type	Condition	Description
_links	Links	Mandatory	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP. Type of links admitted in this response, (further links might be added for ASPSP defined extensions): "status": The link to retrieve the status of the corresponding transaction resource.
psuMessage	Max512Text	Optional	

Example for integrated OAuth solution

Request

```
PUT https://api.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations/123auth456
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Authorization:          Bearer 1234567
```

Response

```
HTTP/1.x 200 OK
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json
{
 "scaStatus": "finalised",
 "_links":{
   "status":  {"href":"/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/status"}
   }
```

Example for redirect solution

Request

```
PUT https://api.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/authorisations/123auth456
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
{ "confirmationCode": "2256ffgh"}
```

Response

```
HTTP/1.x 200 OK
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json
{
 "scaStatus": "finalised",
 "_links":{
   "status":  {"href":"/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/status"}
   }
}
```

8 Signing Baskets

8.1 Establish Signing Basket Request

`POST /v1/signing-baskets/`

Generates a signing basket

Path Parameters

None.

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.
PSU-ID	String	Conditional	Client ID of the PSU in the ASPSP client interface. Might be mandated in the ASPSP's documentation. It might be contained, even if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in an preceding AIS service in the same session. In this case the ASPSP might check whether PSU-ID and token match, according to ASPSP documentation.
PSU-ID-Type	String	Conditional	Type of the PSU-ID, needed in scenarios where PSUs have several PSU-IDs as access possibility. In this case, the mean and use is then defined in the ASPSP's documentation.
PSU-Corporate-ID	String	Conditional	Identification of a Corporate in the Online Channels Might be mandated in the ASPSP's documentation. Only used in a corporate context.

Attribute	Type	Condition	Description
PSU-Corporate-ID-Type	String	Conditional	This is describing the type of the identification needed by the ASPSP to identify the PSU-Corporate-ID content. Mean and use is defined in the ASPSP's documentation. Only used in a corporate context.
Authorization	String	Conditional	Bearer Token. Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in an preceding AIS service in the same session.
Consent-ID	String	Optional	This data element may be contained, if the signing basket transaction is part of a session, i.e. combined AIS/PIS service. This then contains the "consentId" of the related AIS one off consent, which was performed prior to this bulk signing.
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.
TPP-Redirect-Preferred	Boolean	Optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach. If it equals "false", the TPP prefers not to be redirected for SCA. The ASPSP will then choose between the Embedded or the Decoupled SCA approach, depending on the choice of the SCA procedure by the TPP/PSU. If the parameter is not used, the ASPSP will choose the SCA approach to be applied depending on the SCA method chosen by the TPP/PSU.
TPP-Redirect-URI	String	Conditional	URI of the TPP, where the transaction flow shall be redirected to after a Redirect. Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-

Attribute	Type	Condition	Description
			Preferred equals "true". See Section 4.10 for further requirements on this header. It is recommended to always use this header field. **Remark for Future**: This field might be changed to mandatory in the next version of the specification.
TPP-Nok-Redirect-URI	String	Optional	If this URI is contained, the TPP is asking to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the redirect SCA method. This might be ignored by the ASPSP. See Section 4.10 for further requirements on this header.
TPP-Explicit-Authorisation-Preferred	Boolean	Optional	Must equal "true", if contained. Remark: No optimisation processes for creating authorisation resources for signing baskets implicitly, since anyhow several calls have been submitted.
TPP-Notification-URI	String	Optional	URI for the Endpoint of the TPP-API to which the status of the basket should be sent. This header field **may by ignored** by the ASPSP, cp. also the extended service definition in [XS2A-RSNS].
TPP-Notification-Content-Preferred	String	Optional	The string has the form status=X1, …, Xn where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated. The usage of the constants supports the following semantics: SCA: A notification on every change of the scaStatus attribute for all related authorisation processes is preferred by the TPP.

Attribute	Type	Condition	Description
			PROCESS: A notification on all changes of consentStatus or transactionStatus attributes is preferred by the TPP. LAST: Only a notification on the last consentStatus or transactionStatus as available in the XS2A interface is preferred by the TPP. This header field may be ignored, if the ASPSP does not support resource notification services for the related TPP.

Request Body

Attribute	Type	Condition	Description
paymentIds	Array of String	Optional	A non empty array of paymentIds
consentIds	Array of String	Optional	A non empty array of consentIds

The body shall contain at least one entry.

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.

Attribute	Type	Condition	Description
ASPSP-SCA-Approach	String	Conditional	This data element must be contained, if the SCA Approach is already fixed. Possible values are: • EMBEDDED • DECOUPLED • REDIRECT The OAuth SCA approach will be subsumed by REDIRECT.
ASPSP-Notification-Support	Boolean	Conditional	true if the ASPSP supports resource status notification services. false if the ASPSP supports resource status notification in general, but not for the current request. Not used, if resource status notification services are generally not supported by the ASPSP. Shall be supported if the ASPSP supports resource status notification services, see more details in the extended service definition [XS2A-RSNS].

Attribute	Type	Condition	Description
ASPSP-Notification-Content	String	Conditional	The string has the form status=X1, …, Xn where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated. The usage of the constants supports the following semantics: SCA: Notification on every change of the scaStatus attribute for all related authorisation processes is provided by the ASPSP for the related resource. PROCESS: Notification on all changes of consentStatus or transactionStatus attributes is provided by the ASPSP for the related resource. LAST: Notification on the last consentStatus or transactionStatus as available in the XS2A interface is provided by the ASPSP for the related resource. This field must be provided if the ASPSP-Notification-Support =true. The ASPSP might consider the notification content as preferred by the TPP, but can also respond independently of the preferred request.

Response Body

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	Mandatory	The non payment related values defined in Section 14.24 might be used like RCVD or ACTC. For a list of all transactionStatus codes permitted for signing baskets, cp. Section 8.3.

Attribute	Type	Condition	Description
basketId	String	Mandatory	resource identification of the generated signing basket 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 AuthenticationMethodSelection" contained in the response body. These methods shall be presented towards the PSU for selection by the TPP.
chosenScaMethod	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 PsuAuthentication" or "startAuthorisationWith PsuAuthentication" link.
_links	Links	Mandatory	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP.

Attribute	Type	Condition	Description
			Type of links admitted in this response, (further links might be added for ASPSP defined extensions): "startAuthorisation": In case, where an explicit start of the transaction authorisation is needed, but no more data needs to be updated (no authentication method to be selected, no PSU identification nor PSU authentication data to be uploaded). "startAuthorisationWithPsuIdentification": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU identification data. "startAuthorisationWithPsuAuthentication": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU authentication data. "startAuthorisationWithEncrypted PsuAuthentication": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the encrypted PSU authentication data. "startAuthorisationWithAuthentication MethodSelection": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while selecting the authentication method. This link is contained under exactly the same conditions as the data element "scaMethods" "startAuthorisationWithTransactionAuthorisation" :

Attribute	Type	Condition	Description
			The link to the authorisation end-point, where the authorisation sub-resource has to be generated while authorising the transaction e.g. by uploading an OTP received by SMS. "self": The link to the payment initiation resource created by this request. This link can be used to retrieve the resource data.
psuMessage	Max500Text	Optional	Text to be displayed to the PSU
tppMessages	Array of TPP Message Information	Optional	Messages to the TPP on operational issues.

Example

Request

```
POST https://api.testbank.com/v1/signing-baskets
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

{
   "paymentIds": ["123qwert456789", "12345qwert7899"]
}
```

Response (always with explicit authorisation start)

```
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/v1/signing-baskets/1234-
basket-567
```

```
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "basketId": "1234-basket-567",
  "_links": {
        "self": {"href": "/v1/signing-baskets/1234-basket-567"},
        "status": {"href": "/v1/signing-baskets/1234-basket-567/status"},
        "startAuthorisation": {"href": "/v1/signing-baskets/1234-basket-
567/authorisations"}
  }
}
```

8.2 Get Signing Basket Request

Call

```
GET /v1/signing-baskets/{basketId}
```

Returns the content of a signing basket object.

Path Parameters

Attribute	Type	Description
basketId	String	ID of the corresponding signing basket object.

Query Parameters

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	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

Attribute	Type	Condition	Description
			service in the same session, if no such OAuth2 SCA approach was chosen in the current signing basket transaction.

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
payments	array of paymentId	Optional	payment initiations which shall be authorised through this signing basket.
consents	array of consentId	Optional	consent objects which shall be authorised through this signing basket.
transactionStatus	Transaction Status	Mandatory	Only the not explicitly payment related codes like RCVD, PATC, ACTC, RJCT are used. For a list of all transactionStatus codes permitted for signing baskets, cp. Section 8.3.
_links	Links	Optional	The ASPSP might integrate hyperlinks to indicate next (authorisation) steps to be taken.

Example

Request

```
GET https://api.testbank.com/v1/signing-baskets/1234-basket-567
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:46 GMT
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json

{
"payments": ["1234pay567","1234pay568","1234pay888"],
"transactionStatus": "ACTC"
}
```

8.3 Get Signing Basket Status Request

Call

```
GET /v1/signing-baskets/{basketId}/status
```

Returns the status of a signing basket object.

Path Parameters

Attribute	Type	Description
basketId	String	ID of the corresponding signing basket object.

Query Parameters

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	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an

Attribute	Type	Condition	Description
			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 signing basket transaction.

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
transactionStatus	Transaction Status	Mandatory	Only the codes RCVD, PATC, ACTC, CANC and RJCT are supported for signing baskets.

Example

Request

```
GET https://api.testbank.com/v1/signing-baskets/1234-basket-567/status
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:49 GMT
```

Response

```
HTTP/1.x 200 Ok
```

```
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:51 GMT
Content-Type:           application/json

{
"transactionStatus": "ACTC"
}
```

8.4 Multi-level SCA for Signing Baskets

The Establish Signing Basket Request defined above is independent from the need of one or multilevel SCA processing, i.e. independent from the number of authorisations needed for the execution of all transactions contained in the basket. In contrast, the Establish Signing Basket Response defined above in this section are specific to the processing of one SCA. processing. In the following the background is explained on diverging requirements on the Establish Signing Basket Response message.

