THE Berlin GROUP

A EUROPEAN STANDARDS INITIATIVE

Joint Initiative on a PSD2 Compliant XS2A Interface

XS2A Interface Interoperability Framework

Implementation Guidelines

Version 0.99

02.10.2017

Notice

This Specification has been prepared by the Participants of the Joint Initiative pan-European PSD2-Interface Interoperability* (hereafter: Joint Initiative). Permission is hereby granted to use the document solely for the purpose of implementing the Specification subject to the following conditions: (i) that none of the participants of the Joint Initiative nor any contributor to the Specification shall have any responsibility or liability whatsoever to any other party from the use or publication of the Specification; (ii) that one cannot rely on the accuracy or finality of the Specification; and (iii) that the willingness of the participants of the Joint Initiative to provide the Specification does not in any way convey or imply any responsibility for any product or service developed in accordance with the Specification and the participants of the Joint Initiative as well as the contributors to the Specification specifically disclaim any such responsibility to any party.

Implementation of certain elements of this Specification may require licenses under third party intellectual property rights, including without limitation, patent rights. The Participants of the Joint Initiative and any other contributors to the Specification are 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. **This Specification is provided "AS IS", "WHERE IS" and "WITH ALL FAULTS", and no participant in the Joint Initiative makes any warranty of any kind, express or implied, including any implied warranties of merchantability, non-infringement of third party intellectual property rights (whether or not the Participants of the Joint Initiative have been advised, have reason to know, or are otherwise in fact aware of any information), and fitness for a particular purpose (including any errors and omissions in the Specification).**

To the extent permitted by applicable law, neither the Participants of the Joint Initiative nor any contributor to the Specification shall be liable to any user of the Specification for any damages (other than direct actual out-of-pocket damages) under any theory of law, including, without limitation, any special, consequential, incidental, or punitive damages, nor any damages for loss of business profits, business interruption, loss of business information, or other monetary loss, nor any damages arising out of third party claims (including claims of intellectual property infringement) arising out of the use of or inability to use the Specification, even if advised of the possibility of such damages. Participation to the Joint Initiative does not imply either endorsement of any of the solutions identified in this Specification or a commitment to implement them.

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 the Specification.

---

* The 'Joint Initiative pan-European PSD2-Interface Interoperability' brings together participants of the Berlin Group with additional banks (ASPSPs) and payment associations in Europe.

Contents

1 Introduction

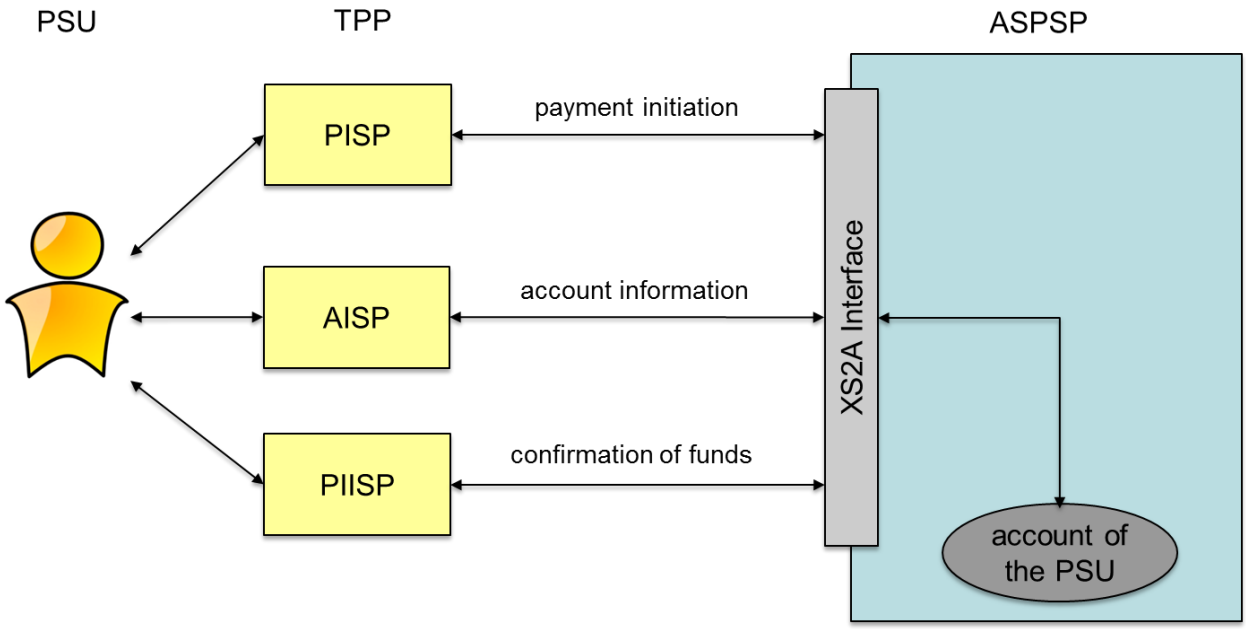
1.1 Background

With [PSD2] the European Union has published a new directive on payment services in the internal market. Member States have 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]:



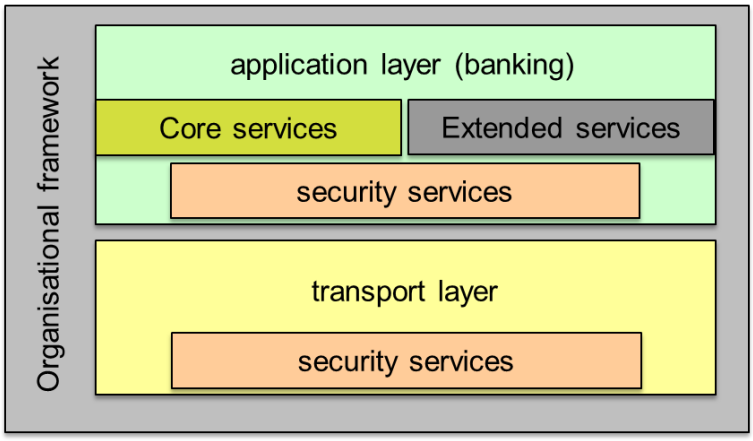
Further requirements on the implementation and usage of this interface are defined by a Regulatory Technical Standard (short RTS) to be published by the European Banking Authority (short EBA) and to be endorsed by the European Commission.

1.2 XS2A Interface Specification

This document is part of the 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 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 Chapter 2 and requirements on the transport layer in Chapter 3. In Chapter 4, guiding principles for the definition of the XS2A interface and the API structure with API endpoints and permitted access methods are described. Chapter 5 then specifies in detail how a Payment Initiation Service Provider (PISP) can initiate

payments within the Berlin Group XS2A. Chapter 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 potentially within both services are specified in Chapter 7. Chapter 8 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 Chapter 9. Following these chapters with functional chapters, the Chapter 10 to Chapter 11 specify core payment structure and general complex data structures.

1.4 Document History

Version	Change/Note	Approved
0.99	Market consultation draft of the Berlin Group XS2A Interface Framework	NextGen PSD2 Taskforce, 27 September 2017

2 Character Sets and Notations

The character set is UTF 8 encoded. This specification up to now is only using the basic data elements “string”, “boolean” "ISODateTime", "ISODate", "UUID" and “integer”. Further basic types and restrictions on character sets might follow after the market consultation process.

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. This TLS-connection is set up and controlled 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

4.1 Guiding Principles

4.1.1 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 https level (https header)
- message parameters by defining the resource path (URL path information) and
- message parameters as part of the https body.

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

- a full ISO20022 payment initiation (pain.001 message) with the corresponding payment initiation report (pain.002 message) or
- account information message (camt.05x message)

is contained.

The following principle is applied when defining the API:

Message parameters as part of the https header:

- Definition of the content syntax
- Certificate and Signature Data where needed
- TPP Redirect information
- 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,

Message parameters as part of the https body:

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

4.1.2 Error Information

If necessary the ASPSP will communicate errors to the TPP within a request/response dialogue. The error information is sent to the TPP using the data element TPP Message Information with the attribute category set to “ERROR”[2]. The attribute code indicates the error 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.

This error element can be embedded in all JSON based response messages[3] of the Berlin Group XS2A Interface. This is not mentioned in the following API call definitions. If an error information is sent to the TPP the transaction status is always set to "Rejected" where applicable.

Example:

```
{
 “transaction_status”: “Rejected”,
 "tpp_messages": [ {
        "category": "ERROR",
        "code": "TOKEN_MISSING",
        “text”: “additional text information of the ASPSP”
   } ]
}
```

[2] The final version of this specification will also deal with tpp_messages of category warning or information, used in cases where the requested service is not mandatorily stopped.

[3] The final version of this specification will also cover XML based error information in cases where XML encoded data was submitted to the ASPSP within the request message body.

4.2 Signing Messages at Application Layer

If requested by the ASPSP the TPP has to sign the request messages to be sent to the ASPSP.

For signing a request message the TPP has to calculate an electronic signature for the request message as defined by [signHTTP].

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.

4.3 Optional Usage of OAuth2 as a Pre-Step

The XS2A API will allow ASPSP to implement OAuth2 as a Pre-Step to this specification for the consent management 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.

When using OAuth2, the API calls will work with an access token instead of using the PSU credentials. The only admitted versions of the token grant step is the "user password grant" or the "authorization flow" of OAuth2. Since it is supposed that an SCA method is needed to handle the consent process of the PSU, then only the authorization flow will apply, which mandates a redirect flow to the TPP.

4.4 XS2A Interface API Structure

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

https://{provider}/v1/{service}

using additional content parameters {parameters}

where

- {provider} is the host of the XS2A API, which is not further mentioned
- v1 is denoting the version one of the interface
- {service} has the values consents, payments, accounts or funds-confirmations, eventually extended by more information on product types and request scope
- {parameters} are content attributes defined in the following in an JSON or XML encoding
    - XML encoding appears only when ISO20022 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 in the following in the categories

- Path: Attributes encoded in the Path, e.g. “payments/sepa-credit-transfers” for {resource}
- Header: Attributes encoded in the https 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 and JSON
    - XML encoding appears only, when camt.05x messages (reports, notifications or account statements)or pain.002 payment status messages are transported. Pain.002 messages will only be delivered in cases where the payment initiation was performed by using pain.001 messages.
    - all other response bodies are encoded in JSON

The HTTPS response codes which might be used in this XS2A interface are specified in Section 11.21. This is not repeated for every API call definition.

Remark: For JSON based responses, this specification only defines body attributes which are responded from ASPSP to TPP following POST or PUT API calls and which have not been already used in the API call requests by the TPP. The ASPSP is free to return the whole addressed resource within the response, following usual REST methodologies.

Remark: The body parameters in JSON encoding are defined in **snake_case** syntax. It will be discussed after market consultation, whether to transform the syntax to **UpperCamelCase** for the final version of this specification.

4.5 API Access Methods

The following table gives an overview on the HTTPS access methods supported by the API endpoints and by resources created through this API. It further defines, whether this method support is mandated by this specification or whether it is an optional feature. 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/{consent-ID} applies only if the endpoint /v1/consents is supported at all.

Please note that all methods submitted by a TPP may only apply to resources which have been created by the same TPP before.

Enpoints/Resources	Method	Condition	Description
payments/{product-name}	POST	Mandatory	Creates a payment initiation resource addressable under {resource-id} with all data relevant for the corresponding payment product.. This is the first step in the API to initiate the related payment
payments/{product-name}/{payment-id}	PUT	Mandatory for Embedded SCA Approach	Updates data on the payment resource if needed. It may authorise a payment within the Embedded SCA Approach where needed.
payments/{product-name}/{payment-id}	GET	Mandatory	Reads the details of an initiated payment.
payments/{product-name}/{payment-id}/status	GET	Mandatory	Reads the transaction status of the payment
accounts	GET	Mandatory	Reads all available account ids, in addition with balance if the optional parameter “with-balance” is used in the method.
accounts/{account-id}	GET	Mandatory	Reads the balance of a given account if the additional parameter “with-balance” is added, otherwise just gives detailed information about the addressed account.

