THE Berlin GROUP

A EUROPEAN STANDARDS INITIATIVE

openFinance API Framework

Operational Rules

Consent API for V2.x

Version 2.0

31 October 2025

License Notice

[*] 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 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 of Funds Service (COF) 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:

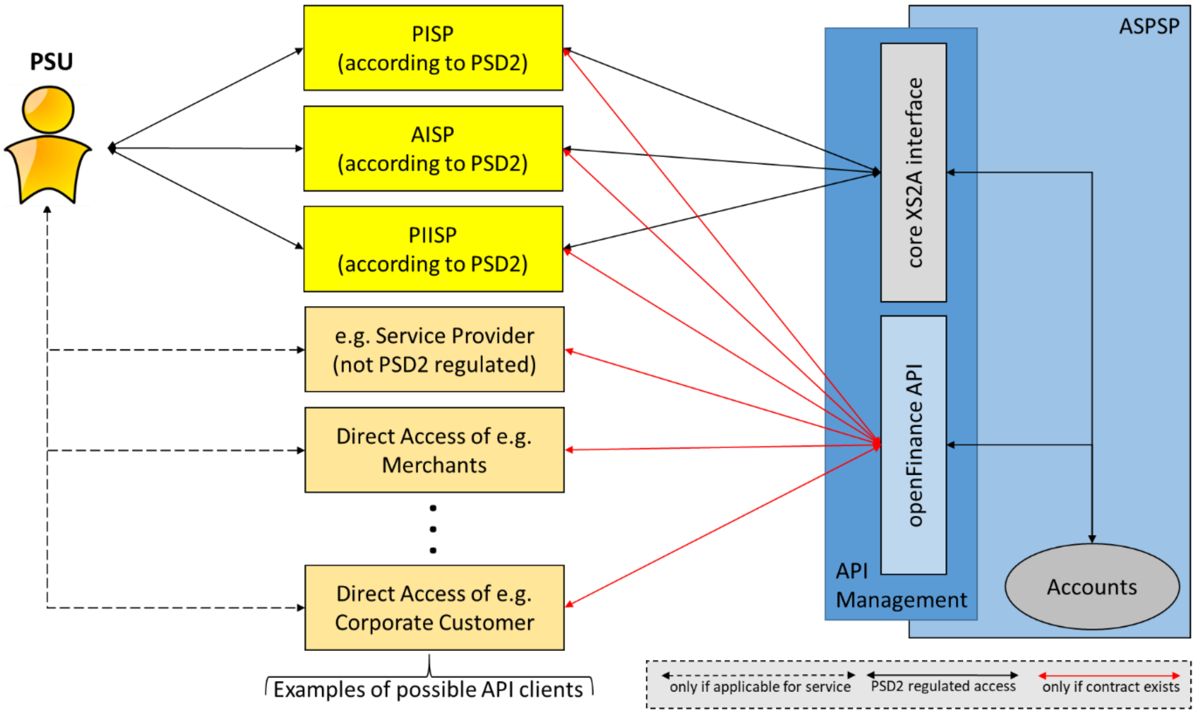
PSU
PISP (according to PSD2)
AISP (according to PSD2)
PIISP (according to PSD2)
e.g. Service Provider (not PSD2 regulated)
Direct Access of e.g. Merchants
Direct Access of e.g. Corporate Customer
Examples of possible API clients
ASPSP
core XS2A interface
openFinance API
API Management
Accounts
only if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

Figure 1: Core XS2A interface and openFinance API

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 e.g. access sensitive PSU data via the API. The legal text [PSD2] and [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

In general, the PSU's consent is crucial for the API Client for any kind of interaction at the API. It is arranged between the PSU and the API Client. For this, there is a distinction between implicit and explicit consent.

PSU consent for some services like payment initiations is worked out implicitly when authorising the payment as such, since the submission of the related data (e.g. payment data) and the related authorisation via Strong Customer Authentication (SCA) is worked out at the same time in one session. So, no specific consent object or related token for later access is mandated in that case. Only when having chosen OAuth as SCA approach, a technical authorization token might still be needed even in that case, cf. Section 8.8 in [oFA-IG-PFSM].

Another example where the consent is given implicitly is a request to pay service. For this, the receiver of a RTP message has to give its consent to the originator of the RTP beforehand in order to receive RTPs form the originator. There is no mandated requirements on the form of the related consent (it could be verbal consent or click consent, etc.) for this standard. The ASPSP might offer the SCA approach, but no consent token is created. The consent is required for the service but not verified explicitly on API level. Thus, for RTP services, the consent is given implicitly, as well.

For other services, the process of establishing consent for a service and executing the respective service is strictly divided into two phases. Due to this separation, the consent needs to be verified and becomes a resource which depending on the service may be needed for the second step 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. Consent established by necessary usage of the consent API is defined to be *explicit consent*. In contrast, consent which does not rely on the consent API in order to be established is *implicit consent*.

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 AISP, 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 [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 so-called *consent categories*.