If any data is needed for starting the next action, like selecting an SCA method, this action is not supported through a hyperlink in the response, since all starts of the multiple authorisations are fully equal. In these cases, first an authorisation sub-resource has to be generated following the "startAuthorisation" link.

Response Body in case of Multi-Level SCA needed

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	Mandatory	The non payment related values defined in Section 14.24 might be used like RCVD, ACTC, PATC, CANC or RJCT
basketId	String	Mandatory	resource identification of the generated signing basket resource.
_links	Links	Mandatory	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP. Type of links admitted in this response, (further links might be added for ASPSP defined extensions):

Attribute	Type	Condition	Description
			"startAuthorisation": In case, where an explicit start of the transaction authorisation is needed, but no more data needs to be updated (no authentication method to be selected, no PSU identification nor PSU authentication data to be uploaded). "startAuthorisationWithPsuIdentification": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU identification data. "startAuthorisationWithPsuAuthentication": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU authentication data. "startAuthorisationWithEncryptedPsuAuthentication": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the encrypted PSU authentication data. "self": The link to the payment initiation resource created by this request. This link can be used to retrieve the resource data.
psuMessage	Max500Text	Optional	Text to be displayed to the PSU
tppMessages	Array of TPP Message Information	Optional	Messages to the TPP on operational issues.

8.5 Cancellation of Signing Baskets

A cancellation of a Signing Basket is only permitted where no (partial) authorisation has been applied for the Signing Basket.

Call

```
DELETE /v1/signing-baskets/{basketId}
```

Deletes a created signing basket if it is not yet (partially) authorised.

Path Parameters

Attribute	Type	Description
basketId	String	Contains the resource-ID of the signing basket to be deleted.

Query Parameters

No specific 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.
Authorization	String	Conditional	Is contained only, if an OAuth2 based SCA has been used in a pre-step.

Request Body

No Request Body.

Response Code

The HTTP response code is 204 in case of successful deletion.

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

No Response Body

Example

Request

```
DELETE https://api.testbank.com/v1/signing-baskets/qwer3456tzui9876
X-Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date                    Sun, 13 Aug 2017 17:05:37 GMT
```

Response

```
HTTP/1.x 204 No Content
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 13 Aug 2017 17:05:38 GMT
```

9 Sessions: Combination of AIS and PIS Services

The implementation of sessions in the sense of [XS2A-OR], i.e. the combination of AIS and PIS services is an optional feature of this interface. The ASPSP will inform about the support by its PSD2 documentation.

This feature might be relevant where account information services are needed within a payment initiation, especially for batch booking banks. In this case, a consent to access the corresponding account information is needed, cp. Section 6.3. The corresponding GET method to read the account data is using there the header parameter "Consent-ID". The TPP then can use this Consent-ID parameter also in the POST method when applying the Payment Initiation Request, cp. Section 5.3. A pre-requisite to use the "Consent-ID" in the subsequent Payment Initiation Request is that the flag "combinedServiceIndicator" in the Account Information Consent Request was set, cp. Section 6.3.1.

The usage of the "Consent-ID" in the subsequent Payment Initiation Request will then yield to not again ask for a first authentication factor, so the ASPSP will not again provide the PSU authentication link. In a case of SCA exemption for the corresponding payment, this can yield to a situation where no further PSU authentication is needed – the payment will then be executed without further confirmation.

In a context, where the consent management for account access is fully provided by the OAuth2 model, the corresponding access tokens will support this feature analogously.

10 Confirmation of Funds Service

10.1 Overview Confirmation of Funds Service

The following table defines the technical description of the abstract data model as defined [XS2A-OR] 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 [XS2A-OR] 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.
- The "Usage" column gives an overview on the usage of data elements in the different services and API Calls. Within [XS2A-OR], 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 for body related data elements only apply in case of HTTP response code 2xx.** In case of HTTP response code 4xx or 5xx requirements as defined in Section 4.13 apply.

The following usage of abbreviations in the Location and Usage columns is defined, cp. also [XS2A-OR] 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	
TPP Certificate Data	TPP-Signature-Certificate		x			c	
Further signature related data	Digest		x			c	
TPP Electronic Signature	Signature		x			c	
TPP Message Information	tppMessages			x			o
Card Number	cardNumber			x		o	
Account Number	account			x		m	
Name Payee	payee			x		o	
Transaction Amount	instructedAmount			x		m	

10.2 Confirmation of Funds Request

Call

```
POST /v1/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 as defined in [XS2A-COFC]. Otherwise not used.
Digest	cp. Section 12	Conditional	Is contained if and only if the "Signature" element is contained in the header of the request.
Signature	cp Section 12	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
TPP-Signature-Certificate	String	Conditional	The certificate used for signing the request, In base64 encoding.

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/v1/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

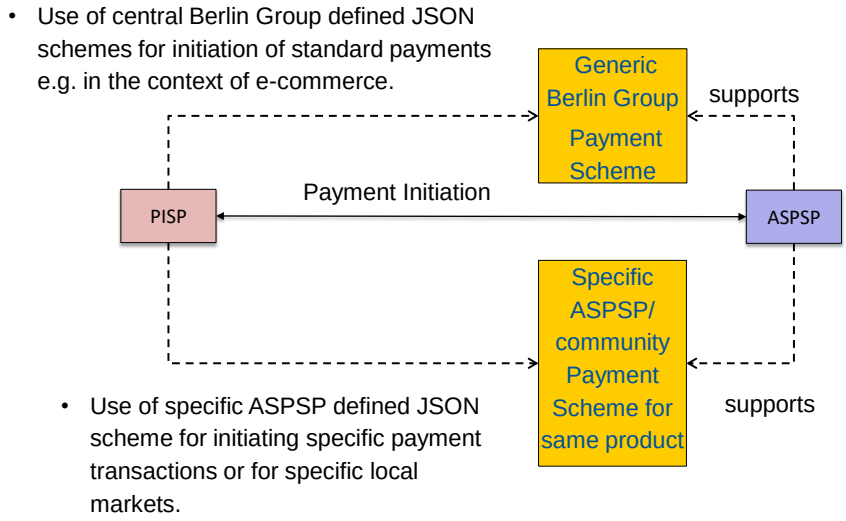
```
{"fundsAvailable": true}
```

11 Core Payment Structures

For core payment products in the European market, this document is defining JSON structures, which will be supported by all ASPSPs

- offering the corresponding payment products to their customers and
- providing JSON based payment endpoints, cp Sections 5.3.1 and 5.3.3.1.

At the same time, the ASPSP may offer in addition more extensive JSON structures for the same payment products since they might offer these extensions also in their online banking system.



11.1 Single Payments

The following table first gives an overview on the generic Berlin Group defined JSON structures of standard SEPA payment products for single payments.

Data Element	Type	SCT EU Core	SCT INST EU Core	Target2 Paym. Core	Cross Border CT Core
endToEnd Identification	Max35Text	optional	optional	optional	n.a.
instructionIdentification	Max35Text	n.a.	n.a.	n.a.	n.a.
debtorName	Max70Text	n.a.	n.a.	n.a.	n.a.
debtorAccount (incl. type)	Account Reference	mandatory[9]	mandatory[9]	mandatory[9]	mandatory[9]
debtorId	Max35Text	n.a.	n.a.	n.a.	n.a.
ultimateDebtor	Max70Text	n.a.	n.a.	n.a.	n.a.
instructedAmount (inc. Curr.)	Amount	mandatory	mandatory	mandatory	mandatory
currencyOfTransfer[10]	Currency Code	n.a.	n.a.	n.a.	n.a.
exchangeRateInformation	Payment Exchange Rate	n.a.	n.a.	n.a.	n.a.
creditorAccount	Account Reference	mandatory	mandatory	mandatory	mandatory
creditorAgent	BICFI	optional	optional	optional	conditional[11]
creditorAgentName	Max140Text	n.a.	n.a.	n.a.	n.a.
creditorName	Max70Text	mandatory	mandatory	mandatory	mandatory
creditorId	Max35Text	n.a.	n.a.	n.a.	n.a.
creditorAddress	Address	optional	optional	optional	conditional[12]
creditorNameAnd Address	Max140Text	n.a.	n.a.	n.a.	n.a.
ultimateCreditor	Max70Text	n.a.	n.a.	n.a.	n.a.
purposeCode	Purpose Code	n.a.	n.a.	n.a.	n.a.

[9] ASPSPs might change the condition on the debtor account for SEPA payments to optional as one way to fulfil the requirement according to item 36 of the EBA Opinion of June 2020.

[10] This is a data element to indicate a diverging interbank transaction currency.

[11] This field might be mandated by ASPSPs generally or depending of the creditor's address' country.

[12] This field might be mandated by ASPSPs generally or depending of the creditor's address' country.

Data Element	Type	SCT EU Core	SCT INST EU Core	Target2 Paym. Core	Cross Border CT Core
chargeBearer	Charge Bearer	n.a.	n.a.	optional	conditional[13]
serviceLevel	Service Level Code	n.a.	n.a.	n.a.	n.a.
remittance Information Unstructured	Max140Text	optional	optional	optional	optional
remittance Information Unstructured Array	Array of Max140Text	n.a.	n.a.	n.a.	n.a.
remittance Information Structured	Remittance	n.a.	n.a.	n.a.	n.a.
remittance Information Structured Array	Array of Remittance	n.a.	n.a.	n.a.	n.a.
requestedExecution Date	ISODate	n.a.	n.a.	n.a.	n.a.
requestedExecution Time	ISODateTime	n.a.	n.a.	n.a.	n.a.

The data elements marked with "n.a." are not used in the addressed core services, shared by all ASPSP offering these product, but they can be used in ASPSP or community wide extensions. Extensions of these tables are permitted by this specification

- if they are less restrictive (e.g. set the debtor account to optional) or
- if they open up for more data elements (e.g. open up the structured remittance information, or ultimate data fields.)

  Remark: The debtor account is a mandatory field for a single payment. If bulk payments are use, the debtor account is only used in the introductory part of the bulk structure, cp. Section 11.3.

  Remark: The ASPSP may reject a payment initiation request where additional data elements are used which are not specified.

[13] This field might be mandated by ASPSPs generally or depending of default usage definitions of the ASPSP.

Remark: An example for the above introduced extensions for the SEPA payments are the extensions for the Austrian market as described in [XS2A-DP].

11.2 Future Dated Payments

One example of an extension of the above defined JSON structure is the requested execution date e.g. for SEPA Credit Transfers. This field is n.a. since not all banks or banking communities might support this as a PSD2 core service.

The ASPSP will indicate the acceptance of future dated payments by issuing an ASPSP specific or community specific JSON scheme, where the attribute "requestedExecutionDate" is an optional field.

11.3 Bulk Payments

This specification offers the bulk payment function in JSON encoding as optional endpoint. The format of the bulk payment is an array of single payments, as offered by the ASPSP, preceded by generic payment information applicable to all individual payments contained.

Data Element	Type	Condition	Description
batchBookingPreferred	Boolean	optional	If this element equals true, the PSU prefers only one booking entry. If this element equals false, the PSU prefers individual booking of all contained individual transactions. The ASPSP will follow this preference according to contracts agreed on with the PSU.
debtorAccount (incl. type)	Account Reference	mandatory	
paymentInformationId	Max35Text	n.a.	Unique identification as assigned by the sending party to unambiguously identify this bulk payment. This attribute may be used by ASPSPs or communities as an optional field. **Remark for Future**: This attribute might be made mandatory in the next version of the specification.
requestedExecutionDate	ISODate	optional	If contained, the payments contained in this bulk will be executed at the addressed date. This field may not be used together with the field requestedExecutionTime.
requestedExecutionTime	ISODateTime	optional	If contained, the payments contained in this bulk will be executed at the addressed Date/Time. This field may not be used together with the field requestedExecutionDate.
payments	Bulk Entry	mandatory	The Bulk Entry Type is a type which follows the JSON formats for the supported products for single payments, see Section 11.1, excluding the data elements

Data Element	Type	Condition	Description
			• debtorAccount, • requestedExecutionDate, • requestedExecutionTime. These three data elements may not be contained in any bulk entry.

Example