Enpoints/Resources	Method	Condition	Description
accounts/{account-id}/transactions	GET	Mandatory	Reads a transaction list For a given account, additional parameters are e.g. the attributes “date_from” and “date_to”. If the attribute “with-balance” is used, the ASPSP will add balances to the transaction list. The latter might be provided by the ASPSP anyhow, if transaction lists without balances are not supported.
accounts/{account-id}/transactions/{transaction-id}	GET	Optional	Reads transaction details of an addressed transaction.
consents	POST	Mandatory if the full consent management is not covered by the use of OAuth2	Creates a consent resource, defining access rights to dedicated accounts of a given PSU-ID. These accounts must be addressed explicitly in the method as parameters.
	GET	Optional, will be covered in a future version.	Shows all access consents for a given PSU-ID
consents/{consent-id}	GET	Mandatory if the full consent management is not covered by the use of OAuth2	Reads the exact definition of the given consent resource {consent-id}
	PUT	Mandatory for Embedded SCA Approach	Updates data on the consent resource, authorises a consent within the Embedded SCA Approach where needed.
	DELETE	Mandatory	Deletes a created consent.
consents/{consent-id}/status	GET	Mandatory	Reads the transaction status of the

Enpoints/Resources	Method	Condition	Description
			addressed consent resource.
consents/all-accounts	POST	Optional	Creates a consent resource, defining access rights to the list of all payment accounts of a given PSU-ID. If the optional parameter "with-balance" is used, then the requested consent is on all accounts together with balances. Remark. This service can be extended also to transaction lists. In this case, the PSU might then reduce the rights while giving his consent to the ASPSP on an ASPSP interface. This is only supported for the Redirect or Decoupled SCA Approach.
funds	POST	Mandatory	Checks whether funds are available for a payment transaction on an account linked with a given tuple cardissuer/cardnumber, or an IBAN and TPP respectively

Remark: Note that the {account-id} parameters can be tokenized 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 tokenization is managed by the ASPSP.

4.6 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.

In the response to these first requests and to all succeeding requests within the services, the ASPSP can leave a hyperlink together with a "tag" for the semantics of this hyperlink. This hyperlink then can be either a relative link 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/sepa-credit-transfers/transaction/asdf-asdf-asdf-1234.

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 an sca method are to be posted to the resource addressed by this link to authorise the payment.

The steering hyperlinks are transported in the "_links" data element. It may contain one or several hyperlinks.

The following table gives an overview on the steering hyperlinks used in this specification as well as additional data elements, which are always transported in the context of the corresponding hyperlink. Further links might be added by ASPSP implementations.

Hyperlink	**Additional Link Related Data**	**Description**
redirect		Routing information for a redirect scenario.
update_psu_identification		The link to the resource, which needs to be updated by a PSU identification.
update_psu_authentication		The link to the payment initiation resource, which need to be updated by a PSU password and eventually the PSU identification if not delivered yet.
select_authentication_method	authentication _methods	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.
authorise_transaction	sca_challenge _data, chosen_sca _method	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.
account_link		A link to an account, which can be directly used for retrieving account information from this dedicated account.
balances		A link to the resource providing the balance of a dedicated account.
transactions		A link to the resource providing the transaction history of a dediated amount.

Hyperlink	**Additional Link Related Data**	**Description**
self		The link to the resource created by the undergoing request. This link can be used to retrieve the resource data
status		The link to retrieve the transaction status of a resource.
first_page_link		Navigation link for account reports.
second_page_link		Navigation link for account reports.
current_page_ link		Navigation link for account reports.
last_page_link		Navigation link for account reports.

5 Payment Initiation Service

Remark: The API design differs across the various SCA approaches (Embedded, Redirect 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 raise the readability of the specification.

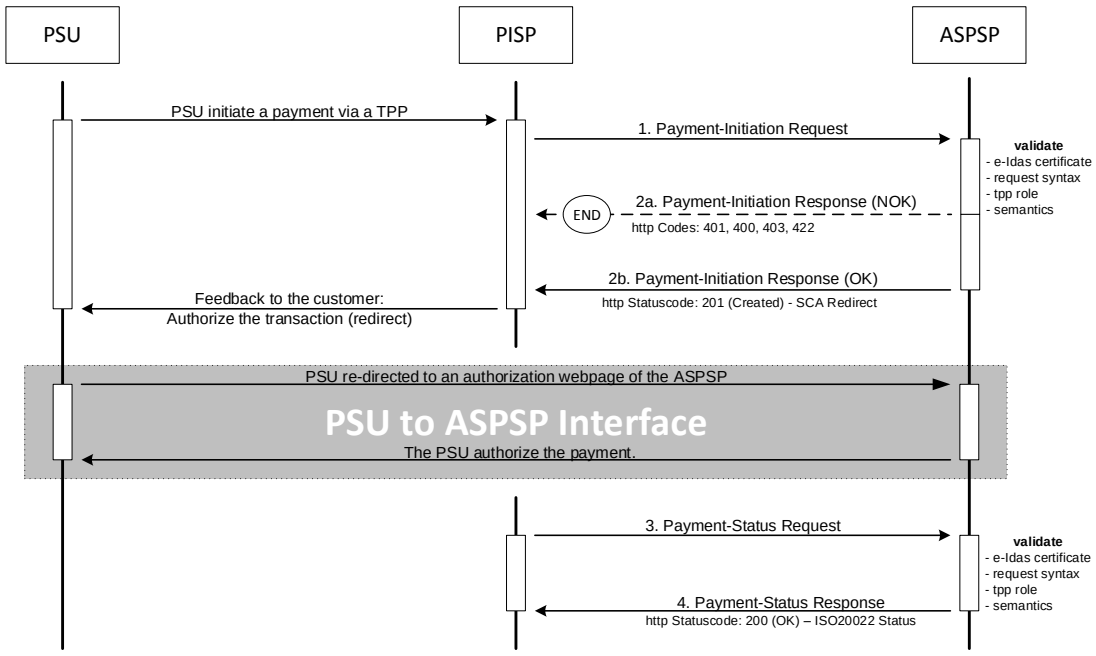
5.1 Payment Initiation Flows

The payment initiation flow depends heavily from 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.

Redirect SCA Approach

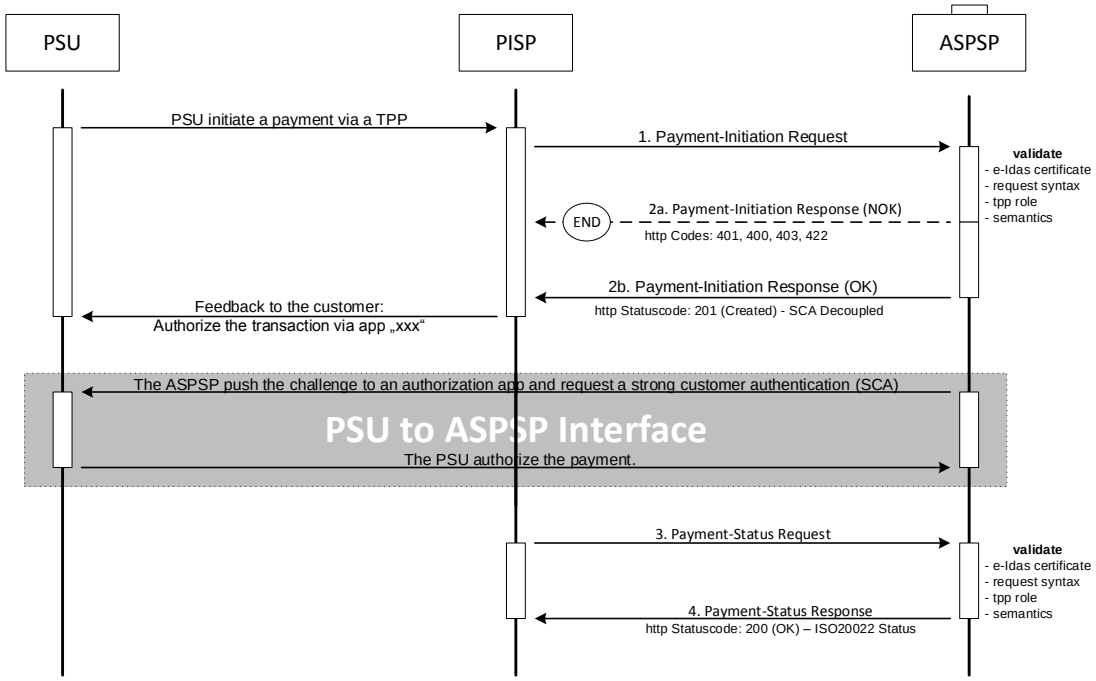
If the ASPSP supports the Redirect SCA Approach, the message flow within the payment initiation service is simple. The Payment Initiation Request is followed by a redirection to the ASPSP SCA authorization site. A status request might be requested by the TPP after the session is re-redirected to the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



Decoupled SCA Approach

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”.

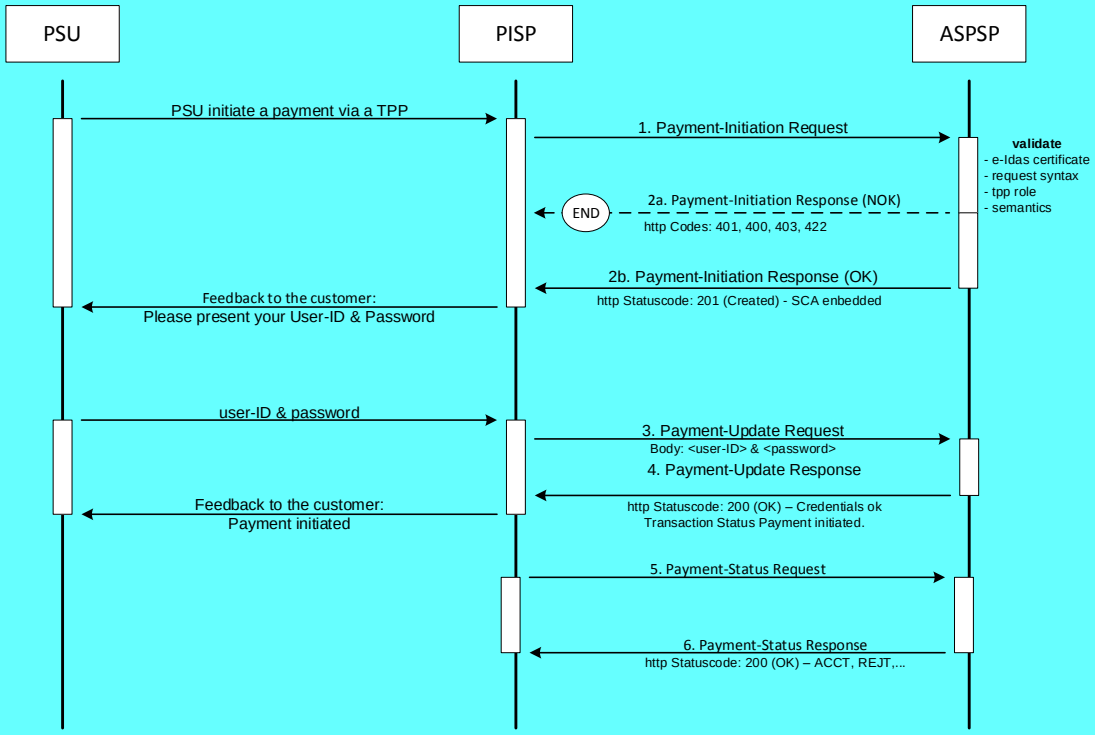
After the SCA having been processed between ASPSP and PSU, the TPP then needs to ask for the result of the transaction.