1.2.3 Consent Token

For account information services, confirmation of funds services and user parameter information services the consent is used as a token in the following way: At the end of a successful authorisation process within the consent API, the API Client uses the related consent resource identification as a token towards the respective openFinance APIs. In this case, the API Client needs to store the token in its system for the whole lifetime of the token.

Note: In the case that the consent was verified by an OAuth2 SCA approach, the ASPSP might in addition offer a bearer token following the OAuth2 standard. Then this second token needs to be used by the API Client in addition when accessing the related APIs. If the API Client is addressing the related API without the related tokens, 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-IG-PFSM]. Further, complex data type definitions and code lists from [oFA DaD] are used within this specification without further reference.

1.3 Document Structure

This document specifies the operational rules for the Consent API Framework in context of establishing consent, e.g. used for account information services. It has to be considered with any other operational rules of a service that uses explicit consent to verify the authorisation of the PSU. In this document the following topics are presented:

- The basic concepts for the consent API in Section 2,
- An introduction of the actors who access the API and their respective roles is presented in Section 3,
- In Section 4, the API services for establishing consents are presented,
- The operational rules for these concepts in Section 5, and
- The message and data model on which the exchange at the consents API relies on is given in Section 6.

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
V2.0	First released version of the operational rules of the V2 consent API	31 October 2025

2 Basics for the Consent API

2.1 Services at the XS2A interface and the openFinance API

At the XS2A interface, the TPP is required by the [PSD2] to verify and prove the PSU's consent in order to execute the respective core service. The openFinance API supports different services, most of which rely on verifying the PSU's consent in order to execute the service as well. At both interfaces, the consent may be given implicitly or explicitly, depending solely on the respective service.

This openFinance API standard is not defining whether a service is to be supported in the openFinance API or not, but it might define a mandatory support of sub-services once a dedicated openFinance API service is offered.

2.2 Services covered in this document

In the following overview, the different categories of consent supported at the openFinance consent API are presented:

Consent Categories	Description
Access to accounts	An API Client gains consent to access the addressed accounts and retrieve account information.
Confirmation of funds	An API Client gains consent to access a dedicated account for the funds-confirmations service.
Access to user parameters	An API Client gains consent for the access of user data for addressed accounts.
Document services	The ASPSP offers SCA approaches to support the PSU – API Client consent process on submitting documents (e.g. RTPs) to the PSU account in the ASPSP's system.

Table 1: Access methods of an API Client at the openFinance API

In contrast to e.g. a payment service, where the service is completed by default when the respective payment is executed, an establishing consent service relying on a consent resource does not have such definite end point pre-defined by its service structure. In theory, the consent may be valid as long as no cancellation of the consent took place. Thus, to avoid an indefinite lifespan of the consent, the consent resource is linked to a certain lifecycle. The stages of the lifecycle are defined by different status of the consent and are transparent for the API Client and the ASPSP:

- During the initiation phase of a consent service, the API Client receives current information about the status within all response messages of the authorisation process, the possible status are
  - Received: the consent request has been received by the ASPSP, is technically correct and will be processed further, i.e. for an authorisation,
  - Rejected: after the request was rejected, e.g. because no (successful) authorisation took place,
  - partially authorised: a multi-level authorisation is needed, some but not all authorisations have been performed yet, or
  - valid: the ASPSP accepts the consent, the authorisation was successfully completed.
- After a successful authorisation of the consent request by the PSU, i.e. the consent status is valid, the status might change again and this must be transparent for the API Client.
- Any kind of consent service at the openFinance API is only valid to a certain date, its expiry date. The following four scenarios are possible how the consent resource becomes no longer valid. In these cases, the status of its resource indicates which scenario occurred:
  - expired: in case that the expiry date of the consent resource is reached while the status of the consent is still "valid",
  - revoked by PSU: the PSU cancels the consent explicitly, using an online channel of the ASPSP,
  - terminated by API Client: similar to the above, the API Client cancels the consent resource explicitly using the API, or
  - replaced by API Client: if the PSU requests the same consent service, while the current service status is still "valid", the current service is cancelled implicitly and substituted by the recurring one, if applicable for the addressed consent category.

2.3 Multi-Currency Account Specifics for Consents

2.3.1 Submission for Consents on Accessing Accounts

When addressing a multi-currency account, this follows the same procedure as a single-currency account but without specifying a corresponding currency. Then asking for consent to retrieve any kind of account information, implies getting information for all sub-accounts.

On the other hand, a single 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, the API Client is only able to access the sub-accounts corresponding to this currency and no consent is provided for accessing the other sub-accounts.

2.3.2 Submission for Consent on Confirmation of Funds

The currency of the account can be addressed optionally when submitting the consent request. Thus, if the currency is not contained, a sub-account is addressed, which is defined as default sub-account by the ASPSP.

3 Actors and Roles

3.1 Third party processor (TPP) related scenario

In general, services offered by an ASPSP at its openFinance API may be accessed/used not only by clients registered by an NCA in the role of a TPP according to the [PSD2] regulation but by any third party. