

THE *Berlin* GROUP

A EUROPEAN STANDARDS INITIATIVE



openFinance API Framework

Implementation Guidelines

Consent API for V2.x

Version 2.2

31 October 2025

License Notice

This Specification has been prepared by the Participants of the openFinance Taskforce*. This Specification is published by the Berlin Group under the following license conditions:

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



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

- Implementation of certain elements of this Specification may require licenses under third party intellectual property rights, including without limitation, patent rights. The Berlin Group or any contributor to the Specification is not, and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.
- Any right, title and interest in and to the copyright and all related rights in topic-related Scheme Rulebooks, belong to the respective Scheme Manager (amongst others, the European Payments Council AISBL - EPC).
- The Specification, including technical data, may be subject to export or import regulations in different countries. Any user of the Specification agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import (parts of) the Specification.

* The openFinance Taskforce brings together participants of the Berlin Group with additional European banks (ASPSPs), banking associations, payment associations, payment schemes and interbank processors.

Contents

1	Introduction.....	1
1.2.1	Implicit and explicit consent.....	2
1.2.2	Consent Categories for explicit consents	3
1.2.3	Consent Token.....	3
2	Application Layer: Guiding Principles.....	5
2.1	API Access Methods	5
2.2	Multi-currency Account Specifics for Consents	8
2.2.1	Submission of Consents on Account Access.....	8
2.2.2	Submission of Consents on Confirmation of Funds	8
2.3	Status Information within the Establish Consent Process and Lifecycle	8
3	Establish Consents Messages.....	10
3.1	Generic Consent Flow	10
3.1.1	Redirect SCA Approach: Implicit Start of the Authorisation Process.....	10
3.1.2	OAuth2 SCA Approach: Implicit Start of the Authorisation Process.....	10
3.1.3	Decoupled SCA Approach: Explicit Start of the Authorisation Process.....	11
3.1.4	Embedded SCA Approach with only one SCA method available.....	12
3.1.5	Embedded SCA Approach with Selection of a SCA method	13
3.1.6	Multilevel SCA Approach: Example Decoupled SCA Approach	14
3.2	Data Transport Overview.....	16
3.3	Multicurrency Accounts in Submission of Consents.....	19
3.4	Establish Consent on Account Information	19
3.4.1	Consent Request for Account Information.....	19
3.4.2	Get Consent Request.....	31
3.5	Establish Consent on Funds Confirmation.....	34
3.5.1	Consent on Funds Confirmation Request.....	34
3.5.2	Get Consent Request.....	42



3.6	Establish Consent on User Parameters Access.....	45
3.6.1	Consent on User Parameters Access Request	45
3.6.2	Get Consent Request.....	53
3.7	Establish Consent on Document Services	54
3.7.1	Establish Consent on Document Services Request.....	55
3.7.2	Get Consent Request.....	62
3.8	Get Consent Status Request	65
3.9	Cancellation of a Consent.....	66
4	References	68
4.1	Documents of the NextGenPSD2 API Framework.....	68
4.2	Documents of the openFinance API Framework.....	68
4.3	Further documents	69
5	Annex A: Change Log	71
5.1	Changes from Version 2.0 to Version 2.1	71
5.2	Changes from Version 2.1 to Version 2.2	72



1 Introduction

1.1 From Core XS2A Interface to openFinance API

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

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

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

This XS2A Framework is now planned to be extended to extended services. This interface is addressed in the following as **openFinance API**. This openFinance API differs from the XS2A interface in several dimensions:

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

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

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

The following account access methods are covered by this framework:

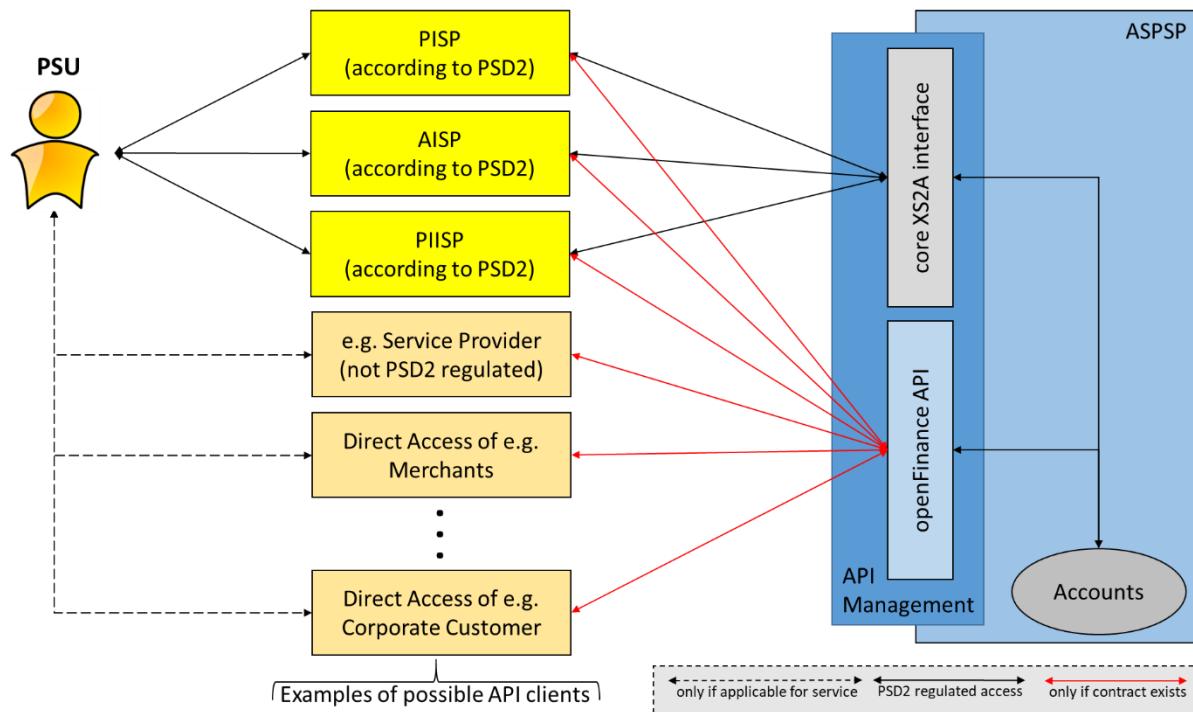


Figure 1: Core XS2A interface and openFinance API

The ASPSP may restrict the access to the services offered at its openFinance API and require dedicated onboarding. The requirements for the rights to access to services offered at the openFinance API are out of scope of this document. These requirements are specified in detail in [oFA-IG-ADM].

In contrast to the services of the openFinance API the ASPSP has to offer access to the services of his core XS2A API to any TPP without any discrimination as long as the TPP has got the necessary licence to access a service as a PISP, AISP or PIISP from an NCA according to the regulations of [PSD2].

1.2 Consent API

XS2A and the openFinance API have to deal with PSU consent for very different scenarios since the API client can access sensitive PSU data via the API. The legal text [PSD2] and [EBA-RTS] introduces the notion of consent for AIS, PIS as well as PIIS. In addition, operational rules of premium services might add requirements on PSU consent for services introduced at all times.

1.2.1 Implicit and explicit consent

PSU consent for some service like payments are worked out implicitly when authorising the payment as such, since the submission of the related data (e.g. payment data) and the related

authorisation is worked out at the same time in one session. So, no specific consent object or related token for later access is needed in that case.

Other services are divided more explicitly in a phase, where the related consent is granted and a separate phase where the related consent is used to access certain endpoints in the openFinance API e.g. for retrieving data. To support such a scenario technically, a dedicated consent API is defined in such cases.

1.2.2 Consent Categories for explicit consents

Consents in different contexts might be consents between PSU and TPP, consent between PSU and ASPSP or both. The most prominent example is the consent to share account information data with an AISPs, where the consent is agreed between TPP and PSU and where the SCA of the PSU is mandated to be used to secure this consent following PSD2. In other examples within the premium openFinance API like the PSU consent for PIISP via the API, the consent is directly between ASPSP and PSU to fulfil the requirements in [EBA-RTS], but technically provided via the TPP. These two different views on consent are still technically managed via the same consent API defined within this document. Note, that the consent API will offer different path parameters to support consents on different consent categories. The consent categories supported in this document are listed in Section 1.3

1.2.3 Consent Token

At the end of a successful authorisation process within the consent API, the API client uses the consent resource identification `consentId` as a token towards the openFinance APIs addressed by the consent categories and the PSU during the consent process. Please note that the API Client needs to store the token in its system for the whole lifetime of the token. Please note that depending of the authorisation process, the ASPSP might in addition offer a bearer token following the OAuth2 standard, which then needs to be used by the API Client in addition when accessing the related APIs. In case the API client is addressing the related API without the related token, then the ASPSP will reject the API calls.

NOTE: This API specification relies strongly on generic API functionality of the openFinance API Framework as defined in [oFA-PFSM]. Further, complex data type definitions and code lists from [oFA DaD] are used within this specification without further reference.

1.3 Document Structure

The document defines the API access methods of the consent API and some application layer fundamentals on consent status and multicurrency accounts in Section 2. Detailed API specifications for the consent API then follow in Section 3 for the consent categories

- account-access,
- funds-confirmations,
- user-parameters-access and
- document-services.

NOTE: The embedded SCA Approach where all PSU credentials are transported via the API is very specific. For this reason, all attributes and processes to be applied for the embedded SCA approach only will be marked with turquoise throughout this document.

NOTE: The openFinance API Framework is still constantly growing, by adding more needs to grant consent to specific API functions. Thus, further consent categories and access rights will be defined in a later version of the document.

1.4 Document History

Version	Change/Note	Approved
Draft V2	Final Draft of consent API for version 2.x	15 July 2023
V2.0	First released version of the V2 consent API	05 October 2023
V2.1	Adaptation to Release V2.1. Card reconciliation accounts now need to be addressed explicitly. Details are contained in Annex A: Change Log.	31 July 2024
V2.2	Adaption to Release V2.2 Including the IBAN Check Service in 3.6.1 Details are contained in Section 5.2	31 October 2025

2 Application Layer: Guiding Principles

2.1 API Access Methods

In the following, the API access methods of the openFinance consent API are presented in an overview. The following categories of consent are supported within {consent-category}:

- account-access: consent for AIS access to addressed accounts (Service Type CONS-AIS for the XS2A interface, service type CONS-XAIS in case of access to any account type not necessarily covered by PSD2).
- funds-confirmations: access consent for the funds-confirmations service for a dedicated account (Service Type CONS-FDC).
- user-parameters-access: consent for user data access for addressed accounts (Service Type CONS-USP).
- document-services: explicit consent to submit e.g. RTPs, documents as such or in addition retrieve PSU identification related data (service type CONS-DOC) within an onboarding of the PSU to the service on the API Client side. A document might be a pure JSON RTP or a pdf or a certain document type (eInvoice, contract etc.) or a combination.

Each consent type is supporting a slightly different data model, which is why they are listed separately.

Endpoints/Resources	Method	Condition	Description
consents/account-access	POST	Mandatory	<p>Create a consent resource, defining access rights to dedicated accounts of a given PSU-ID. These accounts are addressed explicitly in the method as parameters as a core function. (Detailed consent model)</p> <p>Section 3.4.1</p>
consents/account-access	POST	Optional ²	<p>As an option, an ASPSP might optionally accept a specific access right</p> <ul style="list-style-type: none"> • for all AIS services for all available accounts of given account categories <p>(Global consent model)</p>

² ASPSP might choose between the additional consent models to achieve the "no manual IBAN entry" requirement by EBA.

Endpoints/Resources	Method	Condition	Description
			<p>As another option an ASPSP might optionally also accept a command, where only requested access rights are inserted without mentioning the addressed account(s). The actual rights and the relation to accounts is then handled afterwards between PSU and ASPSP. This option is not supported for the Embedded SCA Approach.</p> <p>(ASPSP managed consent model)</p> <p>As a last option, an ASPSP might in addition accept a command with access rights</p> <ul style="list-style-type: none"> • to see the list of available accounts (with or without balance) for given account categories <p>(Account list model)</p> <p>Section 3.4.1</p>
consents/account-access/{consentId}	GET	Mandatory	<p>Reads the exact definition of the given consent resource {consentId} including the validity status.</p> <p>Section 3.4.2</p>
consents/funds-confirmations	POST	Optional	<p>Create a consent resource to register a TPP for the confirmation of funds service for a given account by a PSU for a given PSU ID.</p> <p>See Section 3.5.1</p>
consents/funds-confirmations/{consentId}	GET	Mandatory	<p>Reads the exact definition of the given consent resource {consentId} including the validity status</p> <p>Section 3.5.2</p>
consents/user-parameters-access	POST	Optional	<p>Create a consent resource to grant an API Client access to technical user parameters related to given accounts by a PSU, related to a given PSU respectively. Technical user parameters</p>

Endpoints/Resources	Method	Condition	Description
			<p>are parameters about e.g. global or service specific limits or related signing rights.</p> <p>See Section 3.6.1</p>
consents/user-parameters-access/{consentId}	GET	Mandatory	<p>Reads the exact definition of the given consent resource {consentId} including the validity status</p> <p>Section 3.6.2</p>
consents/document-services	POST	Optional	<p>Create a consent resource to register an Originator-RTPSP or the pair (Originator-RTPSP, RTP-Originator) for a (given) account by a PSU defined by a given PSU ID and to retrieve potentially PSU related identification data during this process. The RTP can be with or without elInvoice and is interpreted as a "document" as such.</p> <p>RFU: This consent will also address postal document services in future.</p> <p>See Section 3.7.1</p>
consents/document-services/{consentId}	GET	Mandatory	<p>Reads the exact definition of the given consent resource {consentId} including the validity status</p> <p>Section 3.7.2</p>
consents/{consent-category}/{consentId}	DELETE	Mandatory	<p>Terminate the addressed consent.</p> <p>Section 3.9</p>
consents/{consent-category}/{consentId}/status	GET	Mandatory	<p>Read the consent status of the addressed consent resource.</p> <p>Section 3.8</p>

2.2 Multi-currency Account Specifics for Consents

2.2.1 Submission of Consents on Account Access

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

A sub-account with a dedicated currency can still be addressed in a dedicated way by adding the currency code to the account reference. In that case, no consent is provided to access the other sub accounts.