```
{"batchBookingPreferred": true,
 "debtorAccount": {"iban": "DE40100100103307118608"},
 "requestedExecutionDate": "2018-08-01",
 "payments":
[{JSON based payment initiation 1}, {JSON based payment initiation 2}]}
```

12 Signatures

When an ASPSP requires the TPP to send a digital signature as defined in [signHTTP], chapter 4 in his HTTP-Requests, the signature must obey the following requirements according or additional to [signHTTP], chapter 4.

12.1 "Digest" Header mandatory

When a TPP includes a signature as defined in [signHTTP], chapter 4, he also must include a "Digest" header as defined in [RFC3230]. The "Digest" Header contains a Hash of the message body, if the message does not contain a body, the "Digest" header must contain the hash of an empty bytelist. The only hash algorithms that may be used to calculate the Digest within the context of this specification are SHA-256 and SHA-512 as defined in [RFC5843].

Remark: In case of a multipart message the same method is used to calculate the digest. I.e. a hash of the (whole) message body is calculated including all parts of the multipart message as well as the separators.

12.2 Requirements on the "Signature" Header

As defined in [signHTTP], chapter 4, a "Signature" header must be present. The structure of a "Signature" header is defined in [signHTTP], chapter 4.1, the following table lists the requirements on the "Signature" header from [signHTTP] and additional requirements specific to the PSD2-Interface.

Elements of the "Signature" Header				
Element	**Type**	**Condition**	**Requirement [signHTTP]**	**Additional Requirement**
keyId	String	Mandatory	The keyId field is a string that the server can use to look up the component they need to validate the signature.	Serial Number of the TPP's certificate included in the "TPP-Signature-Certificate" header of this request. It shall be formatted as follows: keyId="SN=XXX,CA=YYYYYYYYYYYYYYYY" where “XXX" is the serial number of the certificate in hexadecimal coding given in the TPP-Signature-Certificate-Header and "YYYYYYYYYYYYYYYY" is the full Distinguished Name of the Certification Authority

Elements of the "Signature" Header				
Element	**Type**	**Condition**	**Requirement [signHTTP]**	**Additional Requirement**
				having produced this certificate.
Algorithm	String	Mandatory (Optional in **[signHTTP]**)	The "Algorithm " parameter is used to specify the digital signature algorithm to use when generating the signature. Valid values for this parameter can be found in the Signature Algorithms registry located at http://www.iana.org/assignments/signature-algorithms and MUST NOT be marked "deprecated". It is preferred that the algorithm used by an implementation be derived from the key metadata identified by the 'keyId' rather than from this field. […]The 'algorithm' parameter [..] **will most likely be deprecated in the future**.	Mandatory The algorithm must identify the same algorithm for the signature as described for the TPP's public key (Subject Public Key Info) in the certificate (Element "TPP-Signature-Certificate") of this Request. It must identify SHA-256 or SHA-512 as Hash algorithm.

Elements of the "Signature" Header				
Element	**Type**	**Condition**	**Requirement [signHTTP]**	**Additional Requirement**
Headers	String	Mandatory (Optional in **[signHTTP]**)	The "Headers" parameter is used to specify the list of HTTP headers included when generating the signature for the message. If specified, it should be a lowercased, quoted list of HTTP header fields, separated by a single space character. If not specified, implementations MUST operate as if the field were specified with a single value, the `Date` header, in the list of HTTP headers. Note that the list order is important, and MUST be specified in the order the HTTP header field-value pairs are concatenated together during signing.	Mandatory. Must include • "digest", • "x-request-id", Must conditionally include • "psu-id", if and only if "PSU-ID" is included as a header of the HTTP-Request. • "psu-corporate-id", if and only if "PSU-Corporate-ID" is included as a header of the HTTP-Request. • "tpp-redirect-uri", if and only if "TPP-Redirect-URI" is included as a header of the HTTP-Request. **No other entries may be included.** **Remark:** It is intended to introduce a new http header in a coming version. This new header shall indicate the creation date of a request on the side of the TPP. This new header and will also have to be included in this "Headers" element.

Elements of the "Signature" Header				
Element	**Type**	**Condition**	**Requirement [signHTTP]**	**Additional Requirement**
Signature	String	Mandatory	The "signature" parameter is a base 64 encoded digital signature, as described in RFC 4648 [RFC4648], Section 4. The client uses the `algorithm` and `headers` signature parameters to form a canonicalised `signing string`. This `signing string` is then signed with the key associated with `keyId` and the algorithm corresponding to `algorithm`. The `signature` parameter is then set to the base 64 encoding of the signature.	[No additional Requirements]

Example

Assume a TPP needs to include a signature in the following Request

```
POST https://api.testbank.com/v1/payments/sepa-credit-transfers
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-IP-Address:         192.168.8.78
PSU-ID:                 PSU-1234
PSU-User-Agent:         Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
tpp-redirect-uri: https%3A%2F%2FshortURI_Cchallenge_Mmethod="S256"
Date:                   Sun, 06 Aug 2017 15:02:37 GMT

{
   "instructedAmount": {"currency": "EUR", "amount": "123"},
   "debtorAccount": {"iban": "DE23100100101234567 89"},
   "creditor": {"name": "Merchant123"},
   "creditorAccount": {"iban": "DE23100120020123456789"},
   "remittanceInformationUnstructured": "Ref Number Merchant"
}
```

So the body would encode to the following String in Base64:

eyAgICANCiAgICJpbnN0cnVjdGVkQW1vdW50IjogeyJjdXJyZW5jeSI6ICJFVVIiLCAiYW1vd
W50IjogIjEyMyJ9LA0KICAgImRlYnRvckFjY291bnQiOiB7ImliYW4iOiAiREUyMzEwMDEwMD
EwMTIzNDU2Nzg5In0sDQogICAiY3JlZGl0b3IiOiB7Im5hbWUiOiAiTWVyY2hhbnQxMjMifSw
NCiAgICJjcmVkaXRvckFjY291bnQiOiB7ImliYW4iOiAiREUyMzEwMDEyMDAyMDEyMzQ1Nj
c4OSJ9LA0KICAgInJlbWl0dGFuY2VJbmZvcm1hdGlvblVuc3RydWN0dXJlZCI6ICJSZWYgT
nVtYmVyIE1lcmNoYW50Ig0KfQ==

and SHA-256 of the request body is

`iXhCYo105ae/y5v/UJkQWuBe1I+mdKG0JxwU35vwsgo=` in Base64 (`897842628D74E5A7BFCB9BFF5099105AE05ED48FA674A1B4271C14DF9BF0B20A` in hexadecimal representation).

So using signature algorithm rsa-sha256 the signed request of the TPP will be

```
POST https://api.testbank.com/v1/payments/sepa-credit-transfers
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-IP-Address:         192.168.8.78
PSU-ID:                 PSU-1234
PSU-User-Agent:          Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
tpp-redirect-uri: https%3A%2F%2FshortURI_Cchallenge_Mmethod="S256"
Date:                   Sun, 06 Aug 2017 15:02:37 GMT
Digest:                 SHA-
256=ZuYiOtZkVxhjWmwTO5lOpsPevUNMezvk6dfb6fVhebM=
Signature:              keyId="SN=9FA1,CA=CN=D-TRUST%20CA%202-1%202015,O=D-
Trust%20GmbH,C=DE",algorithm="rsa-sha256",
   headers="digest x-request-id psu-id tpp-redirect-uri",
   signature="Base64(RSA-SHA256(signing string))"
TPP-Signature-Certificate: TPP's_eIDAS_Certificate

{
   "instructedAmount": {"currency": "EUR", "amount": "123"},
   "debtorAccount": { "iban": "DE23100100101234567890"},
   "creditor": { "name": "Merchant123"},
   "creditorAccount": {"iban": "DE23100120020123456789"},
   "remittanceInformationUnstructured": "Ref Number Merchant"
}
```

Where *signing string* is