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

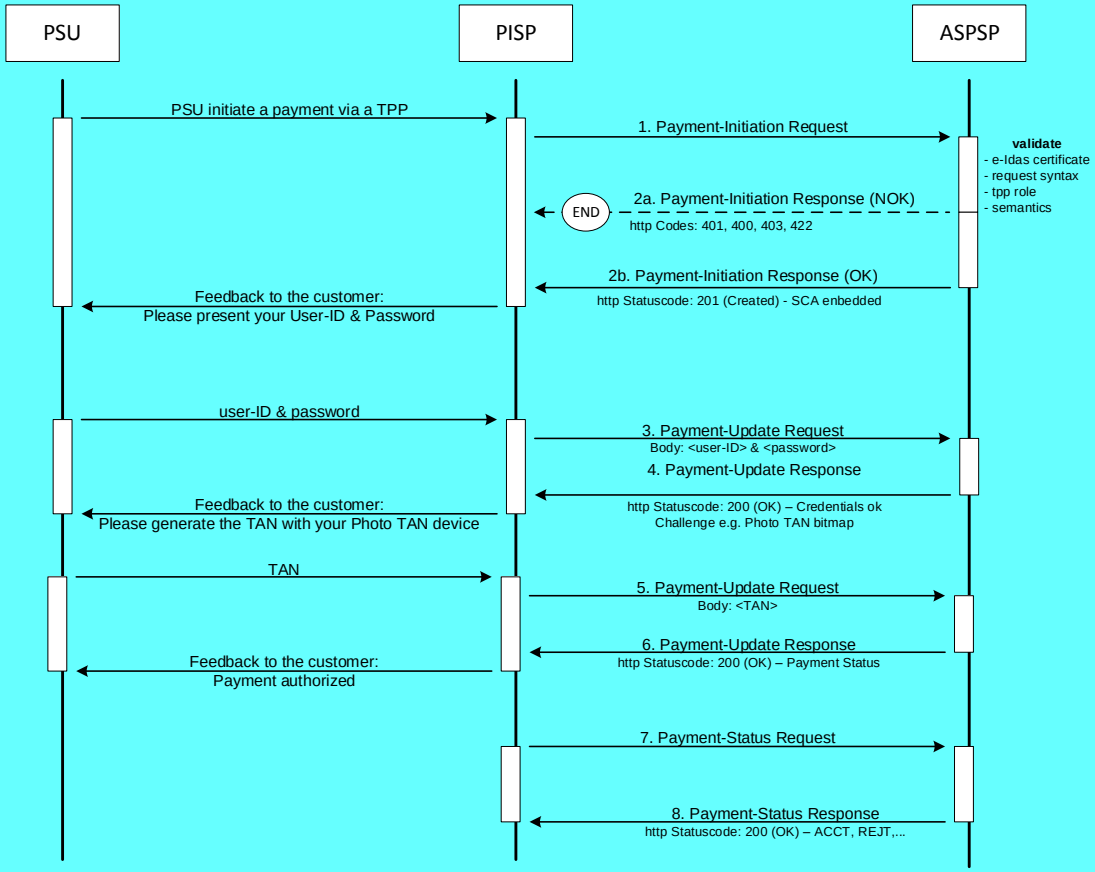
In the following, several exemplary flows are shown, where the ASPSP has chosen to process the SCA methods through the PISP – ASPSP interface. In any case, the PSU normally will need to authenticate himself with a first factor, before any account or SCA method details will be available to the PISP. So 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, the sequence of the PSU authentication with the first authentication factor is omitted. This applies also for all examples for the Embedded SCA Approach.



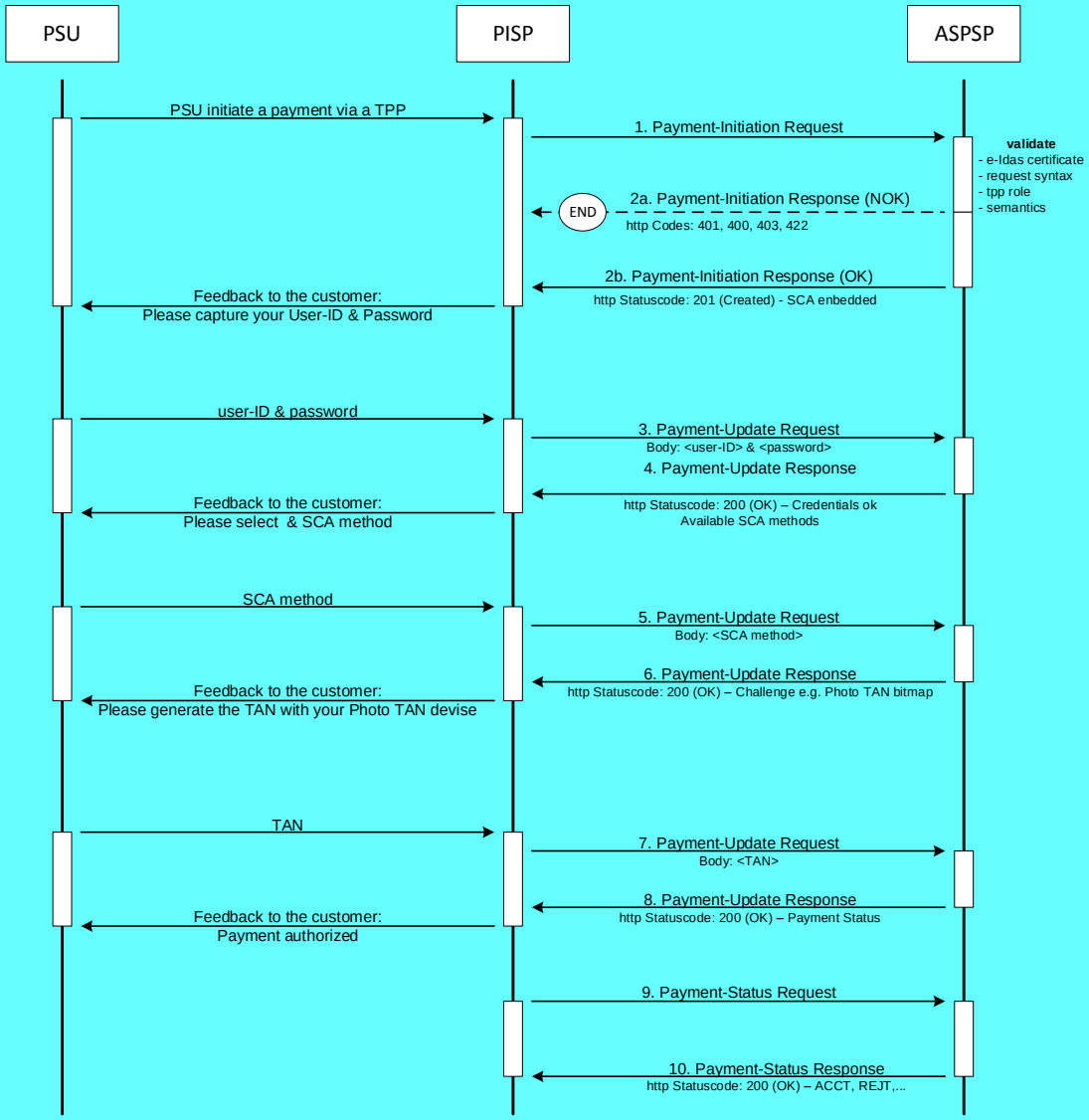
Embedded SCA Approach with only one SCA method available

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.



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.



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 where 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 where 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 methods 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 current status of the payment initiation, e.g. “AcceptedTechnicalValidation” 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 where 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.

5.2 Data Overview Payment Initiation Service

The following table defines the technical description of the abstract data model as defined [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 https parameters, resp. are taken from e-IDas 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 HHTPS 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 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 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 API Calls, condition defined by the ASPSP

Data element	**Attribute encoding**	**Location**				**Usage**							
		Path	Header	Body	Certificate	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	
Transaction Identification	Process-ID (unique id of TPP regarding PSD2 article 46b, 47)		x			m		m		m		m	
Request Identification	Request-ID		x			m		m		m		m	
Resource ID	Resource-ID as part of a hyperlink (a _links element)	x		x[4]			m	m		m		m	
Access Token (from optional OAuth 2)	Authorization Bearer		x			c		c		c		c	
Request Timestamp	Date		x			m		m		m		m	
TPP Signing Certificate Data	certificate		x			c		c		c		c	
TPP Electronic Signature	signature		x			c		c		c		c	
Further signature related data			x			c		c		c		c	
Service Type		x				m		m		m		m	
Response Code			x				m		m		m		m

[4] Is transported in body only in response message.

Data element	Attribute encoding	Location				Usage							
		Path	Header	Body	Certificate	Init Req.	Init Resp.	Upd. Req.	Upd. Resp	Auth. Req.	Auth Resp.	Stat. Req.	Stat. Resp
Transaction Status	transaction_status			x			m		m		m		m
PSU Message Information	psu_message			x			o		o		o		o
TPP Message Information	tpp_messages			x			o		o		o		o
PSU Identification	PSU-ID		x			c		c					
Corporate Identification	Corporate-ID		x			c		c		c		c	
Corporate ID Type	Corporate-ID-Type		x			c		c		c		c	
PSU Password	psu_data.password			x				c					
PSU Authentication Data	sca_authentication_data			x						m			
SCA Challenge Data	sca_challenge_data			x			c		c				
IP Address PSU	PSU-IP-Address		x			m							
PSU User Agent	PSU-User-Agent[5]		x			o							
GEO Information	PSU-Geo-Location		x			o							
Additional Device Information	PSU-Additional-Device-Information[6]		x			o							
Redirect URL ASPSP	_links.redirect			x			c						
Payment Product	payment-product	x				m		m		m		m	

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

[5] This field transports key information for risk management like browser type or PSU device operating system

[6] Details on this field will be added in the final version of the specification. This field might e.g. transport additional device related information like in 3D Secure 2.0 of EMVCo.

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.

5.3 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

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 Annex A. 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.

Request Header

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	application/json
Process-ID	UUID	Mandatory	ID of the transaction as determined by the initiating party.

Attribute	Type	Condition	Description
Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-ID	String	Conditional	Might be mandated in the ASPSP's documentation. Is not contained if the optional OAuth Pre-Step was performed.
PSU-Corporate-ID	String	Conditional	Might be mandated in the ASPSP's documentation. Only used in a corporate context.
Authorization Bearer	String	Conditional	Is contained only, if the optional OAuth2 Pre-Step was performed.
Consent-ID	String	Optional	This data element may be contained, if the payment initiation transaction is part of a combined AIS/PIS service. This then contains the consent id of the related AIS consent.
PSU-Agent	string	Optional	The forwarded Agent header field of the http request between PSU and TPP.
PSU-IP-Address	string	Mandatory	The forwarded IP Address header field consists of the corresponding http request IP Address field between PSU and TPP.
PSU-Geo-Location	string	Optional	The forwarded Geo Location header field of the corresponding http request between PSU and TPP if available.
signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard https header element date and time

Request Body

The payment data to be transported in the request body are dependent of the chosen API endpoint. Some standard definitions related to the above mentioned standard products are defined in Section 10 of this document. Further definitions might be done community or ASPSP specific. In Annex A, a list of community specific payment product definitions and links regarding community/ASPSP specific payment product definitions are given. ASPSP or community definitions should reuse standard attribute names.

Response Header

The Location field is used as link to the created resource. No other specific requirements.

Response Body

Attribute	Type	Condition	Description
transaction_status	Transaction status	Mandatory	The values defined in Section 11.12 might be used.
sca_methods	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 an hyperlink of type "select_authentication_methods" contained in the response body. These methods shall be presented towards the PSU for selection by the TPP.
chosen_sca_method	authentication object	conditional	This data element is only contained in the response if the APSPS 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.
sca_challenge_data	challenge	Conditional	It is containded in addition to the data element chosen_sca_method if challenge data is needed for SCA.

Attribute	Type	Condition	Description
_links	links	Mandatory	A list of hyperlinks to be recognized by the TPP. **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): "redirect" : In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser. "update_psu_identification" : The link to the payment initiation resource, which needs to be updated by the psu identification. This might be used in a redirect or decoupled approach, where the PSU ID was missing in the first request.
			"update_psu_authentication" : The link to the payment initiation resource, which need to be updated by a psu password and eventually the psu identification if not delivered yet. This is used in a case of the Embedded SCA approach. "select_authentication_method" : 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. This link contained under exactly the same conditions as the data element “authentication_methods”, see above. "authorise_transaction" : The link to the payment initiation resource, where the "Payment Authorization Request" is sent to. This is the link to the resource which will authorize the payment by checking the SCA authentication data within the Embedded SCA approach.

Attribute	Type	Condition	Description
			"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.
psu_message	string	Optional	Text to be displayed to the PSU
tpp_messages	Array of Message	Optional	

Example

Request

```
POST https://api.testbank.com/v1/payments/sepa-credit-transfers
Content-Encoding      gzip
Content-Type          application/json
Process-ID            3dc3d5b3-7023-4848-9853-f5400a64e80f
Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-IP-Address        192.168.8.78
PSU-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

{
   "instructed_amount” : {“currency” : “EUR” , “amount” : "123"},
   "debtor_account" : { "iban":"DE23100100101234567 89"},
   “creditor” : { “name” : “Merchant123”} ,
   "creditor_account”: {“iban”:“DE23100120020123456789”},
   “remittance_information_unstructured”  :  “Ref  Number  Merchant-
123456”
}
```

Response in case of a redirect

Response Code 201

Response Body

```
{
  "transaction_status" : "Received",
  "_links" {
        "redirect" :  "www.testbank.com/asdfasdfasdf",
        "self"  :          "/v1/payments/sepa-credit-transfers/1234-
wertiq-983"
  }
}
```

Response in case of the decoupled approach

Response Code 201

Response Header:

Response Body

```
{
  "transaction_status" : "Received",
  "_links" {
     "update_psu_identification":"/v1/payments/sepa-credit-
transfers/1234-wertiq-983",
     "self" : "/v1/payments/sepa-credit-transfers/1234-wertiq-983"
    }
}
```

Response in case of the embedded approach

Response Code 201

```
{
    "transaction_status" : "Received",
    "_links" {
        "update_psu_authentication"   :   "/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.

Path

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. Annex A, Section 13.2

Header

The same header as in Section 5.3.1, only the content type indicates XML encoding ("application/xml").

Request

The request body consists of a pain.001 structure. Further details are defined in Section 10.

Response

The same response as in Section 5.3.1

Example

Request

```
POST https://api.testbank.com/v1/payments/xml-sepa-credit-transfers

Content-Encoding      gzip
Content-Type          application/json
Process-ID            "PI-123456789"
PSU-IP-Address        "192.168.8.78"
PSU-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 Body

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

5.4 Get Status Request

Call

GET /v1/payments/{payment-product}/{resource-id}/status

Can check the status of a payment initiation.