2.2.2 Submission of Consents on Confirmation of Funds

The consent data model provides an account, where the consent for a Confirmation of Funds Request is granted on. The account currency is an optional sub field in the account reference.

In case of multi-currency accounts this implies that the default sub-account is addressed if no currency is submitted in the Establish Confirmation of Funds Request by the TPP. The default sub-account is set by the ASPSP.

2.3 Status Information within the Establish Consent Process and Lifecycle

The status of the consent resource is changing during the initiation process and life cycle. In difference to e.g. a payment initiation process, there are only SCA checks on the consent resource and of course account permissions but no feedback loop with the ASPSP backend e.g. regarding limits or funds. For this reason, the status lifecycle for consents is less detailed than e.g. for payments. The data element for the status of the consent is defined as "consentStatus".

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

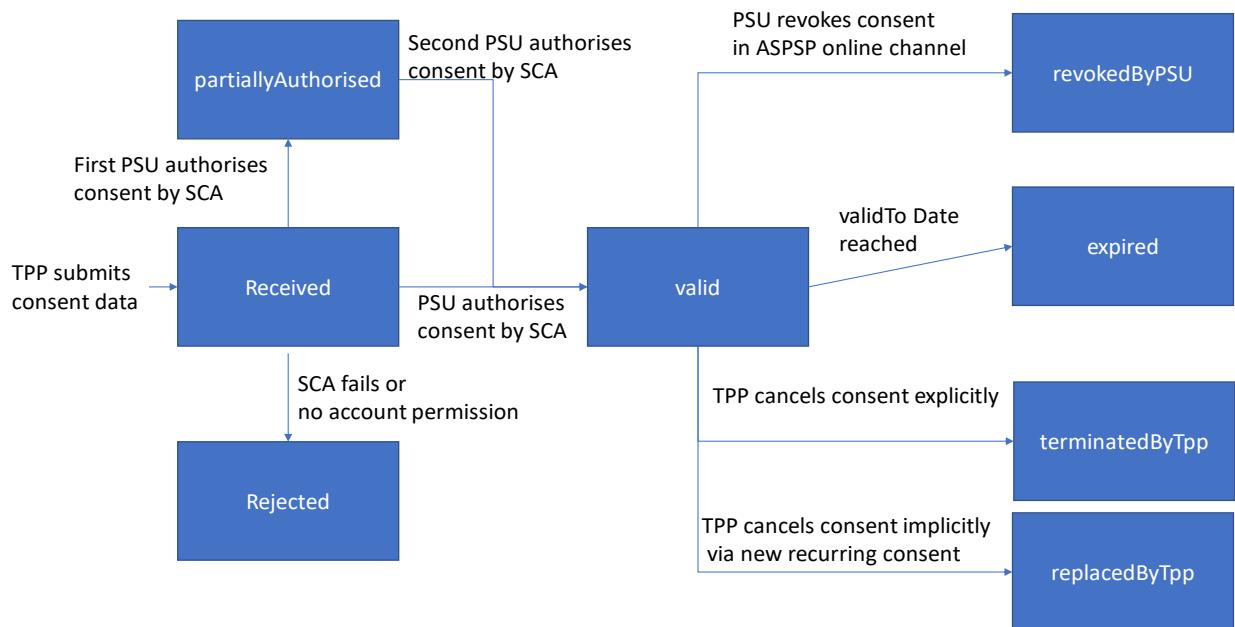
After a successful authorisation of a consent by a PSU, the consent resource might change its status during its lifecycle which needs to be transparent to the AIS. The following codes are supported during the **lifecycle phase** of the consent following the "valid" status:

- "expired": The consent has been expired (e.g. after the period of time which is mandated by an applicable regulation or after the period time requested by the TPP).
- "revokedByPsu": The consent has been revoked by the PSU.
- "terminatedByTpp": The AIS has terminated the consent.
- "replacedByTPP": The AIS has terminated the consent implicitly by submitting a new (recurrent) consent for the same PSU/Corporate.

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

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

The following diagram shows the overall status transition diagram for a consent resource:



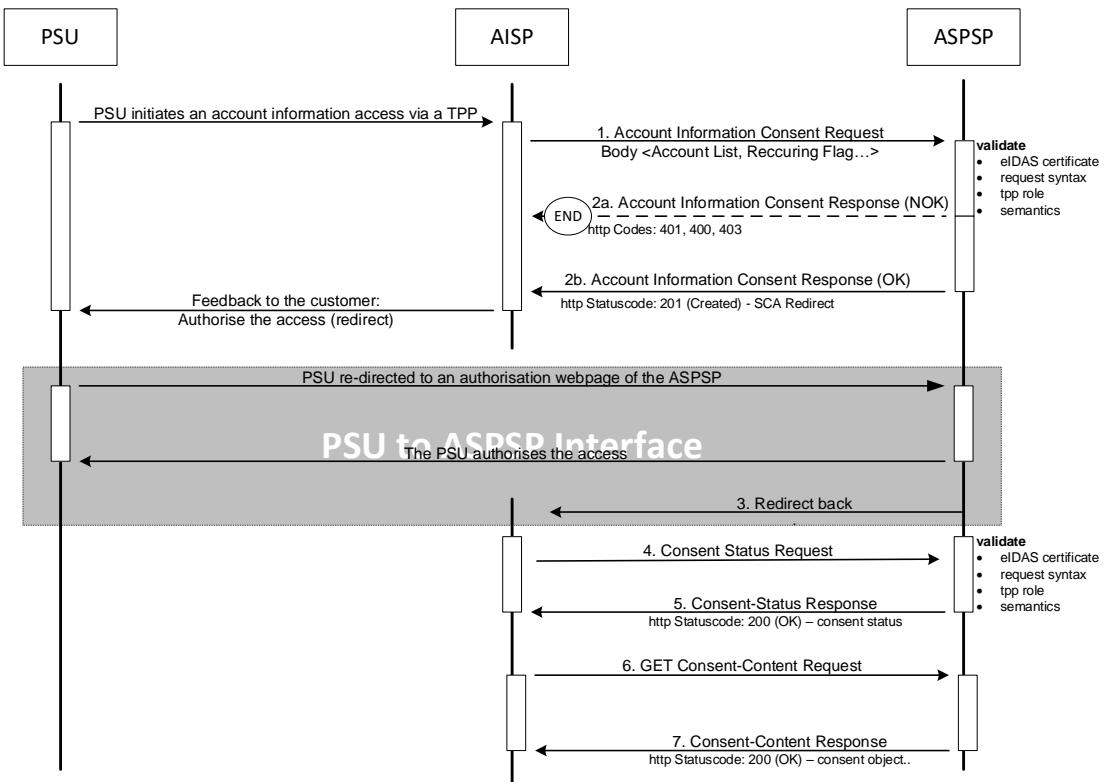
In addition, the ASPSP informs the TPP about the status of the technical SCA process for establishing a consent within the GET SCA Status Response Message. For this status reporting the data element "scaStatus" will be used.

3 Establish Consents Messages

3.1 Generic Consent Flow

3.1.1 Redirect SCA Approach: Implicit Start of the Authorisation Process

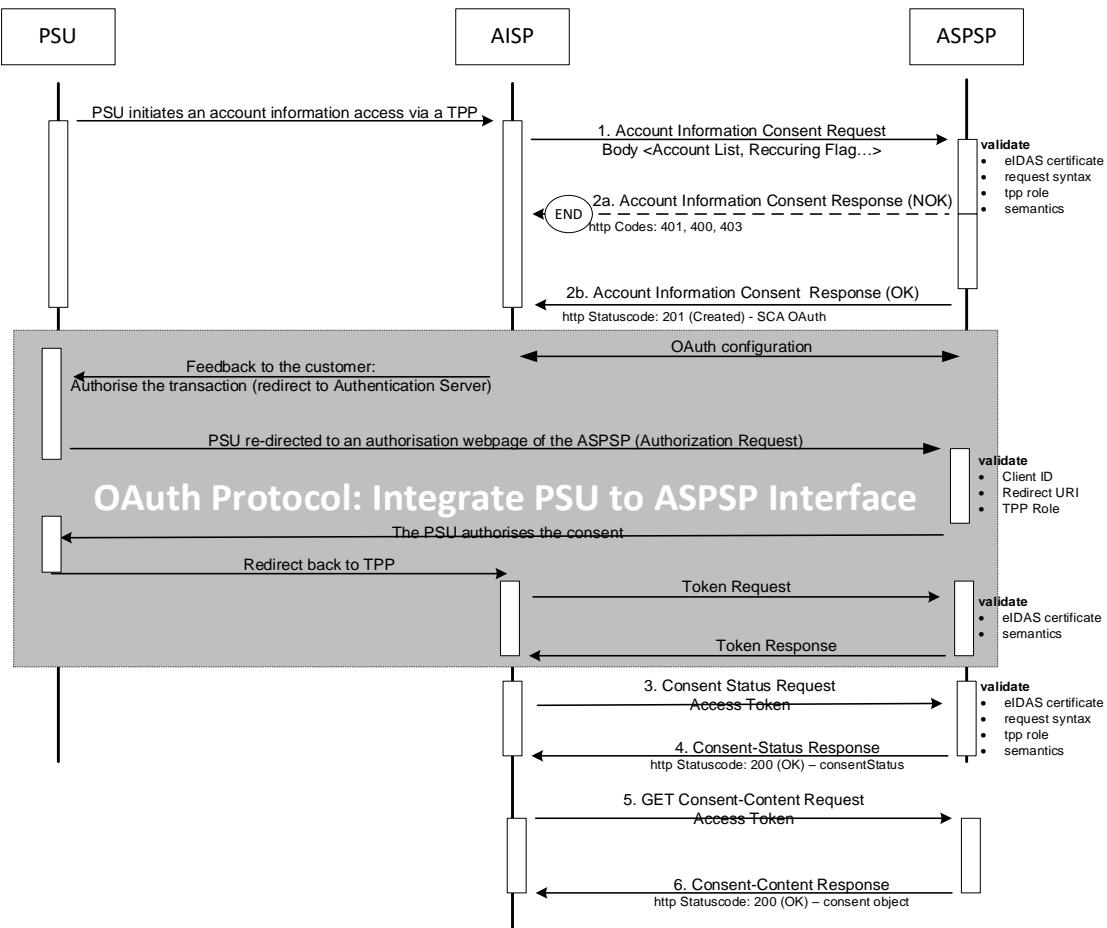
If the ASPSP supports the Redirect SCA Approach, the message flow within the Establishing Consent sub-service is simple. The Service Consent Request is followed by a redirection to the ASPSP SCA authorisation site. A status or content request on the created consent resource might be requested by the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



3.1.2 OAuth2 SCA Approach: Implicit Start of the Authorisation Process

If the ASPSP supports the OAuth2 SCA Approach, the flow is very similar to the Redirect SCA Approach. Instead of redirecting the PSU directly to an authentication server, the OAuth2