```
digest: SHA-256=iXhCYo105ae/y5v/UJkQWuBe1I+mdKG0JxwU35vwsgo=
x-request-id: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
psu-id: PSU-1234
tpp-redirect-uri: https%3A%2F%2FshortURI_Cchallenge_Mmethod="S256"
```

NOTE: The header fields to be signed are denoted in small letters to clarify that the digest will use small letters for normalisation.

13 Requirements on the OAuth2 Protocol

The OAuth2 protocol as used optionally for this API is defined in [RFC6749]. In this section, additional requirements on the protocol are defined.

The requirements on the data exchange between the TPP and the OAuth Server of the ASPSP are identical to the data exchange requirements between TPP and the XS2A Interface, cp. Section 3.

The response type "code" and the grant types "authorization_code" and "refresh_token" are recommended by this specification. It is further strongly recommended to TPPs and ASPSPs to follow the security best practices defined in [OA-SecTop].

The ASPSP is required to provide TPPs with configuration data conforming to the "OAuth 2.0 Authorisation Server Metadata" specification.

13.1 Authorisation Request

For the "authorisation request" of the TPP to the authorisation endpoint the following parameters are defined:

Query Parameters

Attribute	Condition	Description
response_type	Mandatory	"code" is recommended as response type.
client_id	Mandatory	organizationIdentifier as provided in the eIDAS certificate. The organizationIdentifier attribute shall contain information using the following structure in the presented order: - "PSD" as 3 character legal person identity type reference; - 2 character ISO 3166 country code representing the NCA country; - hyphen-minus "-" and - 2-8 character NCA identifier (A-Z uppercase only, no separator) - hyphen-minus "-" and - PSP identifier (authorization number as specified by NCA).
scope	Mandatory	PIS: The scope is the reference to the payment resource in the form "PIS:<paymentId>".

Attribute	Condition	Description
		AIS: The scope is the reference to the consent resource for account access in the form "AIS:<consentId>" PIIS: The scope is the reference to the consent resource for granting consent to confirmation of funds in the form "PIIS:<consentId>". **Note:** The resource ids chosen by the ASPSP need to be unigue to avoid resource conflicts during the SCA process.
state	Mandatory	A dynamical value set by the TPP and used to prevent XSRF attacks.
redirect_uri	Mandatory	the URI of the TPP where the OAuth2 server is redirecting the PSU's user agent after the authorization.
code_challenge	Mandatory	PKCE challenge according to cryptographic RFC 7636 (https://tools.ietf.org/html/rfc7636) used to prevent code injection attacks.
code_challenge_method	Optional	Code verifier transformation method, is "S256" or "plain". "S265" is recommended by this specification.

Example

```
GET /authorise?response_type=code&client_id="PSDES-BDE-3DFD21" &
scope=ais%3A1234-wertiq-983+offline_access&
state= S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw&
redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb&
code_challenge_method="S256"
code_challenge=5c305578f8f19b2dcdb6c3c955c0aa709782590b4642eb890b97e43917cd
0f36 HTTP/1.1
    Host: api.testbank.com
```

13.2 Authorisation Response

The Authorisation Response of the ASPSP to the TPP will deliver the following data:

http Response Code

302

Query Parameters

Attribute	Condition	Description
Location:	Mandatory	redirect URI of the TPP
code	Mandatory	Authorisation code
state	Mandatory	Same value as for the request.

Example

```
HTTP/1.1 302 Found

Location: https://client.example.com/cb
?code=SplxlOBeZQQYbYS6WxSbIA
&state=S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw
```

13.3 Token Request

The TPP sends a POST request to the token endpoint in order to exchange the authorisation code provided in the authorisation response for an access token and, optionally, a refresh token. The following parameters are used:

Request Parameters

Attribute	Condition	Description
grant_type	Mandatory	"authorization_code" is recommended as response type.
client_id	Mandatory	cp. Definition in Section 13.1
code	Mandatory	Authorisation code from the authorisation response
redirect_uri	Mandatory	the exact uri of the TPP where the OAuth2 server redirected the user agent to for this particular transaction
code_verifier	Mandatory	PKCE verifier according to cryptographic RFC 7636 (https://tools.ietf.org/html/rfc7636) used to prevent code injection attacks.

Example

```
POST /token HTTP/1.1
Host: https://api.testbank.com
Content-Type: application/x-www-form-urlencoded
client_id="PSDES-BDE-3DFD21"
&grant_type=authorisation_code
&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri= https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
&code_verifier=7814hj4hjai87qqhjz9hahdeu9qu771367647864676787878
```

The TPP is authenticated during this request by utilising "Oauth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens" in conjunction with the TPP's eIDAS certificate.

13.4 Token Response

The ASPSPS responds with the following parameters:

Response Parameters

Attribute	Condition	Description
access_token	Mandatory	Access Token bound to the scope as requested in the authorisation request and confirmed by the PSU.
token_type	Mandatory	Set to "Bearer"
expires_in	Optional	The lifetime of the access token in seconds
refresh_token	Optional	Refresh Token, which can be utilised to obtain a fresh access tokens in case the previous access token expired or was revoked. Especially useful in the context of AIS.
scope	Mandatory	the scope of the access token

Example

HTTP/1.1 200 OK

```
Content-Type: application/json
Cache-Control: no-store
```

```
Pragma: no-cache

{
 "access_token": "SlAV32hkKG",
 "token_type": "Bearer",
 "expires_in": 3600,
 "refresh_token": "tGzv3JokF0XG5Qx2TlKWIA",
 "scope": "exampleScope"
}
```

13.5 Refresh Token Grant Type

The ASPSP may issue refresh tokens at its discretion, e.g. if an AISP uses the standard scope value "offline_access" or if the recurringIndicator in is set to true.

13.6 API Requests

When using the OAuth SCA approach, subsequent API requests are being authorized using the respective OAuth Access Token. The access token is sent to the API using the “Authorization” Header and the “BEARER” authorization schema as defined in RFC 6750.

This is an example API request