Path

Attribute	Type	Description
payment -product	string	

resource-id	string	

Request Header

Attribute	Type	Condition	Description
Request-ID	UUID	Mandatory	
Process-ID	UUID	Mandatory	
Authorization Bearer	String	Conditional	Is contained only, if the optional OAuth Pre-Step was performed.
signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard http date and time tag.

Request Body

No body.

Response Body in Case of JSON based endpoint

Attribute	Type	Condition	Description
transaction_status			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.

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 is returned as a pain.002 structure using the XML schema definitions as provided by EPC.

Example for JSON based endpoint

Request

```
GET https://api.testbank.com/v1/payments/sepa-credit-
transfers/qwer3456tzui7890/status
Accept                application/json
Process-ID            3dc3d5b3-7023-4848-9853-f5400a64e80f
Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date                  Sun, 06 Aug 2017 15:04:07 GMT
```

Response Code 200

```
Content-Type          application/json
{
  “transaction_status” : "AcceptedCustomerProfile"
}
```

Example for XML based endpoint

```
GET        https://api.testbank.com/v1/payments/pain.001-sepa-credit-
transfers/qwer3456tzui7890/status

Accept                application/xml
Process-ID            3dc3d5b3-7023-4848-9853-f5400a64e80f
Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date                  Sun, 06 Aug 2017 15:04:07 GMT

Response Code 200

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

```
......<GrpSts>ACCT</GrpSts>
....</OrgnlGrpInfAndSts>
....<OrgnlPmtInfAndSts>
......<OrgnlPmtInfId>BIPI-123456789RI-123456789</OrgnlPmtInfId>
......<OrgnlNbOfTxs>1</OrgnlNbOfTxs>
......<OrgnlCtrlSum>123</OrgnlCtrlSum>
......<PmtInfSts>ACCT</PmtInfSts>
....</OrgnlPmtInfAndSts>
..</CstmrPmtStsRpt>
</Document>
```

6 Account Information Service

This specification foresees different types of account information services:

- Transaction reports for (a list of) accounts including balances if applicable.
- Balances of a list of accounts.
- A list of accessible accounts.

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

- 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 above),
  - the Multiplicity of the Account Information Service, i.e. a once-off or recurring access, and
  - in the latter case on the duration of the consent in days or as maximally offered by the ASPSP and optionally the frequency of a recurring request.

  Please note that the initial consent to the “List of Accounts” is always a once-off consent.

  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 the detailed access rights, a consent-id token and delivers aliases to the accounts used during the account access for addressing the accounts.

- 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,
  - in case of transaction reports as Account Information type additionally
    - the addressed account number and

- the period of the transaction report
- in addition optionally a transaction identification of the last transaction that was received by the TPP, indicating the request to get the account reports since last call, if offered by the ASPSP

- the preferred formats of the transaction reports.

In case of a once-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.

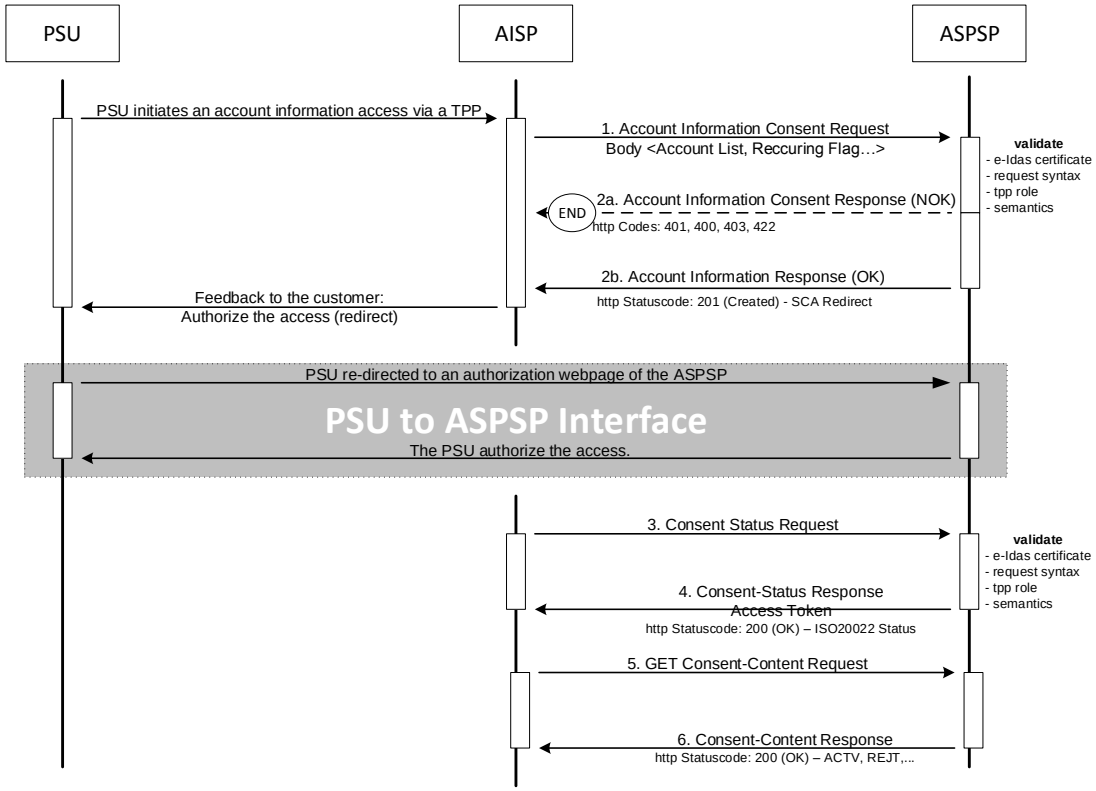
Remark: The use of OAuth2 can be twofold within the Account Information Service. The OAuth2 process can either just abstract the PSU credentials for the sole access as within the Payment Initiation Service. In this case, the consent mechanism as such is still handled within this XS2A API. Or the OAuth2 process is managing the whole consent process for AIS. In the latter case, an ASPSP would offer only the Read Account Data part of this XS2A specification.

6.1 Account Information Service Flows

6.1.1 Account Information Consent Flow

Redirect SCA Approach

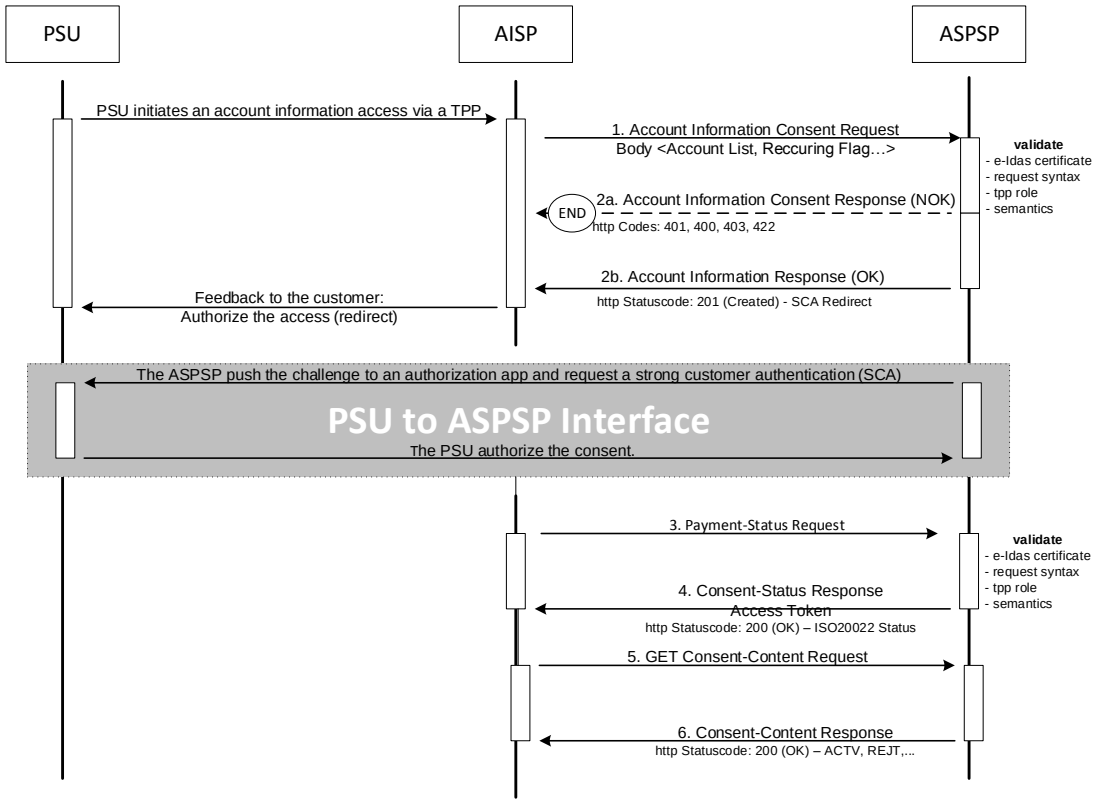
If the ASPSP supports the Redirect SCA Approach, the message flow within the Account Information Consent sub-service is simple. The Account Information Consent Request is followed by a redirection to the ASPSP SCA authorisation site. A status or content request on the created consent resource might be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



Decoupled SCA Approach

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.

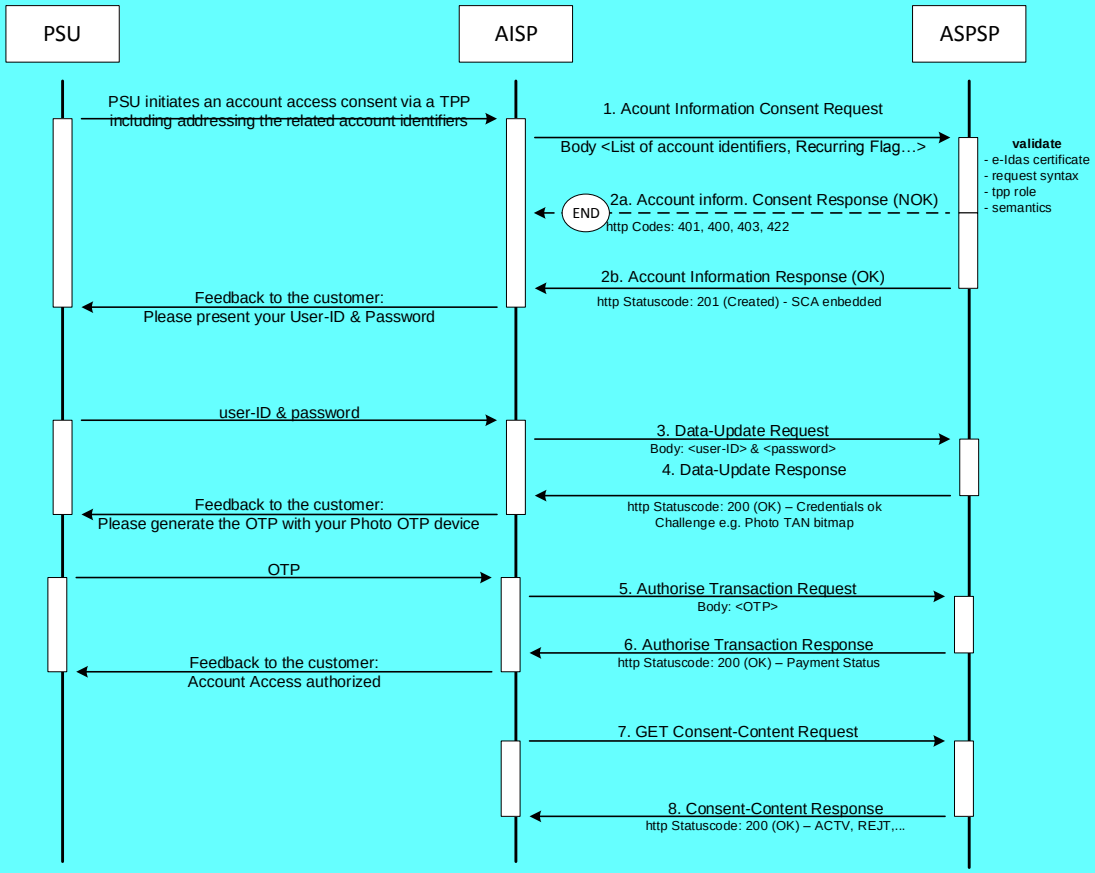


Embedded SCA Approach with only one SCA method available

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.

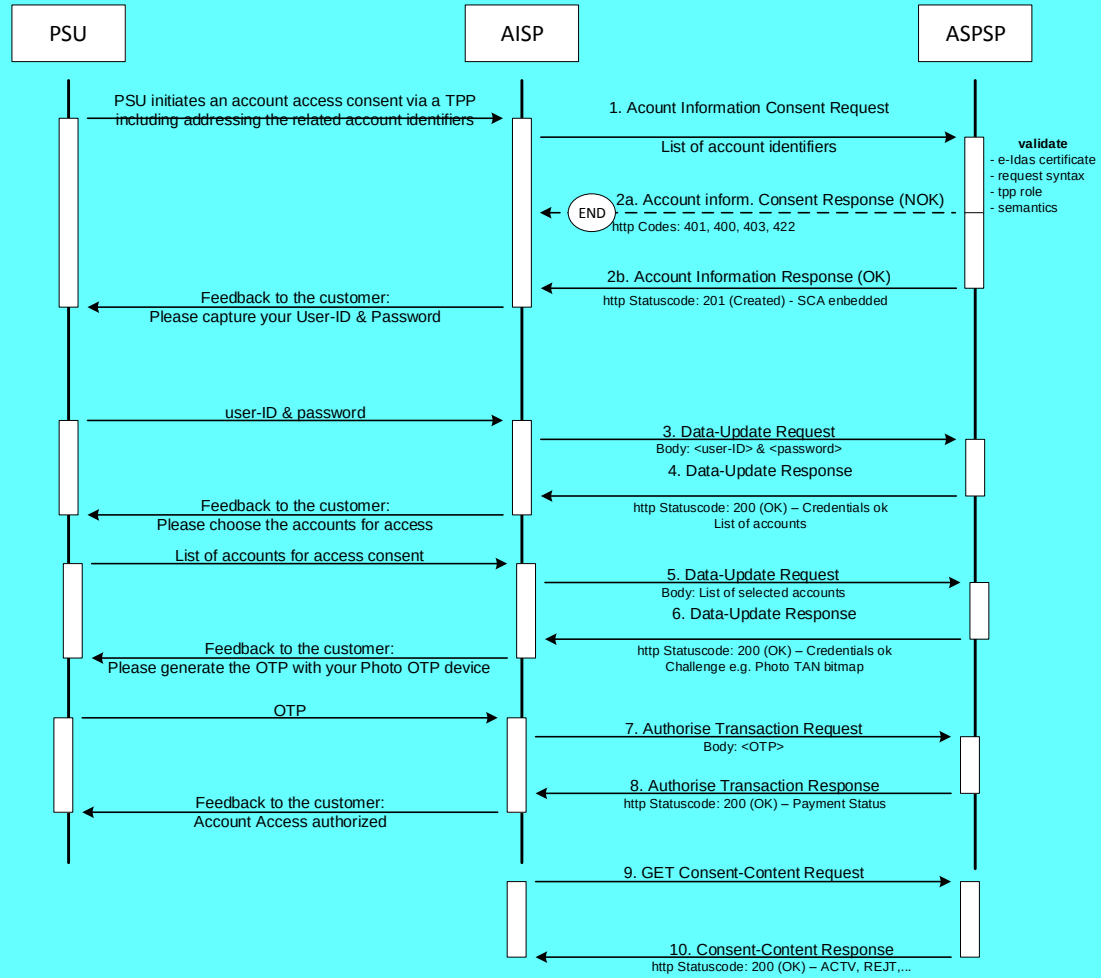
Remark: In case where OAuth2 is requested by the ASPSP as a pre-step to replace PSU-ID and password by an access token, the sequence of the PSU authentication with the first authentication factor is omitted. This applies for all examples for the Embedded SCA Approach.

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.



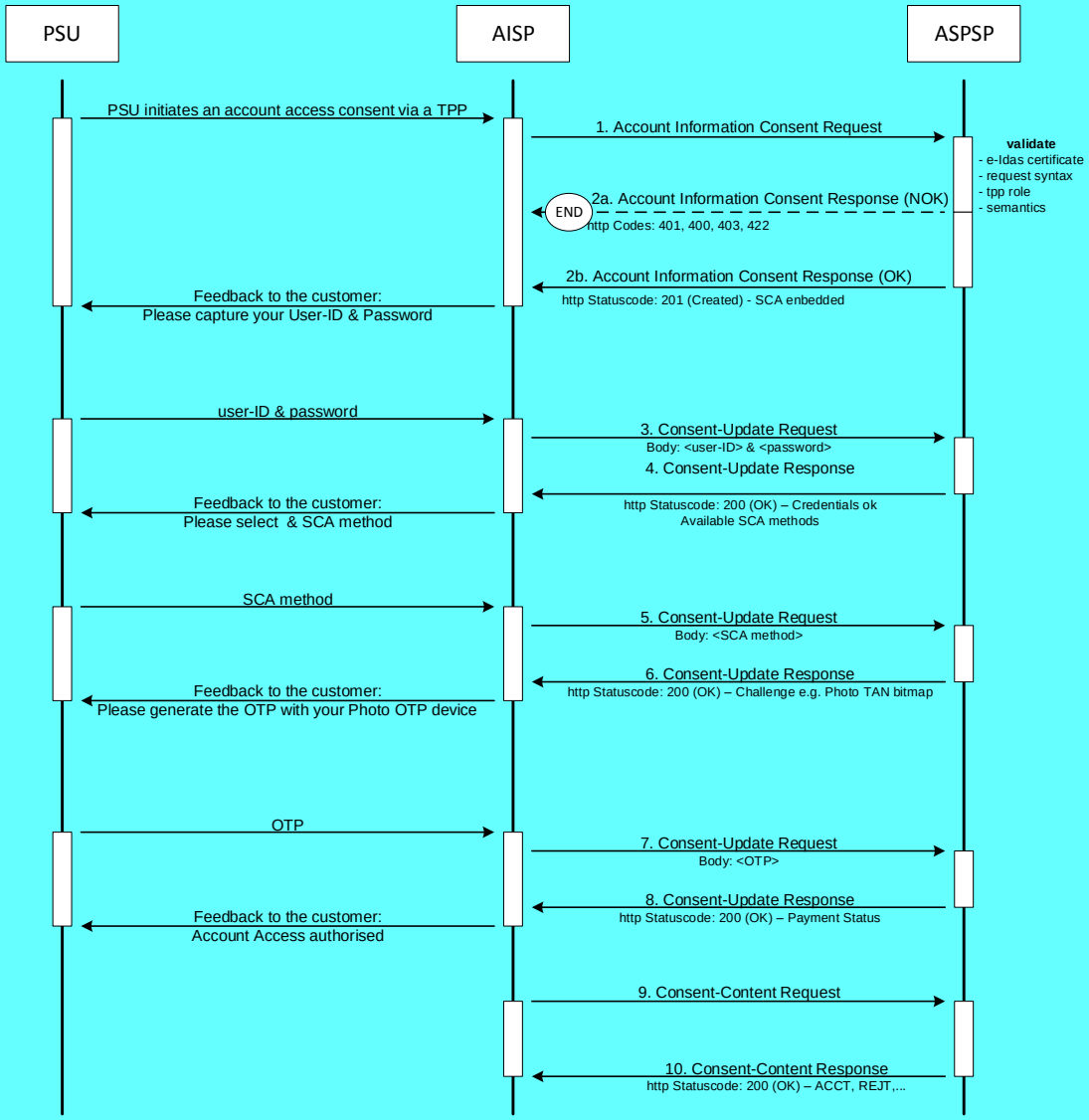
Embedded SCA Approach with only one SCA method available and with Account Unknown

In case where the TPP does not know the Account IDs of the PSU involved in the account access consent, the ASPSP might offer a list of accounts, from which the PSU can choose.



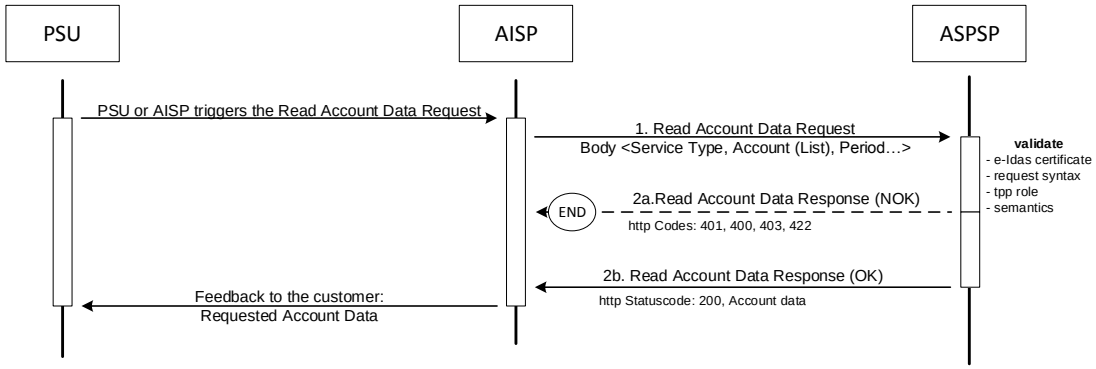
Embedded SCA Approach with Selection of an SCA method

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.



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 https parameters, resp. are taken from e-IDas 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 HHTPS POST, PUT, DELETE and GET commands. The calls are divided into the following calls:
    - Information 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 Get Data Request is the request to retrieve Account Information data.
    - 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

Data element	**Attribute encoding**	**Location**				**Usage**									
		Path	Header	Body	Certificate	Inform. Cons.Req.	Inform. 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	
Transaction Identification	Process-ID		x			m		m		m		m		m	
Request Identification	Request-ID		x			m		m		m		m		m	
Resource ID	Resource-ID	x					m	m		m		m		m	
Access Token (from optional OAuth2)	Authorization Bearer		x			c		c		c		c		c	
Request Timestamp	Date		x			m		m		m		m		m	
TPP Signing Certificate Data	certificate		x			c		c		c		c		c	
TPP Signing Electronic Signature	signature		x			c		c		c		c		c	
Further signature related data			x			c		c		c		c		c	
Service Type		x				m		m		m		m		m	
Response Code			x				m		m		m		m		m
Transaction Status	transaction_status			x			m		m		m		m		
PSU Message Information	psu_message			x			o		o		o		o		o

Data element	Attribute encoding	Location				Usage									
		Path	Header	Body	Certificate	Inform. Cons.Req.	Inform. Cons. Resp.	Upd. Data Req.	Upd. Data Resp	Auth. Req.	Auth Resp.	Status Req.	Status Resp.	Read Data Req.	Read Data Resp
TPP Message Information	tpp_messages			x			o		o		o		o		o
PSU Identification	PSU-ID		x			c		c							
Corporate Identification	PSU-Corporate-ID		x			c		c		c		c			
PSU Password	psu_data.password			x				c							
PSU Authentication Data	psu_data.authentication			x						m					
SCA Challenge Data	sca_challenge_data			x			c		c						
IP Address PSU	PSU-IP-Adress		x			m									
PSU Agent	PSU Agent		x			o									
GEO Information	PSU-Geo-Location		x			o									
Redirect URL ASPSP	_links.redirect			x			c								
PSU Account	psu_account			x										c	
PSU Account List	access_accounts			x		m									
Date From	date_from	x												c	
Date To	date_to	x												c	
Validity Period	valid_until			x		m									
Frequency	frequency_per_day			x		m									
Recurring Indicator	recurring_indicator			x		m								c	
Combined service	combined_service_indicator			x		m									

Remark: The upper table refers to the "Acccount Information Consent Request" referring dedicated amounts, cp. Section 6.3.1.1. The body data for the corresponding request referring all accounts is containing the "with-balance"-attribute only.

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.

6.3 Account Information Consent Management

In this section, the Account Information Consent process is defined for the XS2A Interface. This process might be handled alternatively fully by an OAuth2 process, depending on the implementation decisions of the ASPSP.

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.

Path

No parameters on path level.

Request Header

Attribute	Type	Condition	Description
Process-ID	UUID	Mandatory	ID of the transaction as determined by the initiating party.
Request-ID	UUID	Mandatory	
PSU-ID	String	Conditional	Might be mandated in the ASPSP's documentation, if Oauth is not chosen as Pre-Step.
PSU-Corporate-ID	String	Conditional	Might be mandated in the ASPSP's documentation. Only used in a corporate

Attribute	Type	Condition	Description
			context.
Authorization Bearer	String	Conditional	If Oauth has been chosen as Pre-Step to get the agreement of the PSU for the Consent Management Process
signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
Certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard https header element for Date and Time of the TPP Request.

Request Body

Attribute	Type	Condition	Description
access_accounts	Array of single account access	Mandatory	Requested access service per account.
recurring_indicator	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
valid_until	string	Mandatory	This parameter is requesting a valid until date for the requested consent. The content is the local ASPSP date in ISODate Format, e.g. 2017-10-30
frequency_per_day	Integer	Mandatory	This field indicates the requested maximum frequency for an access per day. For a once-off access, this attribute is set to “1”.

Attribute	Type	Condition	Description
combined_service _indicator	boolean	Mandatory	If "true" indicates that a payment initiation service will be addressed in the same "session".

Response Header

Location hyperlink for the status of the resource.

Response Body

Attribute	Type	Condition	Description
sca_methods	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 an hyperlink of type "select_authentication_methods" contained in the response body. These methods shall be presented towards the PSU for selection by the TPP.
chosen_sca _method	authentication object	conditional	This data element is only contained in the response if the APSPS 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.
sca_challenge _data	challenge	Conditional	It is containded in addition to the data element chosen_sca_method if challenge data is needed for SCA.
_links	links	Mandatory	A list of hyperlinks to be recognized by

Attribute	Type	Condition	Description
			the TPP. Type of links admitted in this response (which might be extended by single ASPSPs as indicated in its XS2A documentation): "redirect" : In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser. "update_psu_identification" : The link to the payment initiation resource, which needs to be updated by the psu identification. This might be used in a redirect or decoupled approach, where the PSU ID was missing in the first request. "update_psu_authentication" : The link to the account information resource, which needs to be updated by a psu password and eventually the psu identification if not delivered yet. This is used in a case of the Embedded SCA approach.
			"select_authentication_method" : This is a link to a resource, where the TPP can select the applicable second factorstrong customer authentication methods for the PSU, if there were several available authentication methods. This link is only contained under exactly the same conditions as the data element “authentication_methods”, see above.,.
			"authorise_transaction" : The link to the resource, where the "Transaction Authorization Request" is sent to. This is the link to the resource which will authorize the transaction by checking the SCA authentication data within the Embedded SCA approach.

Attribute	Type	Condition	Description
			“status”: The link to retrieve the transaction status of the account information consent..
psu_message	string	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-Encoding      gzip
Content-Type          application/json
Process-ID            3dc3d5b3-7023-4848-9853-f5400a64e80g
Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7756
PSU-IP-Address        192.168.8.78
PSU-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_accounts": [
     {"iban": "DE23100100101234567789",
      “access” : [“balance”,”transactions“]},
     {"iban": "DE23100100101234567788",
      “access” :[“balance”],
     {"pan": "12345678912345",
      "access": ["transactions"]  }
  ],
 "recurring_indicator": "true",
 "valid_until": "2017-11-01",
 “frequency_per_day” : “4”
}
```

Response in case of a redirect

Response Code 200

Response Header:

Location "v1/consents/1234-wertiq-983"

Response Body


```
{
  "transaction_status" : "Received",
  "_links" {
  "redirect" :  "www.testbank.com/authentication/1234-wertiq-983"
}
```


Response in case of the decoupled approach

Response Code 201

Response Header:

Location "v1/consents/1234-wertiq-983"

Response Body


```
{
  "transaction_status" : "Received",
  "_links" {
    "update_psu_identification": "/v1/consents/1234-wertiq-983"
  }
}
```


Response in case of the embedded approach

Response Code 201


```
{
  "transaction_status" : "Received",
  "_links" {
    "update_psu_authentication" : "/v1/consents/1234-wertiq-983"
  }
}
```


6.3.1.2 Consent Request on Account List

Call

POST /v1/consents/account-list

Creates an account information consent resource at the ASPSP to return a list of all accessible accounts.

Path

No parameters on path level.

Request Header

Attribute	Type	Condition	Description
Process-ID	UUID	Mandatory	ID of the transaction as determined by the initiating party.
Request-ID	UUID	Mandatory	
PSU-ID	String	Conditional	Might be mandated in the ASPSP's documentation, if OAuth is not chosen as Pre-Step.
PSU-Corporate-ID	String	Conditional	Might be mandated in the ASPSP's documentation. Only used in a corporate context.
Authorization Bearer	String	Conditional	If OAuth has been chosen as Pre-Step to get the agreement of the PSU for the Consent Management Process
signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard https header element for Date and Time of the TPP Request.

Request Body

Attribute	Type	Condition	Description
with-balance	Boolean	Mandatory	If the value equals “true”, then the consent request is on the list of all payment accounts inclusive the balance.

Attribute	Type	Condition	Description
			If the value equals “false”, then the consent request is on the list of accounts only.

Response Header

Location is under /v1/consents

6.3.2 Get Status Request

Call

GET /v1/consents/{consent-id}/status

Can check the status of an account information consent resource.

Path

Attribute	Type	Description
resource-ID	String	

Request Header

See above.

Request Body

No body.

Response Body

Attribute	Type	Condition	Description
transaction_status			This is the “authentication status” of the consent. Remark: To check the validity, the call “GET Consent Request” is used, cp. Section 6.3.3

Example

Request

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

Response

Response Code 200

```
{
  "transaction_status" : " AcceptedTechnicalValidation ",
}
```

6.3.3 Get Consent Request

Call

GET /v1/consents/{consent-id}

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

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

Request Header

See above for Get Status Request

Request Body

No body.

Response Body

Attribute	Type	Condition	Description
access_accounts	Array of single	Mandatory	

Attribute	Type	Condition	Description
	account access		
recurring_indicator	boolean	Mandatory	
valid_until	string	Mandatory	
frequency_per_day	Integer	Mandatory	
transaction_status	String	Mandatory	
consent_status	string	Mandatory	The following code values are permitted "empty", "valid", "blocked", "expired", "deleted". These values might be extended by ASPSP by more values.

Example

Request

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

Response

```
{
  "access_accounts": [
      {"iban": "DE23100100101234567789",
        "access" : ["balance","transactions"]},
      {"iban": "DE23100100101234567788",
       "access" :["balance"],
      {"pan": "12345678912345",
       "access": ["transactions"]  }
  ],
 "recurring_indicator": "true",
 "valid_until": "2017-11-01",
 "frequency_per_day" : "4",
 "transaction_status" : "AcceptedTechnicalValidation",
```

```
  "consent_status": "valid"
}
```

Remark: It needs to be considered yet, whether a link to the accounts might be added directly in the consent object to have a short cut to the single accounts available. If so, this short cut will be considered in the final version of this specification.

6.4 Delete an Account Information Consent Object

The TPP can delete an account information consent object if needed.

Call

DELETE /v1/consents/{consent-id}

Deletes a given consent.

Path

Attribute	Type	Description
Consent-id	String	Contains the resource id of the consent to be deleted.

Request Header

Attribute	Type	Condition	Description
Process-ID	UUID	Mandatory	ID of the transaction as determined by the initiating party. .
Request-ID	UUID	Mandatory	
Authorization Bearer	String	Conditional	Is contained only, if the optional OAuth Pre-Step was performed.

No Request Body.

No Response Body

Example

Request

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

Content-Encoding      gzip
Content-Type          application/json
Process-ID            3dc3d5b3-7023-4848-9853-f5400a64e812
Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date                  Sun, 13 Aug 2017 17:05:37 GMT
```

Response

Response Code: 204

6.5 Read Account Data Request

6.5.1 Read Account List

Call

GET /v1/accounts/ {path-options}

Reads a list of 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 consent-id, respectively the OAuth2 token

Path

Attribute	Type	Description
with-balance	String	If contained, this function reads the list of accessible payment accounts including the balance.
psu-involved	string	If contained, it is indicated that a PSU has directly asked this account access in real-time. The PSU then might be involved in an additional consent process, if the given consent is not any more sufficient.

Request Header

Attribute	Type	Condition	Description
Process-ID	UUID	Mandatory	ID of the transaction as determined by the initiating party.
Request-ID	UUID	Mandatory	
Consent-ID	String	Conditional	Shall be contained if “Establish Consent Transaction” was performed via this API before.
PSU-ID	String	Conditional	To be used, if no OAuth Pre-Step was performed.
Authorization Bearer	String	Conditional	Is contained only, if the optional OAuth Pre-Step was performed.

Response Body

Attribute	Type	Condition	Description
account_list	Array of account	Mandatory	

Example

Response

```
{[
      {“id” :  “3dc3d5b3-7023-4848-9853-f5400a64e80f”,
       “iban” :  “DE23100100101234567 89”,
       “account_type” : “Main Account”,
       “currency” :  “EUR”
       “_links” : {
          “balances”    :    “/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/balances”,
          “transactions”   :   “/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f/transactions”}
      },
      {“id” : “3dc3d5b3-7023-4848-9853-f5400a64e81g”,
       "iban": "DE23100100101234567 88",
       “account_type” : “US Dollar Account”,
       “currency” : “USD”,
       “_links” : {
          “balances”    :    “/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e81g/balances” }
      },
      {“id” : “1234567890-12345”,
       “pan” : “1234567890-12345”,
       “account_type”: “Credit Card”,
       “currency” : “EUR”,
       “_links” : {
          “transactions”        :        “/v1/accounts/1234567890-
12345/transactions”   }
       }
]}
```

6.5.2 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 tokenized 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.

Remark: If the ASPSP is not providing the "GET Account List" call, then the ASPSP must accept e.g. the PSU IBAN as account-id in this call.

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

Path

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 “id” attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Request Header

Attribute	Type	Condition	Description
Process-ID	UUID	Mandatory	ID of the transaction as determined by the initiating party. In case of a once off read data request, this Process-ID equals the Process-ID of the corresponding Account Information Consent Request, cp. Section 6.3.1.
Request-ID	UUID	Mandatory	
Consent-ID	String	Mandatory	
PSU-ID	String	Conditional	To be used, if no OAuth Pre-Step was performed and if a list of account balances or a list of accounts is requested.
Authorization Bearer	String	Conditional	Is contained only, if the optional OAuth Pre-Step was performed.
signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
certificate	details t.b.d.	Conditional	The certificate used for signing the request.

Attribute	Type	Condition	Description
Date	DateTime	Mandatory	Standard https header element for Date and Time

Request Filter Parameters in Path

Attribute	Type	Description
psu-involved	string	If contained, it is indicated that a PSU has directly asked this account access in realtime. The PSU then might be involved in an additional consent process, if the given consent is not any more sufficient.

Response

Attribute	Type	Condition	Description
balances	balances	Mandatory	A list of balances regarding this account, e.g. the current balance, the last booked balance.

Example

Response

```
{
  "Balances" :
      {"closed_booked" :
           {
            "Amount" : {"currency" : "EUR", "500.00"},
            "Date" : "2017-10-25"
           },
      {"expected" :
           {
            "amount" : {"currency" : "amount" : "900.00"},
            "last_action_date_time" : "2017-10-25T15:30:35.035Z"
           }
      }
}
```

6.5.3 Read Transaction List

Call

GET /v1/accounts/{account-id}/transactions {parameter-option}

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

Remark: This account-id can be a tokenized 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.

Remark: If the ASPSP is not providing the "GET Account List" call, then the ASPSP must accept e.g. the PSU IBAN as account-id in this call.

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

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 “id” attribute of the account structure. Its value is constant at least throughout the lifecycle of a

		given consent.

Request Header

Attribute	Type	Condition	Description
Process-ID	UUID	Mandatory	ID of the transaction as determined by the initiating party. In case of a once off read data request, this Process-ID equals the Process-ID of the corresponding Account Information Consent Request, cp. Section 6.3.1.
Request-ID	UUID	Mandatory	
Consent-ID	String	Conditional	Mandatory, if a consent was managed within this interface on /v1/consents for the access of this account.
PSU-ID	String	Conditional	To be used, if no OAuth Pre-Step was performed and if a list of account balances or a list of accounts is requested.
Authorization Bearer	String	Conditional	Is contained only, if the optional OAuth Pre-Step was performed.
Accept	String	Conditional	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 Further definition of content by ASPSP/ communities cp. Annex B.
signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.

Attribute	Type	Condition	Description
certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard https header element for Date and Time

Remark: The Berlin Group intends to apply for vnd-entries within the "accept" attribute for camt.05x and MT94x formats. These values will be added to this specification as soon as available.

Request Filter Parameters in Path

Attribute	Type	Condition	Description
date_from	ISODate	Mandatory	Starting date of the account statement
date_to	ISODate	Mandatory	End date of the account statement It is contained if this is a Read Account Data Request for transaction reports. .
transaction_id	string	Optional	This data attribute is indicating that the AISP is in favour to get all transactions after the transaction with identification transaction_id alternatively to the above defined period. (Implementation of a delta-report). If this data element is contained, the entries "date_from" and "date_to" might be ignored by the ASPSP if a delta report is supported.
psu-involved	Boolean	Optional	If contained, it is indicating that a PSU has directly asked this account access in real-time. The PSU then might be involved in an additional consent process, if the given consent is not any more sufficient.

Response Header

Content-Type : application/json or application/xml or application/

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 case the ASPSP returns a MT94x content, the response body consists of an MT940 or MT942 format in a text structure. 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.

A JSON response is defined as follows:

Attribute	Type	Condition	Description
_links	links	Optional	A list of hyperlinks to be recognized 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.
transactions	Account Report	Optional	JSON based account report.

Example

Request