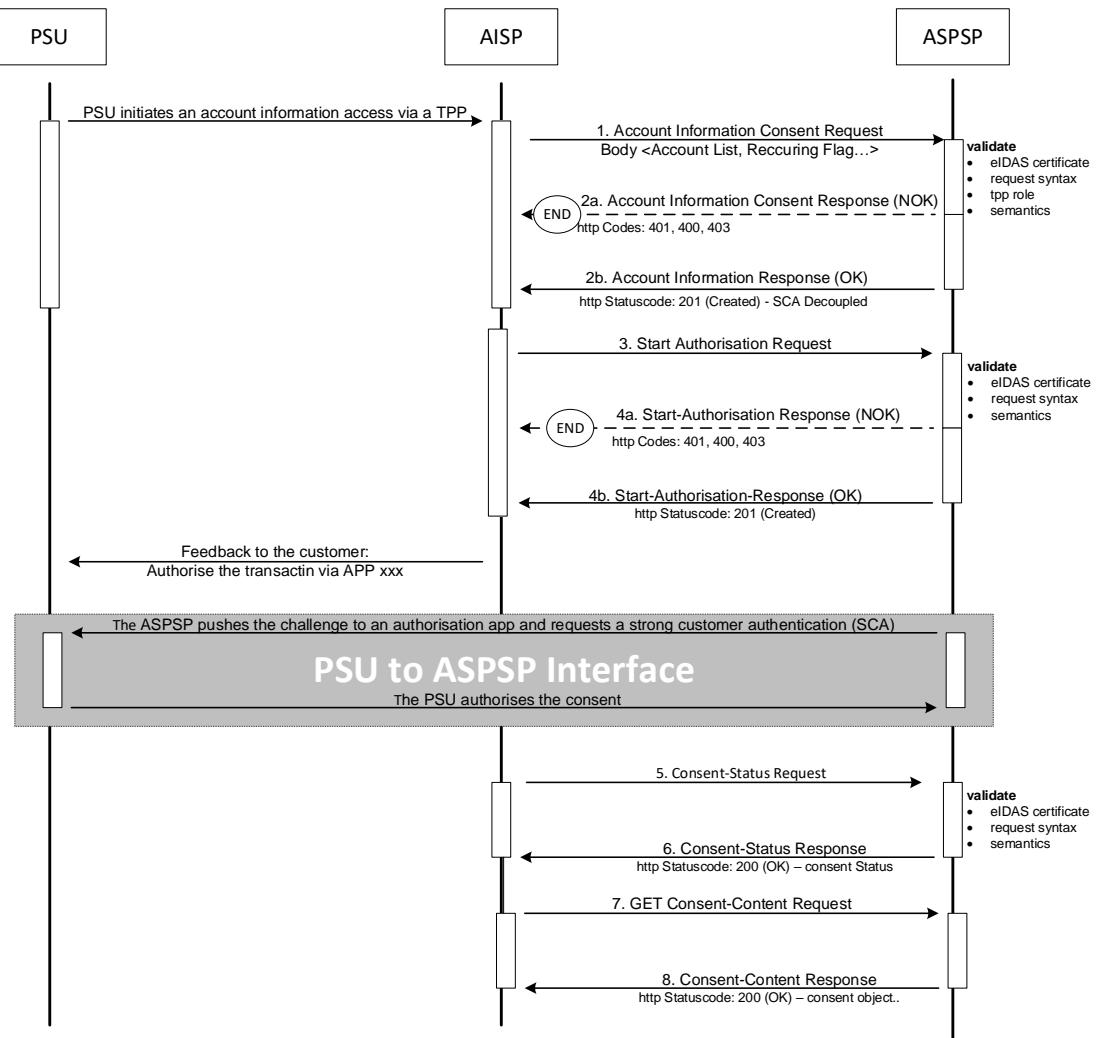
protocol is used for the transaction authorisation process. In the following, a flow is shown, where the Authorisation Process in the NextGenPSD2 API has been implicitly started.



3.1.3 Decoupled SCA Approach: Explicit Start of the Authorisation Process

The transaction flow in the Decoupled SCA Approach is similar to the Redirect SCA Approach. The difference is that the ASPSP is asking the PSU to authorise the account access consent e.g. via a dedicated mobile app. The ASPSP is asking the TPP to inform the PSU about this authentication by sending a corresponding PSU Message like "Please use your xxx App to authorise the account access".

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



NOTE: A similar process is resulting if the ASPSP implicitly starts the authorisation.

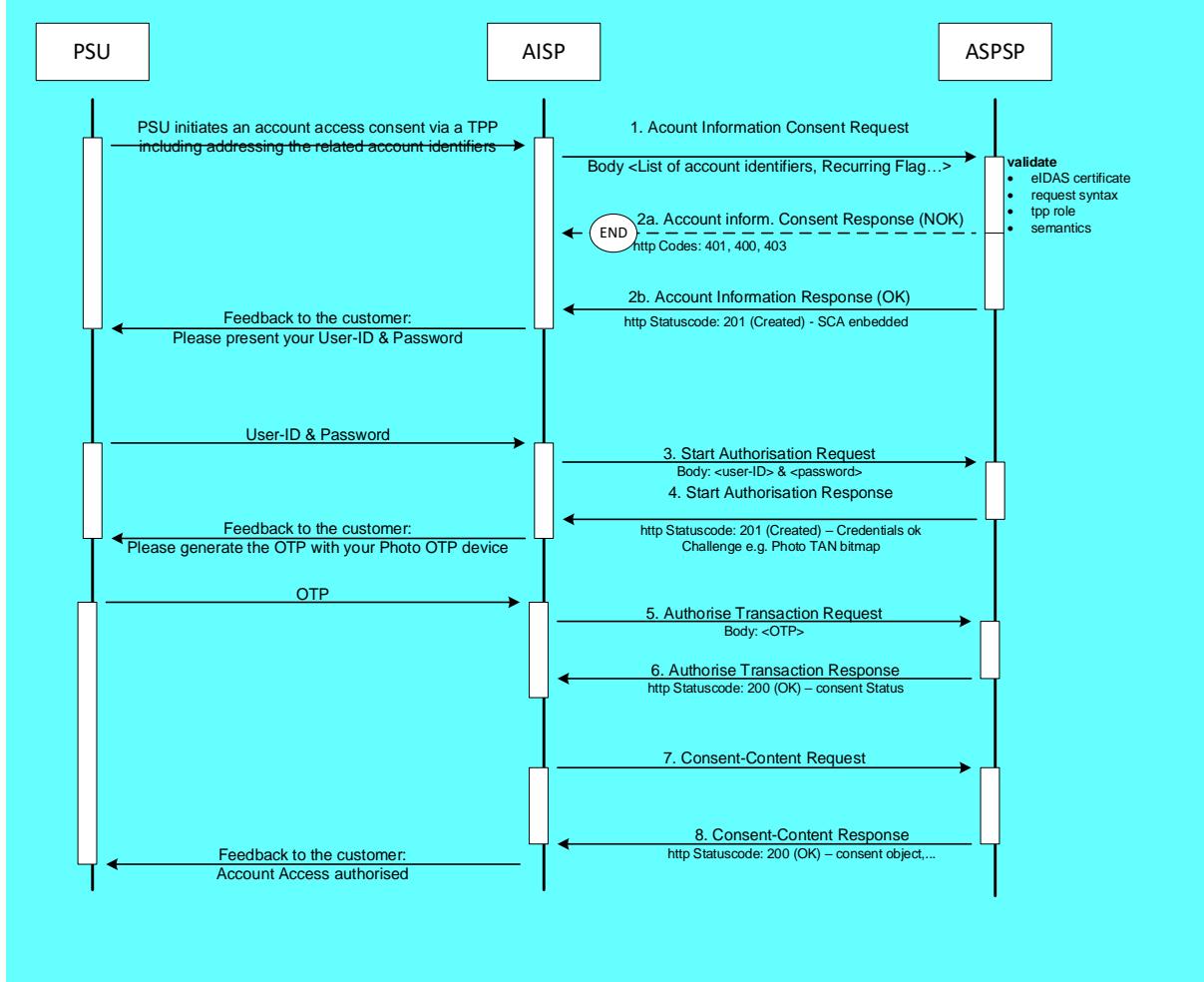
3.1.4 Embedded SCA Approach with only one SCA method available

In the following, several exemplary flows are shown, where the ASPSP has chosen to process the SCA methods for the consent approval through the PISP – ASPSP interface. In any case, the PSU normally will need to authenticate himself with a first factor, before any account or SCA method details will be available to the PISP.

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

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

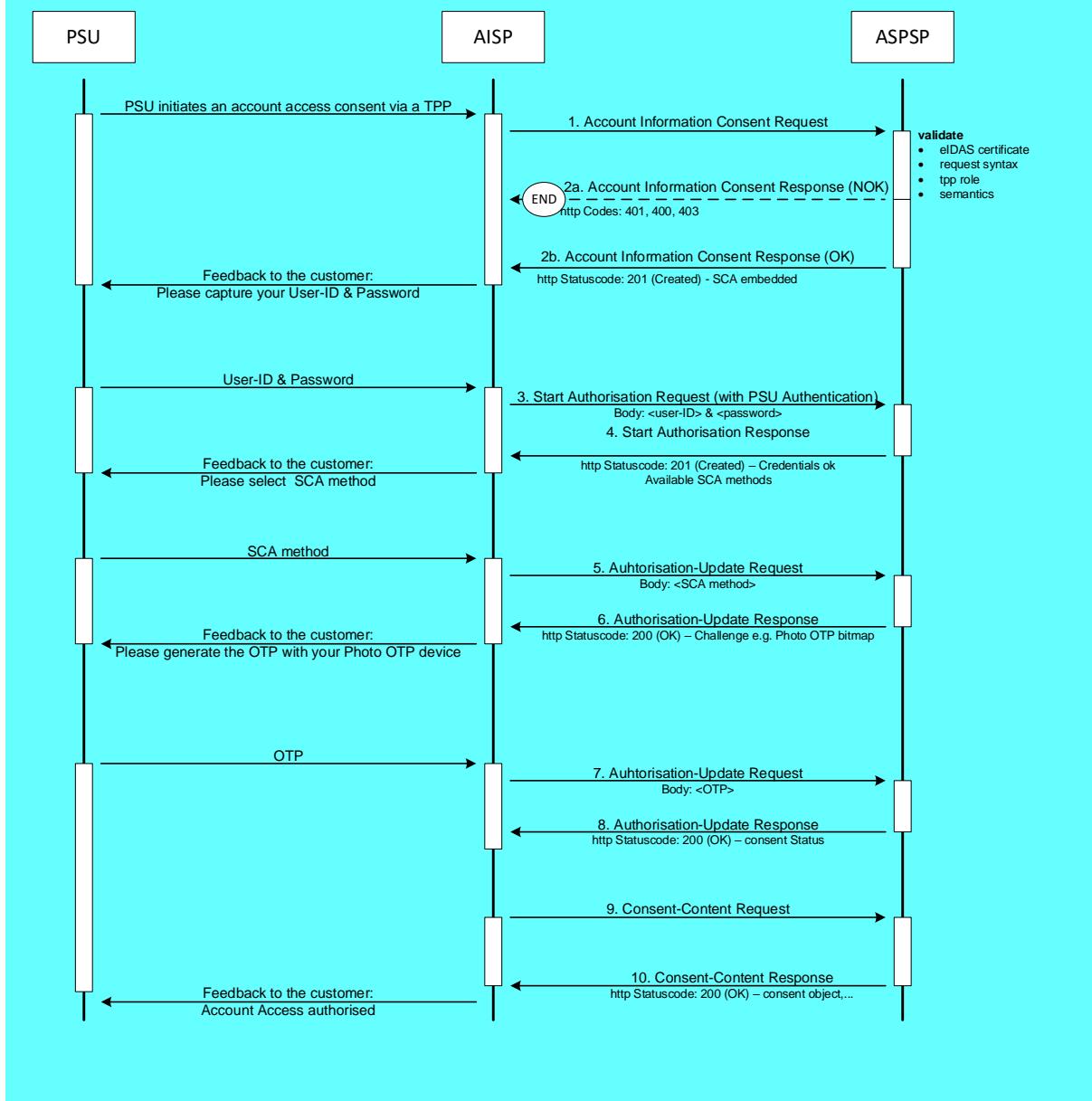
In case where only one SCA method is available, the "Authorise Transaction Request" is added to the flow, where the TPP is transmitting the authentication data of the customer, e.g. an OTP with included dynamic linking to the transaction details.