```
GET /v1/payments/ sepa-credit-transfers/1234-wertiq-983/status HTTP/1.1
Host: https://api.testbank.com
Authorization: Bearer SlAV32hkKG
```

14 Complex Data Types and Code Lists

In the following constructed data types are defined as used within parameter sections throughout this document.

14.1 PSU Data

Attribute	Type	Condition	Description
password	String	Conditional	Contains a password in plaintext.
encrypted Password	String	Conditional	Is used when a password is encrypted on application level.
additional Password	String	Conditional	Contains an additional password in plaintext
additional Encrypted Password	String	Conditional	Is provided when the additional password is used and is encrypted on application level.

14.2 TPP Message Information

Attribute	Type	Condition	Description
category	String	Mandatory	Only "ERROR" or "WARNING" permitted
code	Message Code	Mandatory	
path	String	Conditional	
text	Max500Text	Optional	Additional explaining text.

14.3 Amount

Attribute	Type	Condition	Description
currency	Currency Code	Mandatory	ISO 4217 Alpha 3 currency code

Attribute	Type	Condition	Description
amount	String	Mandatory	The amount given with fractional digits, where fractions must be compliant to the currency definition. Up to 14 significant figures. Negative amounts are signed by minus. The decimal separator is a dot. **Example:** Valid representations for EUR with up to two decimals are: • 1056 • 5768.2 • -1.50 • 5877.78

14.4 Address

Attribute	Type	Condition	Description
streetName	Max70Text	Optional	
buildingNumber	String	Optional	
townName	String	Optional	
postCode	String	Optional	
country	Country Code	Mandatory	

14.5 Remittance

Attribute	Type	Condition	Description
reference	Max35Text	Mandatory	The actual reference.
referenceType	Max35Text	Optional	
referenceIssuer	Max35Text	Optional	

14.6 Links

The structure of Links is conform to [HAL].

Attribute	Type	Condition	Description
scaRedirect	href Type	Optional	A link to an ASPSP site where SCA is performed within the Redirect SCA approach.
scaOAuth	href Type	Optional	The link refers to a JSON document specifying the OAuth details of the ASPSP's authorisation server. JSON document follows the definition given in https://tools.ietf.org/html/draft-ietf-oauth-discovery.
confirmation	href Type	Optional	"confirmation": Might be added by the ASPSP if either the "scaRedirect" or "scaOAuth" hyperlink is returned in the same

Attribute	Type	Condition	Description
			response message. This hyperlink defines the URL to the resource which needs to be updated with • a confirmation code as retrieved after the plain redirect authentication process with the ASPSP authentication server or • an access token as retrieved by submitting an authorization code after the integrated OAuth based authentication process with the ASPSP authentication server.
startAuthorisation	href Type	Optional	A link to an endpoint, where the authorisation of a transaction or the authorisation of a transaction cancellation shall be started with a POST command. No specific data is needed for this process start.
startAuthorisationWithPsuIdentification	href Type	Optional	The link to an endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where PSU identification shall be uploaded with the corresponding call.
updatePsuIdentification	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by the PSU identification if not delivered yet.
startAuthorisationWithProprietaryData	hrefType	Optional	A link to the endpoint, where the authorisation of a transaction or of a transaction cancellation shall be started, and where proprietary data needs to be updated with this call. The TPP can find the scope of missing proprietary data in the ASPSP documentation. The usage of this hyperlink is not further specified in the specification but is used analogously to e.g. the startAuthorisation withPsuIdentification hyperlink.

Attribute	Type	Condition	Description
updateProprietaryData	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by the proprietary data.
startAuthorisationWithPsuAuthentication	href Type	Optional	The link to an endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where PSU authentication data shall be uploaded with the corresponding call.
updatePsuAuthentication	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by a PSU password and eventually the PSU identification if not delivered yet.
updateAdditionalPsuAuthentication	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by an additional PSU password.
startAuthorisationWithEncryptedPsuAuthentication	href Type	Optional	The link to an endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where encrypted PSU authentication data shall be uploaded with the corresponding call.
updateEncryptedPsuAuthentication	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by an encrypted PSU password and eventually the PSU identification if not delivered yet.
updateAdditionalEncryptedPsuAuthentication	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by an additional encrypted PSU password.
startAuthorisationWithAuthenticationMethodSelection	href Type	Optional	This is a link to and endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where the selected SCA method shall be uploaded with the corresponding call.
selectAuthenticationMethod	href Type	Optional	This is a link to a resource, where the TPP can select the applicable second factor

Attribute	Type	Condition	Description
			authentication methods for the PSU, if there were several available authentication methods.
startAuthorisationWithTransactionAuthorisation	href Type	Optional	A link to an endpoint, where an authorisation of a transaction or a cancellation can be started, and where the response data for the challenge is uploaded in the same call for the transaction authorisation or transaction cancellation at the same time in the Embedded SCA Approach.
authoriseTransaction	href Type	Optional	The link to the payment initiation or consent resource, where the "Transaction Authorisation"Request" is sent to. This is the link to the resource which will authorise the payment or the consent by checking the SCA authentication data within the Embedded SCA approach.
self	href Type	Optional	The link to the payment initiation resource created by the request itself. This link can be used later to retrieve the transaction status of the payment initiation.
status	href Type	Optional	A link to retrieve the status of the transaction resource.
scaStatus	href Type	Optional	A link to retrieve the status of the authorisation or cancellation-authorisation sub-resource.
account	href Type	Optional	A link to the resource providing the details of one account
balances	href Type	Optional	A link to the resource providing the balance of a dedicated account.
transactions	href Type	Optional	A link to the resource providing the transaction history of a dedicated account.
cardAccount	href Type	Optional	A link to the resource providing the details of one card account.

Attribute	Type	Condition	Description
cardTransactions	href Type	Optional	A link to the resource providing the transaction history of a dedicated card account.
transactionDetails	href Type	Optional	A link to the resource providing details of a dedicated transaction.
first	href Type	Optional	Navigation link for paginated account reports.
next	href Type	Optional	Navigation link for paginated account reports.
previous	href Type	Optional	Navigation link for paginated account reports.
last	href Type	Optional	Navigation link for paginated account reports.
download	href Type	Optional	Download link for huge AIS data packages.

14.7 href Type

Attribute	Type	Condition	Description
href	String	Mandatory	

14.8 Authentication Object

Attribute	Type	Condition	Description
authenticationType	Authentication Type	Mandatory	Type of the authentication method.
authenticationVersion	String	Conditional	Depending on the "authenticationType". This version can be used by differentiating authentication tools used within performing OTP generation in the same authentication type. This version can be referred to in

Attribute	Type	Condition	Description
			the ASPSP's documentation.
authenticationMethodId	Max35Text	Mandatory	An identification provided by the ASPSP for the later identification of the authentication method selection.
name	String	Mandatory	This is the name of the authentication method defined by the PSU in the Online Banking frontend of the ASPSP. Alternatively this could be a description provided by the ASPSP like "SMS OTP on phone +49160 xxxxx 28". This name shall be used by the TPP when presenting a list of authentication methods to the PSU, if available.
explanation	String	Optional	detailed information about the SCA method for the PSU

14.9 Authentication Type

More authentication types might be added during implementation projects and documented in the ASPSP documentation.

Name	Description
SMS_OTP	An SCA method, where an OTP linked to the transaction to be authorised is sent to the PSU through a SMS channel.
CHIP_OTP	An SCA method, where an OTP is generated by a chip card, e.g. an TOP derived from an EMV cryptogram. To contact the card, the PSU normally needs a (handheld) device. With this device, the PSU either reads the challenging data through a visual interface like flickering or the PSU types in the challenge through the device key pad. The

Name	Description
	device then derives an OTP from the challenge data and displays the OTP to the PSU.
PHOTO_OTP	An SCA method, where the challenge is a QR code or similar encoded visual data which can be read in by a consumer device or specific mobile app. The device resp. the specific app than derives an OTP from the visual challenge data and displays the OTP to the PSU.
PUSH_OTP	An OTP is pushed to a dedicated authentication APP and displayed to the PSU.
SMTP_OTP	An OTP is sent via email to the PSU.

14.10 Challenge

Attribute	Type	Condition	Description
image	String	Optional	PNG data (max. 512 kilobyte) to be displayed to the PSU, Base64 encoding, cp. [RFC4648]. This attribute is used only, when PHOTO_OTP or CHIP_OTP is the selected SCA method.
data	Array of Strings	Optional	A collection of challenge data
imageLink	String	Optional	A link where the ASPSP will provides the challenge image for the TPP.
otpMaxLength	Integer	Optional	The maximal length for the OTP to be typed in by the PSU.
otpFormat	String	Optional	The format type of the OTP to be typed in. The admitted values are "characters" or "integer".
additional Information	String	Optional	Additional explanation for the PSU to explain e.g. fallback mechanism for the chosen SCA method. The TPP is obliged to show this to the PSU.

14.11 Message Code

The permitted message error codes and related HTTP response codes are listed below.

14.11.1 Service Unspecific HTTP Error Codes

Message Code	HTTP Response Code	Description
CERTIFICATE_INVALID	401	The contents of the signature/corporate seal certificate are not matching PSD2 general PSD2 or attribute requirements.
ROLE_INVALID	401	The TPP does not have the correct PSD2 role to access this service.
CERTIFICATE_EXPIRED	401	Signature/corporate seal certificate is expired.
CERTIFICATE_BLOCKED	401	Signature/corporate seal certificate has been blocked by the ASPSP or the related NCA.
CERTIFICATE_REVOKED	401	Signature/corporate seal certificate has been revoked by QSTP.
CERTIFICATE_MISSING	401	Signature/corporate seal certificate was not available in the request but is mandated for the corresponding.
SIGNATURE_INVALID	401	Application layer eIDAS Signature for TPP authentication is not correct.
SIGNATURE_MISSING	401	Application layer eIDAS Signature for TPP authentication is mandated by the ASPSP but is missing.
ROLE_INVALID	401	The TPP does not have the correct PSD2 role to access this service
FORMAT_ERROR	400	Format of certain request fields are not matching the XS2A requirements. An explicit path to the corresponding field might be added in the return message. This applies to headers and body entries. It also applies in cases where these entries are

Message Code	HTTP Response Code	Description
		referring to erroneous or not existing data instances, e.g. a malformed IBAN.
PARAMETER_NOT_CONSISTENT	400	Parameters submitted by TPP are not consistent. This applies only for query parameters.
PARAMETER_NOT_SUPPORTED	400	The parameter is not supported by the API provider. This code should only be used for parameters that are described as "optional if supported by API provider."
PSU_CREDENTIALS_INVALID	401	The PSU-ID cannot be matched by the addressed ASPSP or is blocked, or a password resp. OTP was not correct. Additional information might be added.
SERVICE_INVALID	400 (if payload) 405 (if HTTP method)	The addressed service is not valid for the addressed resources or the submitted data.