```
GET
https://api.testbank.com/v1/accounts/qwer3456tzui7890/transactions?date_from="2017-07-01"&date_to= "2017-07-30"&psu-involved

Accept: application/json, application/text;q=0.9, application/xml;q=0.8
```

Response in JSON format

Response Code 200


```
{“transactions” :
   {“booked” :
     [
      {
       “transaction_id” : “1234567” ,
       “creditor_name” : “John Miles” ,
       “creditor_account : {“iban” : “DE43533700240123456900”},
       “Amount” : {“currency” : “EUR”, “content” : “-256,67”} ,
       “booking_date” : “2017-10-25” ,
       “value_date” : “2017-10-26” ,
       “remittance_information_unstructured”    :    “Example    for
Remittance Information”
      },
      {
       “transaction_id” : “1234568”,
       “debtor_name” : “Paul Simpson” ,
       “debtor_account” : {“iban” : “NL354543123456900”} ,
       “amount” : {“currency” : “EUR”, content: “343,01”} ,
       “booking_date” : “2017-10-25” ,
       “value_date” : “2017-10-26” ,
       “remittance_information_unstructured” : “Another example for
Remittance Information”
      }
     ],
   },
   {“pending” :
     [
      {
       “transaction_id” : “1234569” ,
       “creditor_name” : “Claude Renault” ,
       “creditor_account : {“iban” : “FR33554543123456900”},
       “amount” : {“currency” : “EUR”, “content” : “-100,03”} ,
       “value_date” : “2017-10-26” ,
       “remittance_information_unstructured” : “Third Example for
Remittance Information”
      }
     ]
   },
   {”_links”:
     {“account-link” : “/v1/accounts/3dc3d5b3-7023-4848-9853-
f5400a64e80f”}
   }
```

}

Response in case of huge data amount as a download.

```
{
  "_links : {"download" : www.test-
api.com/xs2a/v1/accounts/12345678999/transactions/download/}
}
```

7 Processes used commonly in AIS and PIS Services

Processes on PSU identification, PSU authentication 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 and the Account Information Service separately. The structure of all parameters in the request header/body and the response header/body are equal.

7.1 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 not applicable.
- 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.
- Embedded SCA Approach: The Update PSU Data Request might be used to Add credentials as a first factor authentication data of the PSU

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.

7.1.1 Update PSU Data (Identification) in the Decoupled Approach

This call is used, when in the preceding call the hyperlink of type “update_psu_identification” was contained.

Call in case of a Payment Initiation Request

PUT /v1/payments/{payment-product}/{resource-id}

Updates the payment initiation data on the server by PSU data, if requested by the ASPSP

Call in case of an Account Information Consent Request

PUT /v1/consents/{resource-id}

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

Path

Attribute	Type	Description
payment -product	string	
resource-id	string	

Request Header

Attribute	Type	Condition	Description
Request-ID	UUID	Mandatory	
Process-ID	UUID	Mandatory	
PSU-ID	String	Conditional	Contained if not yet contained in the first request, and mandated by the ASPSP in the related response
PSU-Corporate-ID	String	Conditional	Contained if not yet contained in the first request, and mandated by the ASPSP in the related response. This field is relevant only in a corporate context.
Authorization Bearer	String	Conditional	Is contained only, if the optional Oauth Pre-Step was performed.
Signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
Certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard http date and time tag.

Request Body

No Body.

Response Body

Attribute	Type	Condition	Description
transaction_status		Mandatory	
psu_message		Optional	

Example

Request

```
PUT https://api.testbank.com/v1/payments/sepa-credit-transfers/qwer3456tzui7890

Request-ID      asdfoeljkasdfoelkjasdf-1234ß089

Process-ID      3dc3d5b3-7023-4848-9853-f5400a64e80f

PSU-ID          PSU-1234
```

Response

```
Response Code 201
```

Response Body

```
{

 "transaction_status" : "AcceptedTechnicalValidation",

 "psu_message"  :  "Please  use  your  BankApp  for  transaction 72uthorization."

}
```

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

This call is used, when in the preceding call the hyperlink of type "update_psu_authentication" was contained.

Call in case of a Payment Initiation

PUT /v1/payments/{payment-product}/{payment-id}

Updates the payment initiation data on the server by PSU data, if requested by the ASPSP

Call in case of an Account Information Consent Request

PUT /v1/consents/{consent-id}

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

Path

Attribute	Type	Description
payment -product	string	
resource-id	string	

Request Header

Attribute	Type	Condition	Description
Request-ID	UUID	Mandatory	
Process-ID	UUID	Mandatory	
PSU-ID	String	Conditional	Contained if not yet contained in the first request, and mandated by the ASPSP in the related response
Authorization Bearer	String	Conditional	Is contained only, if the optional Oauth Pre-Step was performed.
Signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
Certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard http date and time tag.

Request Body

Attribute	Type	Condition	Description
psu_data.password	string	Conditional	

Response Body

Attribute	Type	Condition	Description
chosen_sca _method	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.
sca_challenge _data	challenge	Conditional	Challenge data might be contained, if only one authentication method is available.
Sca_methods	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 recognized by the TPP. Might be contained, if several authentication methods are available for the PSU. Type of links admitted in this response:
			“select_authentication_method” : 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 “sca_methods” contained in the response

Attribute	Type	Condition	Description
			body
			“authorise_transaction” : The link to the resource, where the “Transaction Authorization Request” is sent to. This is the link to the resource which will authorize the transaction by checking the SCA authentication data within the Embedded SCA approach.
			“self” : The link to the resource itself.
			“status”: The link where the transaction status of the resource can be retrieved.
Transaction_status		Mandatory	
psu_message		Optional	

Example

Request in case of Embedded Approach

```
PUT                 https://api.testbank.com/v1/payments/sepa-credit-transfers/qwer3456tzui7890

Request-ID      asdfoeljkasdfoelkjasdf-1234ß091

Process-ID      3dc3d5b3-7023-4848-9853-f5400a64e80f

PSU-ID          PSU-1234

-d {

“psu_data” {

  “password” : “start12”

  }

}
```

Response in case of the embedded approach

Response Code 200

```
{

  "transaction_status" : "AcceptedTechnicalValidation",

  _links{

     "authorise_transaction"       :       "/v1/payments/sepa-credit-transfers/1234-wertiq-983"

   }

}
```

7.1.3 Update PSU Data (Authentication Method) in the Embedded Approach

This call is used, when in the preceding call the hyperlink of type "select_authentication_method" was contained.

Call

PUT /v1/payments/{payment-product}/{resource-id}

Updates the payment initiation data on the server by PSU data, if requested by the ASPSP

Call in case of an Account Information Consent Request

PUT /v1/consents/{resource-id}

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

Path

Attribute	Type	Description
payment -product	string	
resource-id	string	

Request Header

Attribute	Type	Condition	Description

Attribute	Type	Condition	Description
Request-ID	UUID	Mandatory	
Process-ID	UUID	Mandatory	
Authorization Bearer	String	Conditional	Is contained only, if the optional Oauth Pre-Step was performed.
Signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
Certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard http date and time tag.

Request Body

Attribute	Type	Condition	Description
authentication _method_id	String	Mandatory	The authentication method id as provided by the ASPSP.

Response Body

Attribute	Type	Condition	Description
chosen_sca _method	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.
sca_challenge _data	challenge	Conditional	Challenge data might be contained, if only one authentication method is available.
_links	links	Conditional	A list of hyperlinks to be recognized by the TPP. Might be contained, if several authentication methods are available for the PSU.

			Type of links admitted in this response: “authorise_transaction” : The link to the resource, where the “Transaction Authorization Request” is sent to. This is the resource which will check the SCA authentication data within the Embedded SCA approach.
Transaction_status		Mandatory	
psu_message		Optional	

Example

Request in case of Embedded Approach

```
PUT https://api.testbank.com/v1/payments/sepa-credit-transfers/qwer3456tzui7890

Request-ID      asdfoeljkasdfoelkjasdf-1234ß093

Process-ID      3dc3d5b3-7023-4848-9853-f5400a64e80f

{

authentication_method_id: “myAuthenticationID”

}
```

Response in case of the embedded approach

```
{
  “transaction_status” : “AcceptedTechnicalValidation”,
  “chosen_sca_method” : {
     “authentication_type” : “SMS_OTP”,
     “authentication_method_id” : “myAuthenticationID”}
  “sca_challenge_data” : {
     “OTP_max_length” : “6”
     “OTP_format” : “integer”}
  _links{
```

```
    "authorise_transaction":"/v1/payments/sepa-credit-
transfers/1234-wertiq-983"
  }
}
```

7.2 Transaction Authorisation

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

Call in case of an Payment Initiation Request

PUT /v1/payments/{payment-product}/{resource-id}

Transfers data for SCA checks by the ASPSP.

Call in case of an Account Information Consent Request

PUT /v1/consents/{resource-id}

Transfers data for SCA checks by the ASPSP.

Path

Attribute	Type	Description
payment -product		
resource-id		

Request Header

Attribute	Type	Condition	Description
Request-ID	UUID	Mandatory	
Process-ID	UUID	Mandatory	
Authorization Bearer	String	Conditional	Is contained only, if the optional Oauth Pre-Step was performed.
Signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by

Attribute	Type	Condition	Description
			ASPSP.
Certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard http date and time tag.

Request Body

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

Response Body

Attribute	Type	Condition	Description
transaction_status			

Example

Request

```
PUT       https://api.testbank.com/v1/payments/sepa-credit-transfers/
qwer3456tzui7890

-d {

 "sca_authentication_data" : "123456"

}
```

Response in case of the embedded approach

Response Code 200

Response Body

```
{
```

```
  "transaction_status" : "AcceptedCustomerProfile"