3.1.5 Embedded SCA Approach with Selection of a SCA method

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

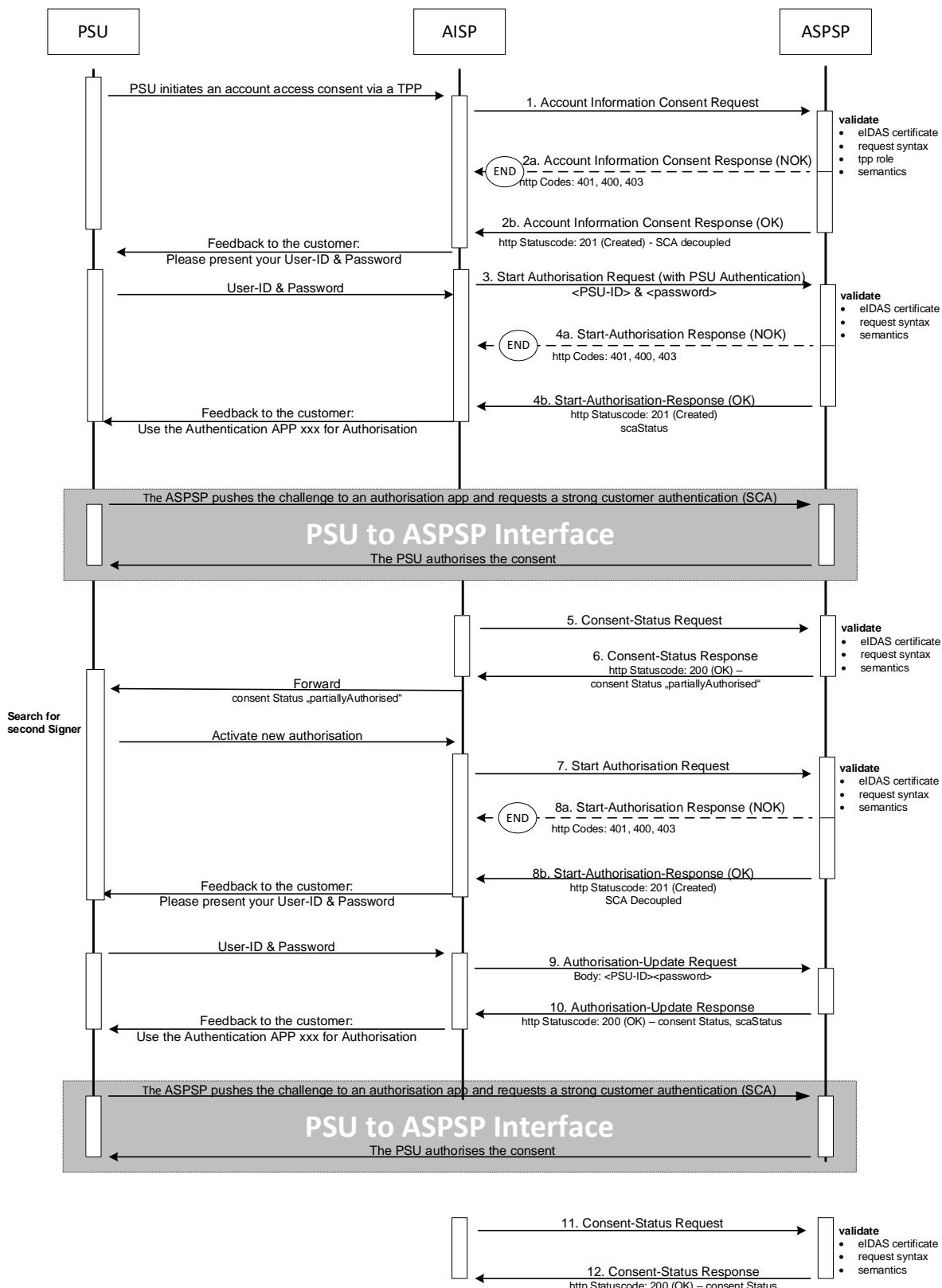
available methods to the PISP. The PISP might filter them, if not all authentication methods can be technically supported. The available methods then are presented to the PSU for choice.



3.1.6 Multilevel SCA Approach: Example Decoupled SCA Approach

The multilevel SCA Approach flows for the Establish Consent Requests will follow exactly the same pattern as for the Payment Initiation Services. Whereas the Redirect SCA Approach

was used there as an example, the following flow will give an example for the Decoupled SCA Approach:



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

3.2 Data Transport Overview

The following table does not only define requirements on request messages but also requirements on data elements for the response messages. The requirements on responses only apply to positive responses (i.e. HTTP response code 2xx). Error information is communicated as defined service independently in [oFA-PFSM].

The Establish Consent Request show more headers in the table than in the detailed definitions below. Please note that all header parameters supported for Transaction Initiation as defined in [oFA-PFSM] e.g. for PSU identification, SCA mechanism or TPP notification apply also to the Establish Consent Request. Details can be retrieved from the related Open API files.

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

Data element	Attribute encoding	Location		Usage						
		Path	Query Param.	Header	Body	Certificate	Estab.. Cons.Req.	Estab.. Cons.Resp.	Consent Status Req.	Consent Status Rsp.
Provider Identification		x					m		m	
TPP Registration Number					x	m		m		
TPP Name					x	m		m		
TPP Role					x	m		m		
TPP National Competent Authority					x	m		m		
Request Identification	X-Request-ID		x			m	m	m	m	
Resource ID	consentId			x			m			
Resource ID ³		x						m		

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

Data element	Attribute encoding	Location			Usage					
		Path	Query Param.	Header	Body	Certificate	Established.. Cons. Req.	Established.. Cons. Resp.	Consent Status Req.	Consent Status Rsp.
Access Token (from optional OAuth2)	Authorization		x			c		c		
API Client Signing Certificate Data	Client-Signature-Certificate		x			c		c		
API Client Signing Electronic Signature	x-jws-signature		x			c		c		
Further signature related data	Digest		x			c		c		
ASPSP-SCA-Approach	ASPSP-SCA-Approach		x				c			
ASPSP Multiple Consents Support	ASPSP-Multiple-Consent-Support		x				c			
Transaction Status	consentStatus			x			m		m	
SCA Status	scaStatus			x						o
PSU Message Information	psuMessage			x			o		o	
API Client Message Information	apiClientMessages			x			o		o	
PSU Identification	PSU-ID		x			c				
PSU Identification Type	PSU-ID-Type		x			c				
Corporate Identification	PSU-Corporate-ID		x			c				
Corporate Type	PSU-Corporate-ID-Type					c				
Available SCA Methods	scaMethods			x			c			
SCA Challenge Data	challengeData			x			c			
IP Address PSU	PSU-IP-Address		x			m		o		
PSU IP Port	PSU-IP-Port		x			o		o		

Data element	Attribute encoding	Location		Usage			
		Path	Query Param.	Header	Body	Certificate	
			x				o Establ.. Cons. Req.
Further PSU related Information	PSU-Accept		x				o Establ.. Cons. Resp.
	PSU-Accept-Charset		x				o Consent Status Req.
	PSU-Accept-Encoding		x				o Consent Status Rsp.
	PSU-Accept-Language		x				o Consent Status Rsp.
	PSU-Http-Method		x				o Consent Status Rsp.
	PSU-Device-ID		x				o Consent Status Rsp.
PSU User Agent	PSU-User-Agent		x				o Consent Status Rsp.
GEO Information	PSU-Geo-Location		x				o Consent Status Rsp.
Redirect URL ASPSP	_links.scaRedirect			x		c	
SCA Approach Preference	Client-SCA-Approach-Preference		x				
Redirect URL TPP	Client-Redirect-URI		x			c	
Authorisation Preference	Client-Explicit-Authorisation-Preferred		x				
Client Notification URI	Client-Notification-URI		x				
Client Notification Content Preference	Client-Notification-Content-Preferred		x				
Client Brand Information	Client-Brand-Logging-Information		x				
PSU Account Access	access			x		m	
Consent Type	consentType						
Validity Period	validTo			x		m	

Data element	Attribute encoding	Location			Usage					
		Path	Query Param.	Header	Body	Certificate	Established Cons. Req.	Established Cons. Resp.	Consent Status Req.	Consent Status Rsp.
Frequency	frequencyPerDay			x		m				
Recurring Indicator	recurringIndicator			x		m				

Remark: The upper table refers to the "Consent Account Information Request" for consentType "detailed". Establish Consent Processes for other consent types or other consent category are defined analogously.

3.3 Multicurrency Accounts in Submission of Consents

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

3.4 Establish Consent on Account Information

In this section, the Establish Consent on Account Information process is defined for the openFinance API.

3.4.1 Consent Request for Account Information

Call

```
POST /v2/consents/account-access
```

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

Side Effects

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

Please note that in the context of multilevel SCA, the side effect may only apply, if a PSU with a given PSU-ID is submitting a new recurring consent and a second recurring consent exists already which had been submitted under the same PSU and potentially corporate identification.

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

Optional Extended Service for Multiple Consents.

The ASPSP may offer optionally to the TPP not to let automatically expire consent tokens when a new recurring consent token for the same PSU is requested. This is called "Multiple Consents Service" in the following. Thus, the above described side effect does not apply in this case.

Remark. The Multiple Consents Service was specified within NextGenPSD2 APIs Version 1.3.x in the dedicated service description [XS2A MCon].

Query Parameters

No specific query parameter.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Mandatory	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. If not available, the TPP shall use the IP Address used by the TPP when submitting this request.
Authorization	String	Conditional	If OAuth2 has been chosen as pre-step to authenticate the PSU.

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

Request Body

Attribute	Type	Condition	Description
access	Account Access	Mandatory	Requested access services.
consentType	Consent Type Code	Mandatory	<p>The technical consent type.</p> <p>NOTE: The choice of this type has an effect on attribute conditions in the "access" attribute, see below.</p>
recurringIndicator	Boolean	Mandatory	<p>true, if the consent is for recurring access to the account data</p> <p>false, if the consent is for one access to the account data</p>
validTo	ISODate	Mandatory	<p>This parameter is defining a valid until date (including the mentioned date) for the requested consent. The content is the local ASPSP date in ISODate Format, e.g. 2024-10-30.</p> <p>Future dates might get adjusted by ASPSP.</p> <p>If a maximum available date is requested, a date in far future is to be used: "9999-12-31".</p> <p>In both cases, the consent object to be retrieved by the GET Consent Request will contain the adjusted date.</p> <p>The support of this adaption of the maximum available date is mandatory for the ASPSP.</p>
frequencyPerDay	Integer	Mandatory	<p>This field indicates the requested maximum frequency for an access without PSU involvement per day. For a one-off access, this attribute is set to "1".</p> <p>The frequency needs to be greater equal to one. If not otherwise agreed bilaterally between TPP and</p>



Attribute	Type	Condition	Description
			ASPSP, the frequency is less equal to 4.

This specification mandates the ASPSP to support the consentType "detailed" as a minimum. The support of all other consentTypes is optional for the ASPSP, taking into consideration that the ASPSP shall offer at least one of the other models to achieve the "no manual IBAN entry" requirement by EBA. In case of the usage of a not supported consentType by the API Client, the ASPSP will use a dedicated Message Code in the error messaging. The following consentTypes have specific restrictions on the usage of access rights, as defined in the following. Please note, that discovery services will provide the related information, if supported by the ASPSP, cp. [oFA Disc].

- detailed

The access rights "accountDetails", "balances", "transactions", "trustedBeneficiaries" as well as "ownerName" and "psuName" may be used by the API Client. The ASPSP may offer in addition the access right "paymentInitiations" if the usage of the consent object for payment initiations is supported. At the minimum one access right code has to be used for every provided account reference.

- global

The access right "ais" shall be used by the API Client for each chosen account category. At least one account category has to be provided by the API Client. In addition, the API Client may use the access right "ownerName".

- aspspManaged

The access rights "accountDetails", "balances", "transactions", "trustedBeneficiaries" as well as "ownerName" and "psuName" may be used by the API Client. Also, **no** access right could be used in this case, indicating that the PSU will choose the related access rights during authorisation.

- accountList

The access right "accountDetails" shall be used by the API Client for each account category. In addition, the API Client may use the access rights "ownerName" and "balances". The usage by the latter rights might lead to the need to apply SCA for the related consents, if the ASPSP offers the "pure" accountList information with just one customer authentication factor.

If the access right "balances" or "transactions" is used, then the consent on providing "accountDetails" is implicitly given.

If the access right "ownerName" is used, then the ASPSP is requested to support the delivery of the account owner. Please note that the account owner could also be delivered within the account details, even if the "ownerName" attribute is not used.

If the access right "trustedBeneficiaries" is used and if the ASPSP is not supporting the extended services to display trusted beneficiaries, then the ASPSP will reject the consent request. Please note that the trusted beneficiary list could also be delivered with the account details even without the trustedBeneficiary access right code used in the related consent. Still, the ASPSP in this case needs to support the "trustedBeneficiary" access right code.

Response Code

HTTP Response Code equals 201.

Response Header

Attribute	Type	Condition	Description
Location	String	Mandatory	Location of the created resource.
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
ASPSP-Multiple-Consent-Support	Boolean	Conditional	<p>true if the ASPSP supports the Multiple Consent Service. In case of support of the Multiple Consent Service, this header attribute is mandatory.</p> <p>false if the ASPSP does not support the Multiple Consent Service.</p> <p>If not provided, this also implies that the ASPSP does not support the Multiple Consent Service.</p>
ASPSP-SCA-Approach	String	Conditional	<p>Possible values are:</p> <ul style="list-style-type: none"> • EMBEDDED • DECOUPLED • REDIRECT • SIGNATURE • ASPSP-CHANNEL <p>OAuth will be subsumed by the constant value REDIRECT</p>

Response Body

Attribute	Type	Condition	Description
consentStatus	Consent Status	Mandatory	authentication status of the consent
consentId	Max70Text	Mandatory	Identification of the consent resource as it is used in the API structure
scaMethods	Array of Authentication Objects	Conditional	<p>This data element might be contained, if SCA is required and if the PSU has a choice between different authentication methods. Depending on the risk management of the ASPSP this choice might be offered before or after the PSU has been identified with the first relevant factor, or if an access token is transported. If this data element is contained, then there is also a hyperlink of type "selectAuthenticationMethods" contained in the response body.</p> <p>These methods shall be presented towards the PSU for selection by the TPP.</p>
chosenScaMethod	Authentication Object	Conditional	This data element is only contained in the response if the ASPSP has chosen the Embedded SCA Approach, if the PSU is already identified with the first relevant factor or alternatively an access token, if SCA is required and if the authentication method is implicitly selected.
challengeData	Challenge	Conditional	<p>It is contained in addition to the data element chosenScaMethod if challenge data is needed for SCA.</p> <p>In rare cases this attribute is also used in the context of the startAuthorisationWithPsuAuthentication or startAuthorisationWithEncryptedPsuAuthentication link.</p>
_links	Links	Mandatory	<p>A list of hyperlinks to be recognised by the API Client</p> <p>Type of links admitted in this response (which might be extended by single ASPSPs as indicated in its XS2A documentation): are defined in [oFA-PFSM] generically for all Transaction Initiation Response messages.</p>
psuMessage	Max500Text	Optional	Text to be displayed to the PSU, e.g. in a Decoupled SCA Approach



Example

Request in a PSD2 compliance XS2A API

```
POST https://api.testbank.com/psd2/v2/consents/account-access
Content-Type: application/json
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
PSU-IP-Address: 192.168.8.78
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0
Date: Sun, 06 Aug 2017 15:05:37 GMT

{
  "access": {
    "payments": /* payment accounts
      [{"account": account ref1,
        "rights": ["accountDetails", "balances", "transactions",
"ownerName", "trustedBeneficiaries"]},
      {"account": account ref2,
        "rights": ["accountDetails", "balances", "transactions",
"ownerName"]}
    ],
    },
    "consentType": "detailed",
    "recurringIndicator": true,
    "validTo": "2024-11-01",
    "frequencyPerDay": 4
  }
}
```

Request in a premium AIS within an API Access Scheme:

```
POST https://api.testbank.com/psd2/v2/consents/account-access
Content-Type: application/json
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
PSU-IP-Address: 192.168.8.78
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0
Date: Sun, 06 Aug 2017 15:05:37 GMT

{
  "access": {
    "payments": /* payment accounts
      [{"account": account ref1, /* account ref n stands for a JSON
structure of type account reference
        "rights": ["accountDetails", "balances", "transactions",
"ownerName", "trustedBeneficiaries"]},
      {"account": account ref2,
        "rights": ["accountDetails", "balances", "transactions",
"ownerName"]}
    ],
    },
    "consentType": "detailed",
    "recurringIndicator": true,
    "validTo": "2024-11-01",
    "frequencyPerDay": 4
  }
}
```

```

"cards": /* this is addressing single cards entries only
  [{"account": account ref3,
    "rights": ["accountDetails", "balances", "transactions"]},
   {"account": account ref4,
    "rights": ["accountDetails", "transactions"]}
  ],
"card-accounts": /* this is addressing specific card recon accounts
  [{"account": account ref4,
    "rights": ["accountDetails", "balances", "transactions"]}
  ],
"savings":
  [{"account": account ref5,
    "rights": ["accountDetails", "balances"]}]
],
"loans":
  [{"account": account ref6,
    "rights": ["accountDetails", "balance", "transactions"]},
   {"account": account ref7,
    "rights": ["accountDetails", "balances"]}
  ],
"securities":
  [{"account": account ref8,
    "rights": ["accountDetails", "balances", "orders",
"transactions"]}]
]
}
"consentType": "detailed",
"recurringIndicator": true,
"validTo": "2024-11-01",
"frequencyPerDay": 4
}

```

Response in case of a redirect

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: REDIRECT
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: /psd2/v2/consents/account-access/3d9a81b3-a47d-4130-8765-a9c0ff861101"
Content-Type: application/json

```
{
"consentStatus": "received",
"consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861101",
"_links": {
  "scaRedirect": {"href":
"https://www.testbank.com/authentication/3d9a81b3-a47d-4130-8765-
a9c0ff861101"},
```



```

    "status": {"href": "/psd2/v2/consents/account-access/3d9a81b3-a47d-
4130-8765-a9c0ff861101/status"},  

    "scaStatus": {"href": "/psd2/v2/consents/account-access/3d9a81b3-a47d-
4130-8765-a9c0ff861101/
authorisations/123auth567"}  

}  

}

```

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

```

HTTP/1.x 201 Created  

X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721  

ASPSP-SCA-Approach: REDIRECT  

Date: Sun, 06 Aug 2017 15:05:47 GMT  

Location: "/psd2/v2/consents/account-access/3d9a81b3-a47d-
4130-8765-a9c0ff861101"  

Content-Type: application/json  
  

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861101",
  "_links": {
    "startAuthorisation": {"href": "/psd2/v2/consents/account-
access/3d9a81b3-a47d-4130-8765-a9c0ff861101/authorisations"}
  }
}

```

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

```

HTTP/1.x 201 Created  

X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721  

ASPSP-SCA-Approach: REDIRECT  

Date: Sun, 06 Aug 2017 15:05:47 GMT  

Location: "/psd2/v2/consents/account-access/3d9a81b3-a47d-
4130-8765-a9c0ff861101"  

Content-Type: application/json  
  

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861101",
  "_links": {
    "self": {"href": "/psd2/v2/consents/account-access/3d9a81b3-a47d-4130-
8765-a9c0ff861101"},  

    "scaStatus": {"href": "/psd2/v2/consents/account-access/3d9a81b3-a47d-
4130-8765-a9c0ff861101/authorisations/123auth567"},  

    "scaOAuth": {"href": "https://www.testbank.com/oauth/.well-known/oauth-
authorization-server"}
  }
}

```

Response in case of the decoupled approach

```

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: DECOUPLED
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "/psd2/v2/consents/account-access/3d9a81b3-a47d-
4130-8765-a9c0ff861101"
Content-Type: application/json

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861101",
  "_links": {
    "startAuthorisationWithPsuIdentification": {"href": "
"/psd2/v2/consents/account-access/3d9a81b3-a47d-4130-8765-
a9c0ff861101/authorisations"}
  }
}

```

Response in case of the embedded approach

```

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: EMBEDDED
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "/psd2/v2/consents/account-access/3d9a81b3-a47d-
4130-8765-a9c0ff861101"
Content-Type: application/json

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861101",
  "_links": {
    "startAuthorisationWithPsuAuthentication": {"href": "
"/psd2/v2/consents/account-access/3d9a81b3-a47d-4130-8765-
a9c0ff861101/authorisations"}
  }
}

```

Response in case of the signature approach

```

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: SIGNATURE
Body-Sig-Profile-Accepted: JAdES_JS
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "/psd2/v2/consents/account-access/3d9a81b3-
a47d-4130-8765-a9c0ff861101"

```

```
Content-Type: application/json

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861101",
  "_links": {
    "startAuthorisationWithTransactionAuthorisation": {"href": "/psd2/v2/consents/account-access/3d9a81b3-a47d-4130-8765-a9c0ff861101/authorisations"}
  }
}
```

Example Consent on Account List of Available Accounts

Request

POST <https://api.testbank.com/psd2/v2/consents/account-access>

Content-Type: application/json
 X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
 PSU-IP-Address: 192.168.8.78
 PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0
 Date: Sun, 06 Aug 2017 15:05:37 GMT

```
{
  "access": {
    "payments": /* payment accounts
      [{"rights": ["accountDetails", "ownerName"]}]
    ],
    "cards": [
      {"rights": ["accountDetails"]}
    ],
    "savings": [
      {"rights": ["accountDetails"]}
    ],
    "loans": [
      {"rights": ["accountDetails"]}
    ],
    "securities": [
      {"rights": ["accountDetails"]}
    ]
  }
}

"consentType": "accountList",
"recurringIndicator": false,
"validTo": "2017-08-06",
"frequencyPerDay": 1
}
```

Example Consent without dedicated Account

Request

```
POST https://api.testbank.com/psd2/v2/consents/account-access
Content-Type application/json
X-Request-ID 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
PSU-IP-Address 192.168.8.78
PSU-User-Agent Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
Date Sun, 06 Aug 2017 15:05:37 GMT

{
  "access": {
    "payments": /* payment accounts
      [{"rights": ["accountDetails", "balances", "transactions",
      "ownerName", "trustedBeneficiaries"]}]
    ],
    "cards": [
      {"rights": ["accountDetails", "transactions"]}
    ],
    "savings": [
      {"rights": ["accountDetails", "balances"]}
    ],
    "loans": [
      {"rights": ["accountDetails", "balances"]}
    ],
    "securities": [
      {"rights": ["accountDetails", "balances", "orders",
      "transactions"]}
    ]
  }
  "consentType": "aspsspManaged",
  "recurringIndicator": true,
  "validTo": "2017-11-01",
  "frequencyPerDay": 4
}
```

Example global consent

Request

```
POST https://api.testbank.com/psd2/v2/consents/account-access
Content-Type application/json
X-Request-ID 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
PSU-IP-Address 192.168.8.78
PSU-User-Agent Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
Date Sun, 06 Aug 2017 15:05:37 GMT

{
  "access": {
```

```

"payments": /* payment accounts
  [{"rights": ["ais", "ownerName"]}]
},
"cards":
  [{"rights": ["ais"]}]
],
"savings":
  [{"rights": ["ais"]}]
],
"loans":
  [{"rights": ["ais"]}],
],
"securities":
  [{"rights": ["ais"]}]
}
}

"consentType": "accountList",
"recurringIndicator": false,
"validTo": "2017-08-06",
"frequencyPerDay": 1
}

```

3.4.2 Get Consent Request

Call

GET /v2/consents/account-access/{consentId}

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

Path Parameters

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

Query Parameters

No specific query parameter.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based SCA was performed in the corresponding consent transaction or if OAuth2 has been used in a pre-step.

Request Body

No request body.

Response Code

HTTP Response Code equals 200.

Response Header

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

Response Body

Attribute	Type	Condition	Description
access	Extended Account Access	Mandatory	
consentType	Consent Type	Mandatory	
recurringIndicator	Boolean	Mandatory	
validTo	ISODate	Mandatory	
frequencyPerDay	Integer	Mandatory	
consentStatus	Consent Status	Mandatory	The status of the consent resource.



NOTE: For the consent category "account-access", the ASPSP **shall** deliver all detailed account references and consented access rights after positive SCA by the PSU within the consent object regardless of the consentType chosen in the related Establish Consent Request. In addition, hyperlinks to all accounts which are provided in the consent shall be added. In this way, the TPP can read from the consent object all relevant account information as a shortcut to get direct access to all accessible accounts without calling the endpoints /accounts, /loans etc. first.

Example

Request

```
GET https://api.testbank.com/psd2/v2/consents/account-access/3d9a81b3-a47d-4130-8765-a9c0ff861101
```

Response (in case of a premium AIS within an API Access Scheme)

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

```
{
  "access": {
    "payments": /* payment accounts
      [{"account": account ref1,
        "rights": ["accountDetails", "balances", "transactions",
"ownerName", "trustedBeneficiaries"],
        "_links": {"account": {"href": "/psd2/v2/accounts/resourceId1"}}
      },
      {"account": account ref2,
        "rights": ["accountDetails", "balances", "transactions",
"ownerName"],
        "_links": {"account": {"href": "/psd2/v2/accounts/resourceId2"}}
      }],
    "cards": [
      {"account": account ref3,
        "rights": ["accountDetails", "balances", "transactions"],
        "_links": {"card": {"href": "/oFA/v2/cards/resourceId3"}}
      },
      {"account": account ref4,
        "rights": ["accountDetails", "transactions"],
        "_links": {"card": {"href": "/oFA/v2/cards/resourceId4"}}
      }
    ],
    "savings": [
      {"account": account ref5,
        "rights": ["accountDetails", "balances"],
```

```

        "_links": {"savingsAccount": {"href": "/oFA/v2/savings-
accounts/resourceId5"}}
    }
],
"loans":
[{"account": account ref6,
    "rights": ["accountDetails", "balances", "transactions"],
    "_links": {"loanAccount": {"href": "/oFA/v2/loan-
accounts/resourceId6"}}
},
 {"account": account ref7,
    "rights": ["accountDetails", "balances"],
    "_links": {"loanAccount": {"href": "/oFA/v2/loan-
accounts/resourceId7"}}
},
 ],
"securities":
[{"account": account ref8,
    "rights": ["accountDetails", "balances", "orders",
"transactions"],
    "_links": {"securitiesAccount": {"href": "/oFA/v2/securities-
accounts/resourceId8"}}
}
]
}

"consentType": "aspssManaged",
"recurringIndicator": true,
"validTo": "2017-11-01",
"frequencyPerDay": 4,
"consentStatus": "valid",
}

```

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

3.5 Establish Consent on Funds Confirmation

In this section, the Establish Consent on Funds Confirmation process is defined for the openFinance API Framework.

3.5.1 Consent on Funds Confirmation Request

Call

POST /v2/consents/funds-confirmations

Creates a consent on confirmation of funds resource at the ASPSP regarding confirmation of funds access to an account specified in this request.

Remark: In [XS2A-CCF], a first version of this API was already published under the endpoint /v2/consents/confirmation-of-funds. This API has been adapted now under the different endpoint name to the new "access" and "consentType" data attribute usage.

No Side Effects

In difference to the Establish Consent on Account Information as defined in Section 3.4.1, there is no side effect by the Establish Consent on Funds Confirmation Request.

Query Parameters

No specific query parameter.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Mandatory	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. If not available, the TPP shall use the IP Address used by the TPP when submitting this request.
Authorization	String	Conditional	If OAuth2 has been chosen as pre-step to authenticate the PSU.

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

Request Body

Attribute	Type	Condition	Description
access	Account Access	Mandatory	requested access rights on accounts, where the confirmation of



Attribute	Type	Condition	Description
			funds service is aimed to be submitted to.
consentType	Consent Type	Mandatory	Only the consentTypes "detailed" and "aspsspManaged" may be supported.
recurringIndicator	Boolean	Mandatory	true, if the consent is for recurring access to the account data false, if the consent is for one access to the account data
validTo	ISODate	Mandatory	This parameter is defining a valid until date (including the mentioned date) for the requested consent. The content is the local ASPSP date in ISODate Format, e.g. 2024-10-30. There is no legal restriction of the validTo entry, but maybe a restriction from the ASPSP. Future dates might get adjusted by ASPSP. If a maximum available date is requested, a date in far future is to be used: "9999-12-31". In both cases, the consent object to be retrieved by the GET Consent Request will contain the adjusted date. The support of this adaption of the maximum available date is mandatory for the ASPSP.
cardNumber	Max35Text	Optional	Card Number of the card issued by the PIISP. Should be delivered if available.
cardExpiryDate	ISODate	Optional	Expiry date of the card issued by the PIISP
cardInformation	Max140Text	Optional	Additional explanation for the card product.



Attribute	Type	Condition	Description
registrationInformation	Max140Text	Optional	Additional information about the registration process for the PSU, e.g. a reference to the TPP / PSU contract

This specification mandates the ASPSP to support the consentType "detailed" and/or "aspsspManaged". In both cases, only accounts of account category "payments" may be contained in the access attribute. In case of the usage of a not supported consentType by the API Client, the ASPSP will use a dedicated Message Code in the error messaging. The following consentTypes have specific restrictions on the usage of access rights, as defined in the following. Please note, that discovery services will provide the related information, if supported by the ASPSP, cp. [oFA Disc].

- detailed

The access rights "fundsConfirmations" and "psuName" may be used by the API Client. The usage of the access right "fundConfirmations" is mandated.

- aspsspManaged

The access rights "fundsConfirmations" and "psuName" may be used by the API Client. The usage of the access right "fundsConfirmations" is mandated.

Response Code

HTTP Response Code equals 201.

Response Header

Attribute	Type	Condition	Description
Location	String	Mandatory	Location of the created resource.
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
ASPSP-SCA-Approach	String	Conditional	Possible values are: <ul style="list-style-type: none"> • EMBEDDED • DECOUPLED • REDIRECT • SIGNATURE • ASPSP-CHANNEL

Attribute	Type	Condition	Description
			OAuth will be subsumed by the constant value REDIRECT

Response Body

Attribute	Type	Condition	Description
consentId	String	Mandatory	Identification of the consent resource as it is used in the API structure
consentStatus	Consent Status	Mandatory	authentication status of the consent
scaMethods	Array of Authentication Objects	Conditional	<p>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 "selectAuthenticationMethods" contained in the response body.</p> <p>These methods shall be presented towards the PSU for selection by the TPP.</p>
chosenScaMethod	Authentication Object	Conditional	This data element is only contained in the response if the ASPSP has chosen the Embedded SCA Approach, if the PSU is already identified with the first relevant factor or alternatively an access token, if SCA is required and if the authentication method is implicitly selected.
challengeData	Challenge	Conditional	<p>It is contained in addition to the data element chosenScaMethod if challenge data is needed for SCA.</p> <p>In rare cases this attribute is also used in the context of the startAuthorisationWithPsuAuthentication or startAuthorisationWithEncryptedPsuAuthentication link.</p>
_links	Links	Mandatory	<p>A list of hyperlinks to be recognised by the TPP.</p> <p>Type of links admitted in this response (which might be extended by single ASPSPs as indicated in its</p>



Attribute	Type	Condition	Description
			XS2A documentation) are defined in [oFA-PFSM] generically for all Transaction Initiation Response messages. The related Open API files will also contain all details.
psuMessage	Max512Text	Optional	Text to be displayed to the PSU, e.g. in a Decoupled SCA Approach

Example

Request

```
POST https://api.testbank.com/v2/consents/funds-confirmations
Content-Type: application/json
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
PSU-IP-Address: 192.168.8.78
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
Date: Sun, 06 Aug 2017 15:05:37 GMT

{
  "access": {
    "payments": /* payment accounts
      [ {"account": account ref1,
        "rights": ["fundsConfirmations", "psuName"] }
      ]
    },
  .."consentType": "detailed",
  "recurringIndicator": "true",
  "validTo": "9999-12-31",
  "cardNumber": "1234567891234",
  "cardExpiryDate": "2020-12-31",
  "cardInformation": "MyMerchant Loyalty Card",
  "registrationInformation": "Your contract Number 1234 with MyMerchant is completed with the registration with your bank."
}

}
```

Response in case of a redirect

```
HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: REDIRECT
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "v2/consents/funds-confirmations/3d9a81b3-a47d-4130-8765-a9c0ff861102"
Content-Type: application/json
```

```
{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861102",
  "_links": {
    "scaRedirect": {"href":
      "https://www.testbank.com/authentication/3d9a81b3-a47d-4130-8765-
      a9c0ff861102"},
    "status": {"href": "/v2/consents/funds-confirmations/3d9a81b3-a47d-
      4130-8765-a9c0ff861102/status"},
    "scaStatus": {"href": "/v2/consents/funds-confirmations/3d9a81b3-a47d-
      4130-8765-a9c0ff861102/
      authorisations/123auth567"}
  }
}
```

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

HTTP/1.x 201 Created
 X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
 ASPSP-SCA-Approach: REDIRECT
 Date: Sun, 06 Aug 2017 15:05:47 GMT
 Location: "v2/consents/funds-confirmations/3d9a81b3-a47d-
 4130-8765-a9c0ff861102"
 Content-Type: application/json

```
{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861102",
  "_links": {
    "startAuthorisation": {"href": "v2/consents/funds-
      confirmations/3d9a81b3-a47d-4130-8765-a9c0ff861102/authorisations"}
  }
}
```

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

HTTP/1.x 201 Created
 X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
 ASPSP-SCA-Approach: REDIRECT
 Date: Sun, 06 Aug 2017 15:05:47 GMT
 Location: "v2/consents/funds-confirmations/3d9a81b3-a47d-
 4130-8765-a9c0ff861102"
 Content-Type: application/json

```
{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861102",
```

```

  "_links": {
    "self": {"href": "/v2/consents/funds-confirmations/3d9a81b3-a47d-4130-
8765-a9c0ff861102"},
    "scaStatus": {"href": "v2/consents/funds-confirmations/3d9a81b3-a47d-
4130-8765-a9c0ff861102/authorisations/123auth567"},
    "scaOAuth": {"href": "https://www.testbank.com/oauth/.well-known/oauth-
authorization-server"}
  }
}

```

Response in case of the decoupled approach

```

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSA-SCA-Approach: DECOUPLED
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "v2/consents/funds-confirmations/3d9a81b3-a47d-
4130-8765-a9c0ff861102"
Content-Type: application/json

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861102",
  "_links": {
    "startAuthorisationWithPsuIdentification": {"href": "
/v2/consents/funds-confirmations/3d9a81b3-a47d-4130-8765-
a9c0ff861102/authorisations"}
  }
}

```

Response in case of the signature approach

```

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSA-SCA-Approach: SIGNATURE
Body-Sig-Profile-Accepted: JAdES_JS
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "v2/consents/funds-confirmations/3d9a81b3-
a47d-4130-8765-a9c0ff861102"
Content-Type: application/json

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861102",
  "_links": {

```

```

    "startAuthorisationWithPsuIdentification": {"href":
"/v2/consents/funds-confirmations/3d9a81b3-a47d-4130-8765-
a9c0ff861102/authorisations"}
}
}

```

Response in case of the embedded approach

HTTP/1.x 201 Created

X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721

ASPSP-SCA-Approach: EMBEDDED

Date: Sun, 06 Aug 2017 15:05:47 GMT

Location: "v2/consents/funds-confirmations/3d9a81b3-a47d-4130-8765-a9c0ff861102"

Content-Type: application/json

```
{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861102",
  "_links": {
    "startAuthorisationWithPsuAuthentication": {"href":
"/v2/consents/funds-confirmations/3d9a81b3-a47d-4130-8765-
a9c0ff861102/authorisations"}
  }
}
```

3.5.2 Get Consent Request

Call

GET </v2/consents/funds-confirmations/{consentId}>

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

Path Parameters

Attribute	Type	Description
consentId	Max70Text	ID of the corresponding consent object as returned by a Funds Confirmation Consent Request

Query Parameters

No specific query parameter.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based SCA was performed in the corresponding consent transaction or if OAuth2 has been used in a pre-step.

Request Body

No request body.

Response Code

HTTP Response Code equals 200.

Response Header

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

Response Body

Attribute	Type	Condition	Description
access	Account Access	Mandatory	Account access details, where the confirmation of funds service is aimed to be submitted to.
consentType	Consent Type	Mandatory	
cardNumber	Max35Text	Optional	Card Number of the card issued by the PIISP. Should be delivered if available.
cardExpiryDate	ISODate	Optional	Expiry date of the card issued by the PIISP
cardInformation	Max140Text	Optional	Additional explanation for the card product.



Attribute	Type	Condition	Description
registrationInformation	Max140Text	Optional	Additional registration information.
consentStatus	Consent Status	Mandatory	The status of the consent resource.

Example

Request

GET <https://api.testbank.com/dv2/consents/funds-confirmations/3d9a81b3-a47d-4130-8777-a9c0ff861110>

Response

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

```
{
  "access": {
    "payments": [
      {"account": {"iban": "DE40100100103307118608"},  

       "accessRights": {"fundsConfirmations"}]
    },
    "consentType": "detailed",
    "cardNumber": "1234567891234",
    "cardExpiryDate": "2020-12-31",
    "cardInformation": "MyMerchant Loyalty Card",
    "consentStatus": "valid"
  }
}
```

3.6 Establish Consent on User Parameters Access

In this section, the Establish Consent on User Parameters Access is defined for the openFinance API Framework. User parameters are user related technical access data which is needed for the API Client so steer the user frontend e.g. for authorisation processes. This data is specifically addressing use cases for direct access of the API by API client software. It can be also used for retrieving this sort of information on PSUs for Third Parties, e.g. in account check services.

NOTE: User Parameter services will be described in [oFA Disc], which defines discovery services on many levels of the API. This document is still in draft mode and will be provided for market consultation later.

Remark: Due to the generic usage within TPP and direct access use cases, the more neutral term "user" is preferred for the owner/people with signing rights for the related account. In situations which are TPP specific, the term "PSU" is still used since this is the term in the TPP services related data model.

3.6.1 Consent on User Parameters Access Request

Call

```
POST /v2/consents/user-parameters-access
```

Creates a consent on user related technical access data (user parameters) at the ASPSP regarding accounts addressed in this request.

Side Effects

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

Please note that in the context of multilevel SCA, the side effect may only apply, if a PSU with a given PSU-ID is submitting a new recurring consent and a second recurring consent exists already which had been submitted under the same PSU and potentially corporate identification.

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

Query Parameters

No specific query parameter.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Mandatory	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. If not available, the TPP shall use the IP Address used by the TPP when submitting this request.
Authorization	String	Conditional	If OAuth2 has been chosen as pre-step to authenticate the PSU.

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

Request Body

Attribute	Type	Condition	Description
access	Account Access	Mandatory	Requested access services.
consentType	Consent Type Code	Mandatory	The technical consent type. NOTE: The choice of this type has an effect on attribute conditions in the "access" attribute, see below.
recurringIndicator	Boolean	Mandatory	true, if the consent is for recurring access to the user parameters false, if the consent is for one access to the user parameters
validTo	ISODate	Mandatory	This parameter is defining a valid until date (including the mentioned date) for the requested consent. The



Attribute	Type	Condition	Description
			<p>content is the local ASPSP date in ISODate Format, e.g. 2024-10-30.</p> <p>Future dates might get adjusted by ASPSP.</p> <p>If a maximum available date is requested, a date in far future is to be used: "9999-12-31".</p> <p>In both cases, the consent object to be retrieved by the GET Consent Request will contain the adjusted date.</p> <p>The support of this adaption of the maximum available date is mandatory for the ASPSP.</p>

This specification mandates the ASPSP to support the consentType "detailed" and/or "aspsspManaged". In case of the usage of a not supported consentType by the API Client, the ASPSP will use a dedicated Message Code in the error messaging. The following consentTypes have specific restrictions on the usage of access rights, as defined in the following. Hereby, the consent usage for the generic user parameters service respectively the restricted IBAN Check Service is different:

Generic User Parameter Service:

- detailed

The access rights "userParameters" and "psuName" may be used by the API Client. The usage of the access right "userParameters" is mandated.

- aspsspManaged

The access rights "userParameters" and "psuName" may be used by the API Client. If no code is provided, the PSU will select the related code during authorisation.

IBAN Check Service:

- detailed

The access right codes "ibanChecks" and "psuName" may be used by the API Client. The access right codes "accountDetails" and "owner" may be used, if supported by the ASPSP. The usage of the access right "ibanChecks" is mandated.

- aspsspManaged

The access rights "ibanChecks" and "psuName" may be used by the API Client. The access right codes "accountDetails" and "owner" may be used, if supported by the ASPSP. The usage of the access right "ibanChecks" is mandated.

Please note, that discovery services will provide the related information, if supported by the ASPSP, cp. [oFA Disc]. Please further note, that the "owner" access right code also include the access on the /accounts endpoint to retrieve the owner name.

Response Code

HTTP Response Code equals 201.

Response Header

Attribute	Type	Condition	Description
Location	String	Mandatory	Location of the created resource.
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
ASPSP-SCA-Approach	String	Conditional	<p>Possible values are:</p> <ul style="list-style-type: none"> • EMBEDDED • DECOUPLED • REDIRECT • SIGNATURE • ASPSP-CHANNEL <p>OAuth will be subsumed by the constant value REDIRECT</p>

Response Body

Attribute	Type	Condition	Description
consentId	Max70Text	Mandatory	Identification of the consent resource as it is used in the API structure
consentStatus	Consent Status	Mandatory	authentication status of the consent

Attribute	Type	Condition	Description
scaMethods	Array of Authentication Objects	Conditional	<p>This data element might be contained, if SCA is required and if the PSU has a choice between different authentication methods. Depending on the risk management of the ASPSP this choice might be offered before or after the PSU has been identified with the first relevant factor, or if an access token is transported. If this data element is contained, then there is also a hyperlink of type "selectAuthenticationMethods" contained in the response body.</p> <p>These methods shall be presented towards the PSU for selection by the TPP.</p>
chosenScaMethod	Authentication Object	Conditional	<p>This data element is only contained in the response if the ASPSP has chosen the Embedded SCA Approach, if the PSU is already identified with the first relevant factor or alternatively an access token, if SCA is required and if the authentication method is implicitly selected.</p>
challengeData	Challenge	Conditional	<p>It is contained in addition to the data element chosenScaMethod if challenge data is needed for SCA.</p> <p>In rare cases this attribute is also used in the context of the startAuthorisationWithPsuAuthentication or startAuthorisationWithEncryptedPsuAuthentication link.</p>
_links	Links	Mandatory	<p>A list of hyperlinks to be recognised by the API Client.</p> <p>Type of links admitted in this response (which might be extended by single ASPSPs as indicated in its documentation) are defined in [oFA-PFSM] generically for all Transaction Initiation Response messages. The related Open API files will also contain all details.</p>
psuMessage	Max512Text	Optional	<p>Text to be displayed to the PSU, e.g. in a Decoupled SCA Approach</p>

Example with detailed access data

Request

```
POST https://api.testbank.com/v2/consents/user-parameters-access
Content-Type: application/json
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
PSU-IP-Address: 192.168.8.78
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
Date: Sun, 06 Aug 2017 15:05:37 GMT

{
  "access": {
    "payments": /* payments
      [{"account": account ref1,
        "rights": ["userParameters", "psuName"]},
      {"account": account ref2,
        "rights": ["userParameters", "psuName"]}
      ],
    "savings": [
      {"account": account ref1,
        "rights": ["userParameters"]},
      {"account": account ref2,
        "rights": ["userParameters"]}]
      ],
    "cards": [
      {"account": account ref1,
        "rights": ["userParameters"]},
      {"account": account ref2,
        "rights": ["userParameters"]}]
      ],
    "securities": [
      {"account": account ref1,
        "rights": ["userParameters"]},
      {"account": account ref2,
        "rights": ["userParameters"]}]
      ],
    "consentType": "detailed"
    "recurringIndicator": true,
    "validTo": "2024-11-01",
  }
}
```

Response in case of a redirect

```
HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: REDIRECT
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: v2/consents/user-parameters-access/3d9a81b3-a47d-
4130-8765-a9c0ff861103"
Content-Type: application/json
```

```
{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861103",
  "_links": {
    "scaRedirect": {"href":
      "https://www.testbank.com/authentication/3d9a81b3-a47d-4130-8765-
      a9c0ff861103"},
    "status": {"href": "/v2/consents/user-parameters-access/3d9a81b3-a47d-
      4130-8765-a9c0ff861103/status"},
    "scaStatus": {"href": "v2/consents/user-parameters-access/3d9a81b3-
      a47d-4130-8765-a9c0ff861103/
      authorisations/123auth567"}
  }
}
```

Specific Example with detailed access data for IBAN check rights

Request

POST <https://api.testbank.com/v2/consents/user-parameters-access>
 Content-Type: application/json
 X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
 PSU-IP-Address: 192.168.8.78
 PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
 Gecko/20100101 Firefox/54.0
 Date: Sun, 06 Aug 2017 15:05:37 GMT

```
{"access": {
  "payments": /* payments
    [{"account": account ref1,
      "rights": ["ibanChecks", "psuName"]},
     {"account": account ref2,
      "rights": ["ibanChecks"]}],},
  "consentType": "detailed",
  "recurringIndicator": false,
  "validTo": "2024-11-01",
}}
```

Example without detailed access data

Request

POST <https://api.testbank.com/psd2/v2/consents/user-parameters-access>
 Content-Type application/json
 X-Request-ID 99391c7e-ad88-49ec-a2ad-99ddcb1f7756

PSU-IP-Address 192.168.8.78
 PSU-User-Agent Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
 Gecko/20100101 Firefox/54.0
 Date Sun, 06 Aug 2017 15:05:37 GMT

```
{"access": {
    "payments": /* payment accounts
        [{"rights": ["userParameters", "psuName"]}]
    ],
    "cards": [
        {"rights": ["userParameters"]}
    ],
    "savings": [
        {"rights": ["userParameters"]}
    ],
    "loans": [
        {"rights": ["userParameters"]}
    ],
    "securities": [
        {"rights": ["userParameters"]}
    ]
}
"consentType": "aspsspManaged",
"recurringIndicator": true,
"validTo": "9999-12-31",
}
```

Example for global PSU parameters access

Request

POST <https://api.testbank.com/psd2/v2/consents/user-parameters-access>

Content-Type application/json
 X-Request-ID 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
 PSU-IP-Address 192.168.8.78
 PSU-User-Agent Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
 Gecko/20100101 Firefox/54.0
 Date Sun, 06 Aug 2017 15:05:37 GMT

```
{"access": {
    "payments": [
        {"rights": ["userParameters"]}], /* in addition specific
elements like ownerName etc. could be added.
    "savings": [
        {"rights": ["userParameters"]}],
    "cards": [
        {"rights": ["userParameters"]}],
    "loans": [
        {"rights": ["userParameters"]}],
    "securities": [
        {"rights": ["userParameters"]}]
}
"consentType": "aspsspManaged",
"recurringIndicator": true,
"validTo": "9999-12-31",
}
```

```

    "securities": [
        {"rights": ["userParameters"]}
    ],
    "consentType": "global",
    "recurringIndicator": true,
    "validTo": "9999-12-31"
}

```

3.6.2 Get Consent Request

Call

`GET /v2/consents/user-parameters-access/{consentId}`

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

Path Parameters

Attribute	Type	Description
consentId	Max70Text	ID of the corresponding consent object as returned by a User Parameters Consent Request

Query Parameters

No specific query parameter.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based SCA was performed in the corresponding consent transaction or if OAuth2 has been used in a pre-step.

Request Body

No request body.

Response Code

HTTP Response Code equals 200.

Response Header

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

Response Body

Attribute	Type	Condition	Description
access	Extended Account Access	Mandatory	
consentType	Consent Type	Mandatory	
recurringIndicator	Boolean	Mandatory	
validTo	ISODate	Mandatory	
consentStatus	Consent Status	Mandatory	The status of the consent resource.

Example

Request

```
GET https://api.testbank.com/dv2/consents/user-parameters-access/3d9a81b3-a47d-4130-8777-a9c0ff861110
```

Response

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

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

3.7 Establish Consent on Document Services

In this section, the Establish Consent on Generic Access process is defined for the openFinance API Framework. A generic access could be different sorts of access, e.g. sending RTPs, sending documents or retrieving (one-time only) PSU identification related data or other PSU related data from the API. This step is sometimes also called "PSU activation"

in API Access Schemes. The actual access is defined only by the required access rights submitted with the consents.

NOTE: This process is not really comparable to other consent mechanisms. It is more an explicit confirmation by the PSU to receive e.g. related rtps including elnvoices via the online channels for the addressed sender/rtpsp service provider, still adding the possibility to retrieve certain PSU related information. For this reason, the consent object is not contained in the actual RTP API, in contrast to other consent categories – while it still is when retrieving data e.g. from the PSU parameter API.

3.7.1 Establish Consent on Document Services Request

Call

```
POST /v2/consents/document-services
```

Creates a consent on document services resource at the ASPSP regarding explicitly confirming the consent to submit RTPs to (a) dedicated account(s).

No Side Effects

In difference to the Establish Consent on Account Information as defined in Section 3.4.1, there is no side effect by the Establish Consent on Generic Access Request.

Query Parameters

No specific query parameter.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
PSU-IP-Address	String	Mandatory	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. If not available, the TPP shall use the IP Address used by the TPP when submitting this request.
Authorization	String	Conditional	If OAuth2 has been chosen as pre-step to authenticate the PSU.

NOTE: The data overview in Section 3.2 defines more request headers, like for PSU identification, PSU environment information, SCA Approach or explicit authorisation preferences of the TPP as well as redirect information. These parameters are described in detail in [oFA-PFSM] or in the related Open API files. These parameters apply to all

Transaction Initiation Requests, i.e. transactions, which need an authorisation within the openFinance API Framework and are left out here to achieve a better readability.

Request Body

Attribute	Type	Condition	Description
access	Account Access	Mandatory	requested access rights on account(s), where the document services are intended to be addressed to.
consentType	Consent Type	Mandatory	Only the consentTypes "detailed" and "aspssManaged" may be supported.
documentOriginator	Creditor Details And Account	Conditional	<p>The details of the sender of a document or the creditor of the related payments. Shall be used by the API Client, if the consent is on the pair (Originator-RTPSP, RTP-Originator) in case of RTP.</p> <p>RFU: This is to be defined later for the document postal services.</p>
documentTypes	Array of Document Type Code	Conditional	Might be mandated by API Access Schemes or bilateral contracts.
additionalRequestInformation	Additional Request Information1	Conditional	Is used only in case of an Originator-RTPSP-Aggregator as API Client.
validTo	ISODate	Mandatory	<p>This parameter is defining a valid until date (including the mentioned date) for the requested consent. The content is the local ASPSP date in ISODate Format, e.g. 2024-10-30.</p> <p>Future dates might get adjusted by ASPSP.</p> <p>If a maximum available date is requested, a date in far future is to be used: "9999-12-31".</p> <p>In both cases, the consent object to be retrieved by the GET Consent</p>

Attribute	Type	Condition	Description
			<p>Request will contain the adjusted date.</p> <p>The support of this adaption of the maximum available date is mandatory for the ASPSP.</p>
registrationInformation	Max140Text	Optional	Additional information about the registration process for the PSU, e.g. a reference to the TPP / PSU contract

This specification mandates the ASPSP to support the consentType "detailed" and/or "aspsspManaged". In case of the usage of a not supported consentType by the API Client, the ASPSP will use a dedicated Message Code in the error messaging. The following consentTypes have specific restrictions on the usage of access rights, as defined in the following. Please note, that discovery services will provide the related information, if supported by the ASPSP, cp. [oFA Disc].

- detailed

The access rights "submitRtps", "submitDocs", "psuName" and "psuIdentification" may be used by the API Client. The access right "submitRtps" and/or "submitDocs" shall be used as a minimum. The ASPSP may restrict this to the usage of one of these two attributes.

- aspsspManaged

The access rights "submitRtps", "submitDocs", "psuName" and "psuIdentification" may be used by the API Client. If no code is provided, the PSU will select the related code during authorisation.

Response Code

HTTP Response Code equals 201.

Response Header

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

Attribute	Type	Condition	Description
ASPSP-SCA-Approach	String	Conditional	<p>Possible values are:</p> <ul style="list-style-type: none"> • EMBEDDED • DECOUPLED • REDIRECT • SIGNATURE • ASPSP-CHANNEL <p>OAuth will be subsumed by the constant value REDIRECT</p>

Response Body

Attribute	Type	Condition	Description
consentId	String	Mandatory	Identification of the consent resource as it is used in the API structure
consentStatus	Consent Status	Mandatory	authentication status of the consent
scaMethods	Array of Authentication Objects	Conditional	<p>This data element might be contained, if SCA is required and if the PSU has a choice between different authentication methods. Depending on the risk management of the ASPSP this choice might be offered before or after the PSU has been identified with the first relevant factor, or if an access token is transported. If this data element is contained, then there is also a hyperlink of type "selectAuthenticationMethods" contained in the response body.</p> <p>These methods shall be presented towards the PSU for selection by the TPP.</p>
chosenScaMethod	Authentication Object	Conditional	This data element is only contained in the response if the ASPSP has chosen the Embedded SCA Approach, if the PSU is already identified with the first relevant factor or alternatively an access token, if SCA is required and if the authentication method is implicitly selected.
challengeData	Challenge	Conditional	It is contained in addition to the data element chosenScaMethod if challenge data is needed for SCA.

Attribute	Type	Condition	Description
			In rare cases this attribute is also used in the context of the startAuthorisationWithPsuAuthentication or startAuthorisationWithEncryptedPsuAuthentication link.
_links	Links	Mandatory	A list of hyperlinks to be recognised by the API Client. Type of links admitted in this response (which might be extended by single ASPSPs as indicated in its XS2A documentation) are defined in [oFA-PFSM] generically for all Transaction Initiation Response messages. The related Open API files will also contain all details.
psuMessage	Max512Text	Optional	Text to be displayed to the PSU, e.g. in a Decoupled SCA Approach

Example

Request

```
POST https://api.testbank.com/v2/consents/document-services
Content-Type: application/json
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7756
PSU-IP-Address: 192.168.8.78
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
Date: Sun, 06 Aug 2017 15:05:37 GMT

{
  "access": {
    "payments": /* payment accounts
      [{"account": account ref1,
        "rights": ["submitRtps", "psuName", "psuIdentifcation"]}]
    },
    ...
  },
  "consentType": "detailed",
  "validTo": "9999-12-31",
  "registrationInformation": "Your contract Number 1234 with MyMerchant is completed with the registration with your bank."
}
```

Response in case of a redirect

```

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: REDIRECT
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "v2/consents/document-services/3d9a81b3-a47d-4130-
8765-a9c0ff861104"
Content-Type: application/json

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861104",
  "_links": {
    "scaRedirect": {"href":
"https://www.testbank.com/authentication/3d9a81b3-a47d-4130-8765-
a9c0ff861104"},
    "status": {"href": "/v2/consents/document-services/3d9a81b3-a47d-4130-
8765-a9c0ff861104/status"},
    "scaStatus": {"href": "v2/consents/document-services/3d9a81b3-a47d-
4130-8765-a9c0ff861104/
authorisations/123auth567"}
  }
}

```

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

```

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: REDIRECT
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "v2/consents/document-services/3d9a81b3-a47d-4130-
8765-a9c0ff861104"
Content-Type: application/json

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861104",
  "_links": {
    "startAuthorisation": {"href": "v2/consents/document-services/3d9a81b3-
a47d-4130-8765-a9c0ff861104/authorisations"}
  }
}

```

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

```

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721

```

ASPSP-SCA-Approach: REDIRECT
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "v2/consents/document-services/3d9a81b3-a47d-4130-8765-a9c0ff861104"
Content-Type: application/json

```
{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861104",
  "_links": {
    "self": {"href": "/v2/consents/document-services/3d9a81b3-a47d-4130-8765-a9c0ff861104"},
    "scaStatus": {"href": "v2/consents/document-services/3d9a81b3-a47d-4130-8765-a9c0ff861104/authorisations/123auth567"},
    "scaOAuth": {"href": "https://www.testbank.com/oauth/.well-known/oauth-authorization-server"}
  }
}
```

Response in case of the decoupled approach

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: DECOUPLED
Date: Sun, 06 Aug 2017 15:05:47 GMT
Location: "v2/consents/document-services/3d9a81b3-a47d-4130-8765-a9c0ff861104"
Content-Type: application/json

```
{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861104",
  "_links": {
    "startAuthorisationWithPsuIdentification": {"href": "/v2/consents/document-services/3d9a81b3-a47d-4130-8765-a9c0ff861104/authorisations"}
  }
}
```

Response in case of the signature approach

HTTP/1.x 201 Created
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach: SIGNATURE
Body-Sig-Profile-Accepted: JAdES_JS
Date: Sun, 06 Aug 2017 15:05:47 GMT

```

Location:           "v2/consents/document-services/3d9a81b3-a47d-
4130-8765-a9c0ff861104"
Content-Type:       application/json

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861104",
  "_links": {
    "startAuthorisationWithPsuIdentification": {"href":
"/v2/consents/document-services/3d9a81b3-a47d-4130-8765-
a9c0ff861104/authorisations"}
  }
}

```

Response in case of the embedded approach

```

HTTP/1.x 201 Created
X-Request-ID:          99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:   EMBEDDED
Date:                  Sun, 06 Aug 2017 15:05:47 GMT
Location:              "v2/consents/document-services/3d9a81b3-a47d-4130-
8765-a9c0ff861104"
Content-Type:           application/json

{
  "consentStatus": "received",
  "consentId": "3d9a81b3-a47d-4130-8765-a9c0ff861104",
  "_links": {
    "startAuthorisationWithPsuAuthentication": {"href":
"/v2/consents/document-services/3d9a81b3-a47d-4130-8765-
a9c0ff861104/authorisations"}
  }
}

```

3.7.2 Get Consent Request

Call

GET </v2/consents/document-services/{consentId}>

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

Path Parameters

Attribute	Type	Description
consentId	Max70Text	ID of the corresponding consent object as returned by an RTP Consent Request

Query Parameters

No specific query parameter.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based SCA was performed in the corresponding consent transaction or if OAuth2 has been used in a pre-step.

Request Body

No request body.

Response Code

HTTP Response Code equals 200.

Response Header

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

Response Body

Attribute	Type	Condition	Description
access	Account Access	Mandatory	Account access details, where the confirmation of funds service is aimed to be submitted to.
consentType	Consent Type	Mandatory	



Attribute	Type	Condition	Description
documentOriginator	Creditor Details And Account	Optional	The details of the sender of a document or the creditor of the related payments. Shall be used by the API Client, if the consent is on the pair (Originator-RTPSP, RTP-Originator) in case of RTP. RFU: This is to be defined later for the document postal services.
documentTypes	Array of Document Type Code	Optional	The consent might be restricted to dedicated document types.
additionalRequestInformation	Additional Request Information1	Optional	Is used only in case of an Originator-RTPSP-Aggregator as API Client.
registrationInformation	Max140Text	Optional	Additional registration information.
consentStatus	Consent Status	Mandatory	The status of the consent resource.

Example

Request

```
GET https://api.testbank.com/v2/consents/document-services/3d9a81b3-a47d-4130-8777-a9c0ff861110
```

Response

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

```
{
  "access": {
    "payments": [
      "account": {"iban": "DE40100100103307118608"},
      "accessRights": {"submitRtps"}
    ],
    "consentType": "detailed",
    "consentStatus": "valid"
  }
}
```

3.8 Get Consent Status Request

Call

```
GET /v2/consents/{consent-category}/{consentId}/status
```

Retrieves the status of an addressed consent resource.

Path Parameters

Attribute	Type	Description
consent-category	String	The related consent type. Currently supported are the following consent types: <ul style="list-style-type: none"> • account-access, • funds-confirmations • user-parameters-access • document-services
consentId	String	The consent identification assigned to the created resource.

Query Parameters

No specific query parameters defined.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based SCA was performed in the corresponding consent transaction or if OAuth2 has been used in a pre-step.

Request Body

No request body.

Response Code

HTTP Response Code equals 200.

Response Header

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

Response Body

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

Example

Request

```
GET https://api.testbank.com/psd2/v2/consents/account-access/3d9a81b3-a47d-4130-8765-a9c0ff861101/status
X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date: Sun, 06 Aug 2017 15:05:46 GMT
```

Response

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

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

3.9 Cancellation of a Consent

The API Client can cancel a consent object if needed with the following call:

Call

```
DELETE /v2/consents/{consent-category}/{consentId}
```

Terminates a given consent.

Path Parameters

Attribute	Type	Description
consent-category	String	Equals one of the strings <ul style="list-style-type: none"> • account-access • funds-confirmations • user-parameters-access • document-services
consentId	String	Contains the resource-ID of the consent to be deleted.

Query Parameters

No specific query parameters.

Request Header

Attribute	Type	Condition	Description
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Conditional	Is contained only, if an OAuth2 based SCA was performed in the corresponding consent transaction or if OAuth2 has been used in a pre-step.

Request Body

No Request Body.

Response Code

HTTP Response Code equals 204.

Response Header

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



Response Body

No Response Body

Example

Request

```
DELETE https://api.testbank.com/psd2/v2/consents/account-access/3d9a81b3-a47d-4130-8765-a9c0ff861101
```

X-Request-ID 99391c7e-ad88-49ec-a2ad-99ddcb1f7757

Date Sun, 13 Aug 2017 17:05:37 GMT

Response

HTTP/1.x 204 No Content

X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7757

Date: Sun, 06 Aug 2017 15:05:47 GMT

4 References

4.1 Documents of the NextGenPSD2 API Framework

[XS2A MCon] NextGenPSD2 XS2A Framework, Implementation Guidelines, Extended Services, Multiple Consents Service, Version 1.0, 30 October 2020

[XS2A-CCF] NextGenPSD2 XS2A Framework, Implementation Guidelines, Extended Services, Confirmation of Funds Consent, Version 2.0, 01 March 2019,

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

4.2 Documents of the openFinance API Framework

[oFA-IG-ADM] openFinance API Framework, Operational Rules for Administrative Service, Version 1.1, 9 September 2024

[oFA-PFSM] openFinance API Framework, Protocol Functions and Security Measures, Version 2.3, 31 October 2025

[oFA DomDef] openFinance API Framework, Domestic AIS/PIS definitions, current version

[oFA Disc] openFinance API Framework, Discovery Services, Version 1.1, 24 January 2025

[oFA DaD] openFinance API Framework, Data dictionary for V2 services, Version 2.3, 31 October 2025



4.3 Further documents

- [EBA-FR] Final Report, Draft Regulatory Technical Standards, amending Commission Delegated Regulation (EU) 2018/389 supplementing Directive (EU) 2015/2366 of the European Parliament and of the Council with regard to regulatory technical standards for strong customer authentication and common and secure open standards of communication, published 5 April 2022
- [EBA-OP2] Opinion of the European Banking Authority on obstacles under Article 32(3) of the RTS on SCA and CSC, EBA/OP/2020/10, published 4 June 2020
- [EBA-RTS] Commission Delegated Regulation (EU) 2018/389 of 27 November 2017 supplementing Directive 2015/2366 of the European Parliament and of the Council with regard to Regulatory Technical Standards for Strong Customer Authentication and Common and Secure Open Standards of Communication, C(2017) 7782 final, published 13 March 2018
- [EBA RTSA] COMMISSION DELEGATED REGULATION (EU) 2022/2360 of 3 August 2022, amending the regulatory technical standards laid down in Delegated Regulation (EU) 2018/389 as regards the 90-day exemption for account access, 05 December 2022
- [eIDAS] Regulation (EU) No 910/2014 of the European Parliament and of the Council on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market, 23 July 2014, published 28 August 2014
- [ETSI TS 119 182-1] ETSI Technical Specification, Electronic Signatures and Infrastructures (ESI); JAdES digital signatures; Part 1: Building blocks and JAdES baseline signatures, V1.1.1 (2021-03)
- [ETSI TS 119 495] ETSI Technical Specification, Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Certificate Profiles and TSP Policy Requirements for Open Banking, V1.5.1 (2021-04)
- [FAPI-CBPIA] OpenID Foundation, Financial-grade API (FAPI) Working Group, Cross-Browser Payment Initiation Attack, https://bitbucket.org/openid/fapi/src/master/TR-Cross_browser_payment_initiation_attack.md, 3.01.2019
- [HAL] Kelley, M., "HAL - Hypertext Application Language", 2013-09-18, http://stateless.co/hal_specification.html
- [OA-SecTop] OAuth 2.0 Security Best Current Practice draft-ietf-oauth-security-topics-13, Lodderstedt et al., 8 July 2019, <https://tools.ietf.org/html/draft-ietf-oauth-security-topics-13>
- [OBEsign] Open Banking Europe: JSON Web Signature Profile for Open Banking, Version 001-001, 18.05.2021, <https://www.openbankingeurope.eu/media/2095/obe-json-web-signature-profile-for-open-banking.pdf>

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



5 Annex A: Change Log

5.1 Changes from Version 2.0 to Version 2.1

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

Section	Change	Reason
All sections	The attribute consentId has been restricted to max70Text	General approach for resource identifications.
All sections	In some cases, the access right code "balance" was used. This is corrected to "balances", also in many examples.	Errata
3.2	Rename TPP-SCA-Preference to Client-SCA-Preference. Rename TPP-Explicit-Authorisation-Preferred to Client-Explicit-Authorisation-Preferred Rename TPP-Redirect-URI to Client-Redirect-URI header.	Make these parameters also usable for direct access models for corporates. Prepare the API Framework better for corporate direct access.
3.4.1	Extended the example with card-accounts attribute	The data dictionary is now distinguishing between single cards and card accounts.
Several Sections	Add the value ASPSP-CHANNEL to the ASPSP-SCA-APPROACH header.	Adding asynchronous SCA via ASPSP channels explicitly to the framework resulting from new services.
3.7.2	Some attributes have been changed in the response body.	Erratum, copy and paste error from funds-confirmations consent category.
3.8, 3.9	Consent category "rtps" has been replaced by "document-services" at several occurrences.	Erratum, consent category had been generalised from rtp to documents before already, but some replacements had been overseen.

5.2 Changes from Version 2.1 to Version 2.2

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

Section	Change	Reason
3.4.1, 3.5.1, 3.7.1	Added new “signature” ASPSP-SCA-Approach	Adapting to new framework specifications
3.6.1	Added the IBAN check service and the possible consent types for this service.	Addition of the consent type for the IBAN check service as a specific user parameters Access
3.6.1	Renaming the access right “psuParameters” to “userParameters”	Adapting to framework changes
4	Updated References	To be in line with the framework documentation