SERVICE_BLOCKED	403	This service is not reachable for the addressed PSU due to a channel independent blocking by the ASPSP. Additional information might be given by the ASPSP.
CORPORATE_ID_INVALID	401	The PSU-Corporate-ID cannot be matched by the addressed ASPSP.
CONSENT_UNKNOWN	403 (if path) 400 (if header)	The Consent-ID cannot be matched by the ASPSP relative to the TPP.
CONSENT_INVALID	401	The consent was created by this TPP but is not valid for the addressed service/resource.
CONSENT_EXPIRED	401	The consent was created by this TPP but has expired and needs to be renewed.

Message Code	HTTP Response Code	Description
TOKEN_UNKNOWN	401	The OAuth2 token cannot be matched by the ASPSP relative to the TPP.
TOKEN_INVALID	401	The OAuth2 token is associated to the TPP but is not valid for the addressed service/resource.
TOKEN_EXPIRED	401	The OAuth2 token is associated to the TPP but has expired and needs to be renewed.
RESOURCE_UNKNOWN	404 (if account-id in path) 403 (if other resource in path) 400 (if payload)	The addressed resource is unknown relative to the TPP. An example for a payload reference is creating a signing basket with an unknown resource identficiation.
RESOURCE_EXPIRED	403 (if path) 400 (if payload)	The addressed resource is associated with the TPP but has expired, not addressable anymore.
RESOURCE_BLOCKED	400	The addressed resource is not addressable by this request, since it is blocked e.g. by a grouping in a signing basket.
TIMESTAMP_INVALID	400	Timestamp not in accepted time period.
PERIOD_INVALID	400	Requested time period out of bound.
SCA_METHOD_UNKNOWN	400	Addressed SCA method in the Authentication Method Select Request is unknown or cannot be matched by the ASPSP with the PSU.
SCA_INVALID	400	Method Application on authorisation resource (e.g. Confirmation Request) blocked since SCA status of the resource equals "failed".

Message Code	HTTP Response Code	Description
STATUS_INVALID	409	The addressed resource does not allow addtitional authorisation.

14.11.2 PIS Specific HTTP Error Codes

Message Code	HTTP Response Code	Description
PRODUCT_INVALID	403	The addressed payment product is not available for the PSU .
PRODUCT_UNKNOWN	404	The addressed payment product is not supported by the ASPSP.
PAYMENT_FAILED	400	The payment initiation POST request failed during the initial process. Additional information may be provided by the ASPSP.
REQUIRED_KID_MISSING	401	The payment initiation has failed due to a missing KID. This is a specific message code for the Norwegian market, where ASPSP can require the payer to transmit the KID.
EXECUTION_DATE_INVALID	400	The requested execution date is not a valid execution date for the ASPSP.
CANCELLATION_INVALID	405	The addressed payment is not cancellable e.g. due to cut off time passed or legal constraints.

14.11.3 AIS Specific HTTP Error Codes

Message Code	HTTP Response Code	Description
CONSENT_INVALID	401	The consent definition is not complete or invalid. In case of being not complete, the bank

Message Code	HTTP Response Code	Description
		is not supporting a completion of the consent towards the PSU. Additional information will be provided.
SESSIONS_NOT_SUPPORTED	400	The combined service flag may not be used with this ASPSP.
ACCESS_EXCEEDED	429	The access on the account has been exceeding the consented multiplicity without PSU involvement per day.
REQUESTED_FORMATS _INVALID	406	The requested formats in the Accept header entry are not matching the formats offered by the ASPSP.

14.11.4 PIIS Specific Error Codes

Message Code	HTTP Response Code	Description
CARD_INVALID	400	Addressed card number is unknown to the ASPSP or not associated to the PSU.
NO_PIIS_ACTIVATION	400	The PSU has not activated the addressed account for the usage of the PIIS associated with the TPP.

14.11.5 Signing Basket Specific Error Codes

Message Code	HTTP Response Code	Description
REFERENCE_MIX_INVALID	400	The used combination of referenced objects is not supported in the ASPSPs signing basket function.

Message Code	HTTP Response Code	Description
REFERENCE_STATUS_INVALID	409	At least one of the references is already fully authorised.

14.12 Error Information

This is a data element to support the declaration of additional errors in the context of [RFC7807].

Attribute	Type	Condition	Description
title	Max70Text	Optional	Short human readable description of error type. Could be in local language. To be provided by ASPSPs.
detail	Max500Text	Optional	Detailed human readable text specific to this instance of the error. XPath might be used to point to the issue generating the error in addition. **Remark for Future**: In future, a dedicated field might be introduced for the XPath.
code	Message Code	Mandatory	Message code to explain the nature of the underlying error.

14.13 Transaction Status

The transaction status is filled with codes of the ISO 20022 data table:

Code	Name	ISO 20022 Definition
ACCC	AcceptedSettlementCompleted	Settlement on the creditor's account has been completed.

Code	Name	ISO 20022 Definition
ACCP	AcceptedCustomerProfile	Preceding check of technical validation was successful. Customer profile check was also successful.
ACSC	AcceptedSettlementCompleted	Settlement on the debtor's account has been completed. **Usage:** this can be used by the first agent to report to the debtor that the transaction has been completed. **Warning:** this status is provided for transaction status reasons, not for financial information. It can only be used after bilateral agreement
ACSP	AcceptedSettlementInProcess	All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.
ACTC	AcceptedTechnicalValidation	Authentication and syntactical and semantical validation are successful
ACWC	AcceptedWithChange	Instruction is accepted but a change will be made, such as date or remittance not sent.
ACWP	AcceptedWithoutPosting	Payment instruction included in the credit transfer is accepted without being posted to the creditor customer's account.
RCVD	Received	Payment initiation has been received by the receiving agent.
PDNG	Pending	Payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be performed.
RJCT	Rejected	Payment initiation or individual transaction included in the payment initiation has been rejected.
CANC	Cancelled	Payment initiation has been cancelled before execution **Remark:** This code is accepted as new code by ISO20022.
ACFC	AcceptedFundsChecked	Pre-ceeding check of technical validation and customer profile was successful and an automatic funds check was positive . **Remark:** This code is accepted as new code by ISO20022.
PATC	PartiallyAcceptedTechnical Correct	The payment initiation needs multiple authentications, where some but not yet all have been performed. Syntactical and semantical validations are successful. **Remark**: This code is is accepted as new code by ISO20022.
PART	PartiallyAccepted	A number of transactions have been accepted, whereas another number of transactions have not yet achieved 'accepted' status. **Remark:** This code may be used only in case of bulk payments. It is only used in a situation where all mandated authorisations have been applied, but some payments have been rejected.

14.14 Consent Status

Code	Description
received	The consent data have been received and are technically correct. The data is not authorised yet.
rejected	The consent data have been rejected e.g. since no successful authorisation has taken place.
partiallyAuthorised	The consent is due to a multi-level authorisation, some but not all mandated authorisations have been performed yet.
valid	The consent is accepted and valid for GET account data calls and others as specified in the consent object.
revokedByPsu	The consent has been revoked by the PSU towards the ASPSP.
expired	The consent expired.
terminatedByTpp	The corresponding TPP has terminated the consent by applying the DELETE method to the consent resource.

The ASPSP might add further codes. These codes then shall be contained in the ASPSP's documentation of the XS2A interface.

14.15 SCA Status

The following codes are defined for this data type.

Remark for Future: A rework of the coding will follow, first Codes are given below:

Code	Description
received	An authorisation or cancellation-authorisation resource has been created successfully.
psuIdentified	The PSU related to the authorisation or cancellation-authorisation resource has been identified.
psuAuthenticated	The PSU related to the authorisation or cancellation-authorisation resource has been identified and authenticated e.g. by a password or by an access token.

Code	Description
scaMethodSelected	The PSU/TPP has selected the related SCA routine. If the SCA method is chosen implicitly since only one SCA method is available, then this is the first status to be reported instead of "received".
started	The addressed SCA routine has been started.
unconfirmed	SCA is technically successfully finalised by the PSU, but the authorisation resource needs a confirmation command by the TPP yet.
finalised	The SCA routine has been finalised successfully (including a potential confirmation command). This is a final status of the authorisation resource.
failed	The SCA routine failed. This is a final status of the authorisation resource.
exempted	SCA was exempted for the related transaction, the related authorisation is successful. This is a final status of the authorisation resource.

14.16 Account Access

Attribute	Type	Condition	Description
accounts	Array of Account Reference	Optional	Is asking for detailed account information. If the array is empty in a request, the TPP is asking for an accessible account list. This may be restricted in a PSU/ASPSP authorization dialogue. If the array is empty, also the arrays for balances, additionalInformation sub attributes or transactions shall be empty, if used.
balances	Array of Account Reference	Optional	Is asking for balances of the addressed accounts. If the array is empty in the request, the TPP is asking for the balances of all accessible account lists. This may be restricted in a PSU/ASPSP authorization dialogue. If the array is empty, also the arrays for accounts,

Attribute	Type	Condition	Description
			additionalInformation sub attributes or transactions shall be empty, if used.
transactions	Array of Account Reference	Optional	Is asking for transactions of the addressed accounts. If the array is empty in the request, the TPP is asking for the transactions of all accessible account lists. This may be restricted in a PSU/ASPSP authorization dialogue. If the array is empty, also the arrays for accounts, additionalInformation sub attributes or balances shall be empty, if used.
additional Information	Additional Information Access	Optional if supported by API provider	Is asking for additional information as added within this structured object. The usage of this data element requires at least one of the entries "accounts", "transactions" or "balances" also to be contained in the object. If detailed accounts are referenced, it is required in addition that any account addressed within the additionalInformation attribute is also addressed by at least one of the attributes "accounts", "transactions" or "balances".
availableAccounts	String	Optional if supported by API provider	The values "allAccounts" and "allAccountsWithOwnerName" are admitted. The support of the "allAccountsWithOwnerName" value by the ASPSP is optional.
availableAccounts WithBalance	String	Optional, if supported by API provider	The values "allAccounts" and "allAccountsWithOwnerName" are admitted. The support of the "allAccountsWithOwnerName" value by the ASPSP is optional.
allPsd2	String	Optional if supported by API	The values "allAccounts" and "allAccountsWithOwnerName" are admitted. The support of the "allAccountsWithOwnerName" value by the

Attribute	Type	Condition	Description
		provider	ASPSP is optional.

14.17 Additional Information Access

Attribute	Type	Condition	Description
ownerName	Array of Account Reference	Optional	Is asking for account owner name of the accounts referenced within. If the array is empty in the request, the TPP is asking for the account owner name of all accessible accounts. This may be restricted in a PSU/ASPSP authorization dialogue. If the array is empty, also the arrays for accounts, balances or transactions shall be empty, if used. The ASPSP will indicate in the consent resource after a successful authorisation, whether the ownerName consent can be accepted by providing the accounts on which the ownerName will be delivered. This array can be empty.

Remark for Future: In future, other additional informations might be addressable through new sub attributes of the additionalInformation consent attribute.

14.18 Account Reference

This type is containing any account identification which can be used on payload-level to address specific accounts. The ASPSP will document which account reference type it will support. Exactly one of the attributes defined as "conditional" shall be used.

> **Remark:** The currency of the account is needed, where the currency is an account characteristic identifying certain sub-accounts under one external identifier like an IBAN. These sub-accounts are separated accounts from a legal point of view and have separated balances, transactions etc.

Attribute	Type	Condition	Description
iban	IBAN	Conditional	

Attribute	Type	Condition	Description
bban	BBAN	Conditional	This data elements is used for payment accounts which have no IBAN.
pan	Max35Text	Conditional	Primary Account Number (PAN) of a card, can be tokenised by the ASPSP due to PCI DSS requirements.
maskedPan	Max35Text	Conditional	Primary Account Number (PAN) of a card in a masked form.
msisdn	Max35Text	Conditional	An alias to access a payment account via a registered mobile phone number.
currency	Currency Code	Optional	ISO 4217 Alpha 3 currency code

14.19 Account Details

Remark: The ASPSP shall give at least one of the account reference identifiers listed as optional below.

Attribute	Type	Condition	Description
resourceId	String	Conditional	This is the data element to be used in the path when retrieving data from a dedicated account, cp. Section 6.5.3 or Section 6.5.4 below. This shall be filled, if addressable resource are created by the ASPSP on the /accounts endpoint.
iban	IBAN	Optional	This data element can be used in the body of the Consent Request Message for retrieving account access consent from this payment account, cp. Section 6.3.1.1.
bban	BBAN	Optional	This data element can be used in the body of the Consent Request Message for retrieving account access consent from this account, cp. Section 6.3.1.1. This data elements is used for payment accounts which have no IBAN.

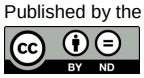
Attribute	Type	Condition	Description
msisdn	Max35Text	optional	An alias to access a payment account via a registered mobile phone number. This alias might be needed e.g. in the payment initiation service, cp. Section 5.3.1. The support of this alias must be explicitly documented by the ASPSP for the corresponding API Calls.
currency	Currency Code	Mandatory	Account currency
ownerName	Max140Text	Optional	Name of the legal account owner. If there is more than one owner, then e.g. two names might be noted here. For a corporate account, the corporate name is used for this attribute. Even if supported by the ASPSP, the provision of this field might depend on the fact whether an explicit consent to this specific additional account information has been given by the PSU.
name	Max70Text	Optional	Name of the account, as assigned by the ASPSP, in agreement with the account owner in order to provide an additional means of identification of the account.
displayName	Max70Text	Optional	Name of the account as defined by the PSU within online channels.
product	Max35Text	Optional	Product Name of the Bank for this account, proprietary definition
cashAccountType	Cash Account Type	Optional	ExternalCashAccountType1Code from ISO 20022

Attribute	Type	Condition	Description
status	String	Optional	Account status. The value is one of the following: • "enabled": account is available • "deleted": account is terminated • "blocked": account is blocked e.g. for legal reasons If this field is not used, than the account is available in the sense of this specification.
bic	BICFI	Optional	The BIC associated to the account.
linkedAccounts	Max70 Text	Optional	This data attribute is a field, where an ASPSP can name a cash account associated to pending card transactions.
usage	Max4 Text	Optional	Specifies the usage of the account - PRIV: private personal account - ORGA: professional account
details	Max500 Text	Optional	Specifications that might be provided by the ASPSP - characteristics of the account - characteristics of the relevant card
balances	Array of Balances	Conditional	
_links	Links	Optional	Links to the account, which can be directly used for retrieving account information from this dedicated account. Links to "balances" and/or "transactions" These links are only supported, when the corresponding consent has been already granted.

14.20 Card Account Details

Attribute	Type	Condition	Description
resourceId	String	Conditional	This is the data element to be used in the path when retrieving data from a dedicated account, cp. Section 6.6.2, Section 6.6.3 or 6.6.4 below. This shall be filled, if addressable resource are created by the ASPSP on the /card-accounts endpoint.
maskedPan	Max35Text	Mandatory	Primary Account Number (PAN) of the main card in masked form. This data element can be used in the body of the Consent Request Message for retrieving account access consent from this card, cp. Section 6.3.1.1.
currency	Currency Code	Mandatory	Account currency
ownerName	Max140Text	Optional	Name of the legal account owner. If there is more than one owner, then e.g. two names might be noted here. For a corporate account, the corporate name is used for this attribute. Even if supported by the ASPSP, the provision of this field might depend on the fact whether an explicit consent to this specific additional account information has been given by the PSU.
name	Max70Text	Optional	Name of the account, as assigned by the ASPSP, in agreement with the account owner in order to provide an additional means of identification of the account.
displayName	Max70Text	Optional	Name of the account as defined by the PSU within online channels.

Attribute	Type	Condition	Description
product	Max35Text	Optional	Product Name of the Bank for this account, proprietary definition
status	String	Optional	Account status. The value is one of the following: • "enabled": account is available • "deleted": account is terminated • "blocked": account is blocked e.g. for legal reasons If this field is not used, than the account is available in the sense of this specification.
usage	Max140 Text	Optional	Specifies the usage of the account - PRIV: private personal account - ORGA: professional account
details	Max500 Text	Optional	Specifications that might be provided by the ASPSP - characteristics of the account - characteristics of the relevant card
creditLimit	Amount	Optional	Defines the credit limit of the PSU aggregated for all cards related to this card account in total.
balances	Array of Balances	Optional	The specific card account balances associated to this card accounts.
_links	Links	Optional	Links to the cardAccount, which can be directly used for retrieving account information from this dedicated account. Links to "balances" and/or "cardTransactions" These links are only supported, when the corresponding consent has been already granted.



14.21 Balance Type

The following balance types are excluding credit limits unless the creditLimitIncluded element is present and equals true in the corresponding balance element.

Remark: This definition is following ISO20022 logic for defining balance types.

Type	Description
closingBooked	Balance of the account at the end of the pre-agreed account reporting period. It is the sum of the opening booked balance at the beginning of the period and all entries booked to the account during the pre-agreed account reporting period. For card-accounts, this is composed of • invoiced, but not yet paid entries
expected	Balance composed of booked entries and pending items known at the time of calculation, which projects the end of day balance if everything is booked on the account and no other entry is posted. For card accounts, this is composed of • invoiced, but not yet paid entries, • not yet invoiced but already booked entries and • pending items (not yet booked)
openingBooked	Book balance of the account at the beginning of the account reporting period. It always equals the closing book balance from the previous report.
interimAvailable	Available balance calculated in the course of the account 'servicer's business day, at the time specified, and subject to further changes during the business day. The interim balance is calculated on the basis of booked credit and debit items during the calculation time/period specified. For card-accounts, this is composed of • invoiced, but not yet paid entries, • not yet invoiced but already booked entries

Type	Description
interimBooked	Balance calculated in the course of the account servicer's business day, at the time specified, and subject to further changes during the business day. The interim balance is calculated on the basis of booked credit and debit items during the calculation time/period specified.
forwardAvailable	Forward available balance of money that is at the disposal of the account owner on the date specified.
nonInvoiced	Only for card accounts, to be defined yet.

14.22 Balance

Attribute	Type	Condition	Description
balanceAmount	Amount	Mandatory	
balanceType	Balance Type	Mandatory	
creditLimitIncluded	Boolean	Optional	A flag indicating if the credit limit of the corresponding account is included in the calculation of the balance, where applicable.
lastChangeDateTime	ISODateTime	Optional	This data element might be used to indicate e.g. with the expected or booked balance that no action is known on the account, which is not yet booked.
referenceDate	ISODate	Optional	indicates the date of the balance
lastCommitted Transaction	Max35Text	Optional	entryReference of the last commited transaction to support the TPP in identifying whether all PSU transactions are already known.

14.23 Account Report

Attribute	Type	Condition	Description
booked	Array of transactions	Conditional	Shall be contained if bookingStatus parameter is set to "booked" or "both".
pending	Array of transactions	Optional	Not contained if the bookingStatus parameter is set to "booked" or "information".
information	Array of transactions	Optional	Only contained if the bookingStatus is set to "information" and if supported by ASPSP.
_links	Links	Mandatory	The following links might be used within this context: • account (mandatory) • first (optional) • next (optional) • previous (optional) • last (optional)

14.24 Transactions

Attribute	Type	Condition	Description
transactionId	String	Optional	Can be used as access-ID in the API, where more details on an transaction is offered. If this data attribute is provided this shows that the AIS can get access on more details about this transaction using the GET Transaction Details Request as defined in Section 6.5.5
entryReference	Max35Text	Optional	Is the identification of the transaction as used e.g. for reference for deltafunction on application level. The same identification as for example used within camt.05x messages.
endToEndId	Max35Text	Optional	Unique end to end identity.

Attribute	Type	Condition	Description
mandateId	Max35Text	Optional	Identification of Mandates, e.g. a SEPA Mandate ID
checkId	Max35Text	Optional	Identification of a Cheque
creditorId	Max35Text	Optional	Identification of Creditors, e.g. a SEPA Creditor ID
bookingDate	ISODate	Optional	The Date when an entry is posted to an account on the ASPSPs books.
valueDate	ISODate	Optional	The Date at which assets become available to the account owner in case of a credit
transactionAmount	Amount	Mandatory	The amount of the transaction as billed to the account.
currencyExchange	Array of Report Exchange Rate	Optional	
creditorName	Max70Text	Optional	Name of the creditor if a "Debited" transaction
creditor Account	Account Reference	Conditional	
creditorAgent	BICFI	Optional	
ultimate Creditor	Max70Text	Optional	
debtorName	Max70Text	Optional	Name of the debtor if a "Credited" transaction
debtorAccount	Account Reference	Conditional	
debtorAgent	BICFI	Optional	
ultimateDebtor	Max70Text	Optional	

Attribute	Type	Condition	Description
remittance Information Unstructured	Max140Text	Optional	
remittance Information Unstructured Array	Array of Max140Text	Optional	**Remark for Future:** In version 2.0 these two unstructured remittance fields might be merged.
remittance Information Structured	Max140Text	Optional	Reference as contained in the structured remittance reference structure (without the surrounding XML structure). **Remark For Future**: This field will be re-typed in a future version of the interface to the structured data type Remittance or might be omitted. For migration reasons, this is not supported in version 1.3.x.
remittance Information Structured Array	Array of Remittance	Optional	**NOTE**: More details about the Remittance Data Type will be published in an Errata in due course. For usage of the fields e.g. for domestic elements, Berlin Group should be contacted. This would enable to publish usage of structured remittance information in the domestic payment documentation, cp. [XS2A-DP].