}
```

8 Combination of AIS and PIS Services

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 “combined_service_indicator” in the Account Information Consent Request was set, cp. Section 6.3.1.

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.

9 Confirmation of Funds Service

9.1 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 https parameters, resp. are taken from e-Idas 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 HTTPS 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 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	
Transaction Identification	transaction_id			x			
Request Timestamp	DateTime		x			m	
TPP Certificate Data	certificate			x		c	
TPP Electronic Signature	signature			x		c	
Service Type		x				m	
Response Code			x				m
TPP Message Information	tpp_message			x			o
Card Number	card_number			x		c	
Account Number	psu_account			x		c	
Name Payee	payee			x		o	
Transaction Amount	amount			x		m	

9.2 Confirmation of Funds Request

Call

POST /v1/confirmation-of-funds

Creates a confirmation of funds request at the ASPSP.

Path

No specific path parameters.

Request Header

Attribute	Type	Condition	Description
Process-ID	UUID	Mandatory	ID of the transaction as determined by the initiating party.
Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization Bearer	String	Conditional	Is contained only, if the optional Oauth Pre-Step was performed.
Signature	details t.b.d.	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
Certificate	details t.b.d.	Conditional	The certificate used for signing the request.
Date	DateTime	Mandatory	Standard https header element date and time

Request Body

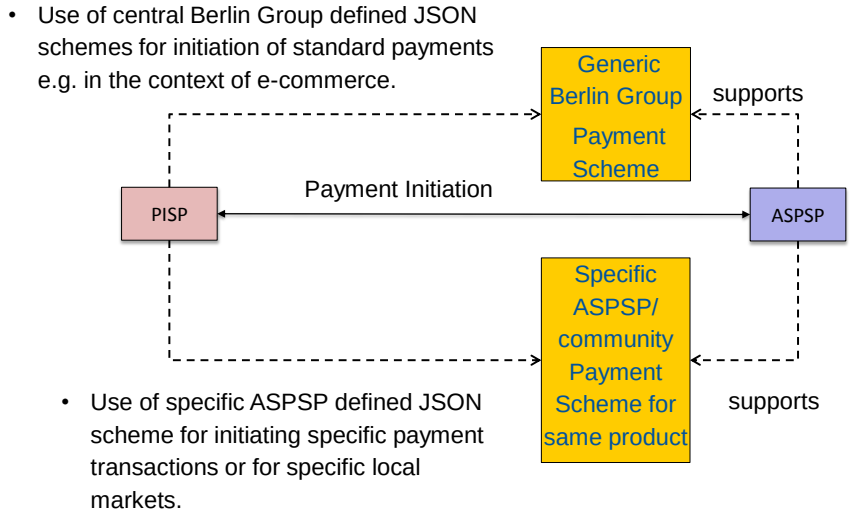
Attribute	Type	Condition	Description
card_number	string	Conditional	Card Number of the card issued by the PIISP. Must be delivered if available.
psu_account	account reference	Mandatory	PSU's account number.
payee	string	Optional	The merchant where the card is accepted as an information to the PSU.
instructed_amount	Amount	Mandatory	Transaction amount to be checked within the funds check mechanism.

10 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 Section 5.3.1.

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.



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

Data Element	Type	SCT EU Core	SCT_INST EU Core	Target2 Paym. Core	Cross Curr CT Core
end_to_end_ identification	String	Optional	Optional	Optional	n.a.
debtor_account (incl. type)	Account Reference	mandatory	mandatory	mandatory	mandatory
debtor_account _currency	CODE	n.a.	n.a.	n.a.	optional
ultimate_debtor	String	n.a.	n.a.	n.a.	n.a.
instructed_amount (inc. Curr.)	Amount	mandatory	mandatory	mandatory	mandatory

creditor_account	Account Reference	mandatory	mandatory	mandatory	mandatory
creditor_agent	BICFI	optional	optional	optional	optional
creditor name	String	mandatory	mandatory	mandatory	mandatory
creditor address	Address	optional	optional	optional	mandatory
ultimate_creditor	String	n.a.	n.a.	n.a.	n.a.
purpose_code	CODE	n.a.	n.a.	n.a.	n.a.
remittance _information _unstructured	String	optional	optional	optional	optional
remittance _information-structured	Remittance	n.a.	n.a.	n.a.	n.a.
requested_execution _time	DateTime	n.a.	n.a.	n.a.	n.a.

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.

Only fields defined in Section 11 for payment structures may be added.

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

11 Complex Data Types and Code Lists

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

***NOTE**: This is a first complete draft of the complex data types. These definitions are given as background explanation to this specification during market consultation. They are not complete and will be further detailed and extended after market consultation.*

11.1 PSU Data

Attribute	Type	Condition	Description
password	string	Optional	

11.2 TPP Message Information

Attribute	Type	Condition	Description
category	String	Mandatory	”ERROR” permitted
code	Message Code	Mandatory	
path	string	Conditional	
text	string	optional	Additional explaining text.

11.3 Amount

Attribute	Type	Condition	Description
currency	string	Mandatory	ISO 4217 code
content		Mandatory	

11.4 Creditor

Attribute	Type	Condition	Description
Name			
Address			

11.5 Reference Party

Attribute	Type	Condition	Description
Name	string		
Id	string		

11.6 Remittance

To be completed after market consultation.

Attribute	Type	Condition	Description

11.7 Links

Attribute	Type	Condition	Description
redirect	string	Optional	A link to an ASPSP site where SCA is performed within the Redirect SCA approach.
update_psu_identification	string	optional	The link to the payment initiation or account information resource, which needs to be updated by the psu identification if not delivered yet.

Attribute	Type	Condition	Description
update_psu_authentication	string	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.
select_authentication_method	string	optional	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.
authorise_ transaction	String	optional	The link to the payment initiation or consent resource, where the "Transaction Authorization”Request" is sent to. This is the link to the resource which will authorize the payment or the consent by checking the SCA authentication data within the Embedded SCA approach.
self	String	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	String	optional	
account-link	String	optional	
balances	String	optional	A link to the resource providing the balance of a dedicated account.
transactions	String	Optional	A link to the resource providing the transaction history of a dediated amount.
first_page_link	String	Optional	Navigation link for account reports.
second_page_link	String	Optional	Navigation link for account reports.
current_page_ link	String	Optional	Navigation link for account reports.

Attribute	Type	Condition	Description
last_page_link	string	optional	Navigation link for account reports.

11.8 Authentication Object

Attribute	Type	Condition	Description
authentication_type	Authentication Type	Mandatory	Type of the authentication method.
authentication _method_id	String	Mandatory	An identification provided by the ASPSP for the later identification of the authentication method selection.
name	String	Optional	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

11.9 Authentication Type

Will be detailed after market consultation.

Name	Description
SMS_OTP	
CHIP_OTP	
PHOTO_OTP	
PUSH_OTP	

11.10 Challenge

Attribute	Type	Condition	Description
image	String	Optional	PNG data (max. 512 kilobyte) to be displayed to the PSU, Base64 encoding , cp. [RFC 4648]. This attribute is used only, when PHOTO_OTP or CHIP_OTP is the selected SCA method.
OTP_max_length	integer	optional	The maximal length for the OTP to be typed in by the PSU.
OTP_format	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

11.11 Message Code

The permitted message error code will be defined in detail after market consultation.

Message Code	Description

11.12 Transaction Status

The transaction status is filled with value of the ISO20022 data table, where the entry from the name column is used.

Code	Name	ISO 20022 Definition
ACCP	AcceptedCustomerProfile	Preceding check of technical validation was successful. Customer profile check was also successful.
ACSC	AcceptedSettlementCompleted	Settlement on th' 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.

If the response is XML based, then the Code entry is used, as required by the pain.002 schema.

If the response is JSON based, then the Name entry is used, to get a better readability.

11.13 Single Account Access

Attribute	Type	Condition	Description
Account-	account-reference	Mandatory	
access type	array of string	Mandatory	The“values ”balan“e" and "tran”actions" are permitted.

11.14 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.

Attribute	Type	Condition	Description
iban	string	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	string	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.
pan	string	optional	Primary Account Number (PAN) of a card, can be tokenized by the ASPSP due to PCI DSS requirements. 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.
msisdn	string	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.

11.15 Account

Attribute	Type	Condition	Description
id	string	mandatory	This is the data element to be used in the path when retrieving data from a dedicated account, cp. Section 6.5.2 or Section 6.5.3 Below.
iban	string	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	string	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.
pan	string	optional	Primary Account Number (PAN) of a card, can be tokenized by the ASPSP due to PCI DSS requirements. 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.
msisdn	string	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.
name	string	optional	Name given by the bank or the PSU in Online-Banking
account_type	string	optional	Product Name of the Bank for this account
BIC	string	optional	The BIC associated to the account.
balances	Balances	conditional	
currency	Currency type	Mandatory	
_links	links	optional	Links to the account, which can be directly used for retrieving account information from

Attribute	Type	Condition	Description
			this dedicated account. Links to “balances” and/or “transactions”

11.16 Balances

Attribute	Type	Condition	Description
booked	single balance	Optional	Last known book balance of the account.
expected	single balance	Optional	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.
authorised	single balance	Optional	The expected balance together with the value of a pre-approved credit line the ASPSP makes permanently available to the user.
opening_booked	single balance	Optional	Book balance of the account at the beginning of the account reporting period. It always equals the closing book balance from the previous report.
closing_booked	single balance	Optional	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.
interim_available	single balance	Optional	Available balance calculated in the course of the account ’ervicer'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.

11.17 Single Balance

Attribute	Type	Condition	Description
amount	Amount	Mandatory	
last_action_date_time	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.
date	ISODate	Optional	

11.18 Account Report

Attribute	Type	Condition	Description
booked	array of transactions	mandatory	
pending	array of transactions	optional	
_links	links	Mandatory	The following links might be used within this context:
			account link (mandatory)
			first_page_link (optional)
			second_page_link (optional)
			current_page_ link (optional)
			last_page_link (optional)

11.19 Transactions

Attribute	Type	Condition	Description

Attribute	Type	Condition	Description
transaction_id	string	optional	Can be used as access-id in the API, where more details on an transaction is offered.
entry_date	ISODate	optional	
amount	amount	mandatory	
credit_debit	string	mandatory	“Credited” or “Debited” as permitted values.
creditor	string	optional	Name of the creditor if a “Debited” transaction
creditor _account	account	conditional	
ultimate _creditor	string	optional	
debtor	string	optional	Name of the debtor if a “Crebited” transaction
debtor _account	account	conditional	
ultimate _debtor	string	optional	
remittance_ information	string	optional	

Remark: This list will be extended by further optional subfields after market consultation.

11.20 DateTime

This interface requires the HTTP1 Date format which is supporting the fixed length format of RFC1123.

11.21 HTTP Response Codes

Status Code	Description

200	PUT, GET Response Codes This return code is permitted when the request is repeated due to a time-out. The response in that might be either a 200 or 201 code depending on the ASPSP implementation.
201 (Created)	POST Response code where Payment Initiation, Consent Request or Funds Request was correctly performed.
204	DELETE Response code where a consent resource was successfully deleted
400 (Bad Request)	Validation error occurred.
401 (Unauthorized)	
403 (Forbidden)	TPP role as defined in the certificate is not matching a payment initiation.
422	

12 References

[EBA-RTS] Final Report: Draft Regulatory Technical Standards on Strong Customer Authentication and common and secure communication under Article 98 of Directive 2015/2366 (PSD2), 23.02.2017

[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

[PSD2] Directive (EU) 2015/2366 of the European Parliament and of the Council on payment services in the internal market, 25.11.2016

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

[XS2A-OR] The Berlin Group – XS2A Interface Interoperability Framework – Operational Rules, Version 0.99, 02 Oct. 2017

13 Appendix A: Additional Payment Products

Remark: Additional payment products might in future also be published on the Berlin Group website to be able to react in a shorter time.

13.1 JSON based Payment Products

Norway

Data Element	Type	Norway CT non-Euro
end_to_end_ identification	String	n.a.
debtor_account (incl. type)	Account	mandatory
debtor_account _currency	CODE	optional
ultimate_debtor	String	optional
instructed_amount (inc. Curr.)	Amount	mandatory
creditor_account	Account	mandatory
creditor_agent	BICFI	optional
creditor name	String	mandatory
creditor address	Address	mandatory
ultimate_creditor	String	optional
purpose_code	CODE	optional
remittance _information _unstructured	String	optional
remittance _information-structured	Remittance	optional
requested_execution _time	DateTime	optional
remittance_information_structured. referred_document_Information		optional
remittance_information_structured. referred_document_information. creditor_reference_nformation	string	optional

Sweden

Data Element	Type	Swedish Dom CT SEK	Swedish Dom CT SEK Fast	Swedish Dom Giro Paym	Cross Curr CT Core
end_to_end_ identification	String	n.a.	n.a.	n.a.	optional

Data Element	Type	Swedish Dom CT SEK	Swedish Dom CT SEK Fast	Swedish Dom Giro Paym	Cross Curr CT Core
debtor_account (incl. type)	Account	mandatory	mandatory	mandatory	mandatory
debtor_account _currency	CODE	optional	optional	optional	mandatory
ultimate_debtor	String	n.a.	n.a.	n.a.	n.a.
instructed_amount (inc. Curr.)	Amount	mandatory	mandatory	mandatory	mandatory
creditor_account (incl. type)	Account	mandatory	mandatory	mandatory	mandatory
creditor_agent	BICFI	optional	optional	optional	optional
creditor name	String	n.a.	n.a.	n.a.	optional
creditor address	Address	n.a.	n.a.	n.a.	Mandatory (country code)
ultimate_creditor	String	n.a.	n.a.	n.a.	n.a.
purpose_code	CODE	n.a.	n.a.	n.a.	n.a.
category_purpose		n.a.	optional	n.a.	mandatory
service_level		n.a.	mandatory	n.a.	conditional
chargebearer				n.a.	mandatory
remittance _information _unstructured	String	optional	optional	optional	optional
remittance _information-structured	Remittance	n.a.	n.a.	optional	n.a.
requested_execution _time	DateTime	n.a.	n.a.	n.a.	n.a.
requested_execution _date	ISODate	optional	optional	optional	optional

13.2 XML based Payment Products

Links to community based specifications based on pain.001 formats will be added in the final version of this specification for the communities implementing the Berlin Group standard.

14 Appendix B: Transaction Report Formats

Links to community based specifications based on pain.001 formats will be added in the final version of this specification for the communities implementing the Berlin Group standard.