additionalInformation	Max500Text	Optional	Might be used by the ASPSP to transport additional transaction related information to the PSU
additionalInformation Structured	Structured Additional Information	Conditional	Is used if and only if the bookingStatus entry equals "information". Every active standing order related to the dedicated payment account result into one entry.

Attribute	Type	Condition	Description
purposeCode	Purpose Code	Optional	
bank TransactionCode	Bank Transaction Code	Optional	Bank transaction code as used by the ASPSP and using the sub elements of this structured code defined by ISO20022. For standing order reports the following codes are applicable: "PMNT-ICDT-STDO" for credit transfers, "PMNT-IRCT-STDO" for instant credit transfers "PMNT-ICDT-XBST" for cross-border credit transfers "PMNT-IRCT-XBST" for cross-border real time credit transfers and "PMNT-MCOP-OTHR" for specific standing orders which have a dynamical amount to move left funds e.g. on month end to a saving account
proprietaryBank TransactionCode	Max35Text	Optional	proprietary bank transaction code as used within a community or within an ASPSP e.g. for MT94x based transaction reports
balanceAfter Transaction	Balance	Optional	This is the balance after this transaction. Recommended balance type is interimBooked.
_links	Links	Optional	The following links could be used here: transactionDetails for retrieving details of a transaction.

14.25 Structured Additional Information Data Type

Attribute	Type	Condition	Description
standingOrderDetails	Standing Order Details	Mandatory	Details of underlying standing orders.

14.26 Standing Order Details Data Type

Attribute	Type	Condition	Description
startDate	ISODate	Mandatory	The first applicable day of execution starting from this date the first payment was/will be executed.
endDate	ISODate	Optional	The last applicable day of execution If not given, it is an infinite standing order.
executionRule	String	Optional	"following" or "preceding" supported as values. This data attribute defines the behavior when a transaction date resulting from a standing order falls on a weekend or bank holiday. The payment is then executed either the "preceding" or "following" working day.
withinAMonthFlag	Boolean	Optional	This element is only used in case of frequency equals "monthly". If this element equals false it has no effect. If this element equals true, then the execution rule is overruled if the day of execution would fall into a different month using the execution rule. **Example**: executionRule equals "preceding", dayOfExecution equals "02" and the second of a month is a Sunday. In this case, the transaction date would be on the last day of the month before. This would be overruled if withinAMonthFlag equals true and the payment is processed on Monday the third of the Month.

Attribute	Type	Condition	Description
			Remark: This attribute is rarely supported in the market.
frequency	Frequency Code	Mandatory	The frequency of the recurring payment resulting from this standing order.
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". Example: An execution on January, April and October each year is addressed by ["1". "4", "10"].
multiplicator	Numerical	Optional	This is multiplying the given frequency resulting the exact frequency, e.g. Frequency=weekly and multiplicator=3 means every 3 weeks. **Remark**: This attribute is rarely supported in the market.
dayOfExecution	Max2Text	Optional	"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.
limitAmount	Amount	Conditional	limitAmount Amount limit for fund skimming, e.g. skim all funds above this limit to savings account, i.e. typically a specific periodic payments with fixed remaining amount rather than

Attribute	Type	Condition	Description
			fixed transaction amount. Amount may be zero as well as below zero, i.e. negative. Constraints: transactionAmount needs to be zero and bankTransactionCode needs to specify PMNT-MCOP-OTHR for fund skimming

14.27 Card Account Report

Attribute	Type	Condition	Description
booked	Array of Card Transactions	Mandatory	Card transaction which have been booked already to the card account.
pending	Array of Card Transactions	Optional	
_links	Links	Mandatory	The following links might be used within this context: • cardAccount (mandatory) • first (optional) • next (optional) • previous (optional) • last (optional)

14.28 Card Transactions

Attribute	Type	Condition	Description
cardTransactionId	Max35Text	Optional	Unique end to end identity.

Attribute	Type	Condition	Description
terminalId	Max35Text	Optional	Identification of the Terminal, where the card has been used.
transactionDate	ISODate	Optional	date of the actual card transaction
acceptorTransactionDateTime	ISODateTime	Optional	Timestamp of the actual card transaction within the acceptance system
bookingDate	ISODate	Optional	booking date of the related booking on the card account
transactionAmount	Amount	Mandatory	The amount of the transaction as billed to the card account.
currencyExchange	Array of Report Exchange Rate	Optional	For card accounts, this often is restricted by the ASPSP to use only one exchange rate.
originalAmount	Amount	Optional	Original amount of the transaction at the Point of Interaction in orginal currency
markupFee	Amount	Optional	Any fee related to the transaction in billing currency.
markupFeePercentage	String	Optional	Percentage of the involved transaction fee in relation to the billing amount, e.g. "0.3" for 0,3%
cardAcceptorId	Max35Text	Optional	Identification of the Card Acceptor (e.g. merchant) as given in the related card transaction.
cardAcceptor Address	Address	Optional	Address of the Card Acceptor as given in the related card transaction.

Attribute	Type	Condition	Description
cardAcceptorPhone	Phone Number	Optional	Merchant phone number
merchantCategory Code	Merchant Category Code	Optional	Merchant Category Code of the Card Acceptor as given in the related card transaction.
maskedPAN	Max35Text	Optional	The masked PAN of the card used in the transaction.
transactionDetails	Max140Text	Optional	Additional details given for the related card transactions.
invoiced	Boolean	Optional	Flag indicating whether the underlying card transaction is already invoiced.
proprietaryBank TransactionCode	Max35Text	Optional	proprietary bank transaction code as used within a community or within an ASPSP e.g. for MT94x based transaction reports

14.29 Report Exchange Rate

Attribute	Type	Condition	Description
sourceCurrency	Currency Code	Mandatory	Currency from which an amount is to be converted in a currency conversion.
exchangeRate	String	Mandatory	Factor used to convert an amount from one currency into another. This reflects the price at which one currency was bought with another currency.
unitCurrency	Currency Code	Mandatory	Currency in which the rate of exchange is expressed in a currency exchange. In the example 1EUR = xxxCUR, the unit currency is EUR.

Attribute	Type	Condition	Description
targetCurrency	Currency Code	Mandatory	Currency into which an amount is to be converted in a currency conversion.
quotationDate	ISODate	Mandatory	Date at which an exchange rate is quoted.
contractIdentification	String	Optional	Unique identification to unambiguously identify the foreign exchange contract.

14.30 Payment Exchange Rate

Attribute	Type	Condition	Description
unitCurrency	String	Optional	Currency in which the rate of exchange is expressed in a currency exchange. In the example 1EUR = xxxCUR, the unit currency is EUR.
exchangeRate	String	Optional	Factor used to convert an amount from one currency into another. This reflects the price at which one currency was bought with another currency.
contractIdentification	String	Optional	Unique identification to unambiguously identify the foreign exchange contract.
rateType	String	Optional	Specifies the type used to complete the currency exchange. Only SPOT, SALE and AGRD is allowed.

14.31 Geo Location

Format using [RFC2426], i.e. GEO:<latitude>;< longitude >.

14.32 Frequency Code

The following codes from the "EventFrequency7Code" of ISO 20022 are supported:

- Daily

- Weekly
- EveryTwoWeeks
- Monthly
- EveryTwoMonths
- Quarterly
- SemiAnnual
- Annual
- MonthlyVariable

14.33 Charge Bearer

Type	Description
DEBT	All transaction charges are to be borne by the debtor.
CRED	All transaction charges are to be borne by the creditor.
SHAR	In a credit transfer context, means that transaction charges on the sender side are to be borne by the debtor, transaction charges on the receiver side are to be borne by the creditor. In a direct debit context, means that transaction charges on the sender side are to be borne by the creditor, transaction charges on the receiver side are to be borne by the debtor.
SLEV	Charges are to be applied following the rules agreed in the service level and/or scheme.

This is following ChargeBearerType1Code from ISO20022.

14.34 Other ISO-related basic Types

The following codes and definitions are used from ISO 20022

- **Purpose Code**: ExternalPurpose1Code
- **Cash Account Type**: ExternalCashAccountType1Code
- **Bank Transaction Code**: ExternalBankTransactionDomain1Code
- **BICFI**: BICFIIdentifier
- **IBAN**: IBAN2007Identifier

  Pattern: [A-Z]{2}[0-9]{2}[A-Z0-9]{1-30}

- **BBAN**: BBANIdentifier

- **Phone Number**: PhoneNumber
- **Merchant Category Code**: Category code conform to ISO 18245
- **Service Level Code**: ExternalServiceLevel1Code

The following code is a concatenated code from ISO20022

- BankTransactionCode: This code type is concatenating the three ISO20022 Codes Domain Code, Family Code and SubFamiliy Code by hyphens, resulting in “DomainCode”-“FamilyCode”-“SubFamilyCode”.

  Example: PMNT-RCDT-ESCT defining a transaction assigned to the PayMeNT Domain (PMNT), belonging to the family of ReceivedCreDitTransfer (RCDT) that facilitated the EuropeanSEPACreditTransfer (ESCT)

For all codes used in JSON structures, not the abbreviation defined for XML encoding, but the name of the code is used as value.

The following Codes are used from other ISO standards:

- **Currency Code**: Codes following ISO 4217 Alpha 3
- **Country Code:** Two characters as defined by ISO 3166

Further basic ISO data types:

- **ISODateTime**: A particular point in the progression of time defined by a mandatory date and a mandatory time component, expressed in either UTC time format (YYYY-MM-DDThh:mm:ss.sssZ), local time with UTC offset format (YYYY-MM-DDThh:mm:ss.sss+/-hh:mm), or local time format (YYYY-MMDDThh:mm:ss.sss). These representations are defined in "XML Schema Part 2: Datatypes Second Edition - W3C Recommendation 28 October 2004" which is aligned with ISO 8601.
- **ISODate**: A particular point in the progression of time in a calendar year expressed in the YYYY-MM-DD format.

15 References

[XS2A-OR] NextGenPSD2 XS2A Framework, Operational Rules, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, version 1.3, published 21 December 2018

[XS2A-DP] NextGenPSD2 XS2A Framework, Domestic Payment Definitions, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, current version

[XS2A-COFC] NextGenPSD2 XS2A Framework, Extended Services, Confirmation of Funds Consent Service, Version 2.0, 01 March 2019

[XS2A-RSNS] NextGenPSD2 XS2A Framework, Extended Services, Resource Status Notification Service, Version 1.0, 01 March 2019

[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

[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

[signHTTP] Signing HTTP messages, Network Working Group, Internet Draft version 10, https://datatracker.ietf.org/doc/draft-cavage-http-signatures/

[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

[RFC2426] Dawson, F. and T. Howes, T., "vCard MIME Directory Profile", September 1998, https://tools.ietf.org/html/rfc2426

[RFC3230] Mogul, J. and A. Van Hoff, "Instance Digests in HTTP", RFC 3230, DOI 10.17487/RFC3230, January 2002,https://www.rfc-editor.org/info/rfc3230

[RFC4648] Josefsson, S.," The Base16, Base32, and Base64 Data Encodings", October 2006, https://tools.ietf.org/html/rfc4648

[RFC5843] Bryan, A, "Additional Hash Algorithms for HTTP Instance Digests", RFC 5843, DOI 10.17487/RFC5843, April 2010, https://www.rfc-editor.org/info/rfc5843

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", October 2012, https://tools.ietf.org/html/rfc6749

[RFC7231] R. Fielding, J. Reschke, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content

[RFC7807] M. Nottingham, Akamai, E. Wilde, „Problem Details for HTTP APIs“, March 2016, https://tools.ietf.org/html/rfc7807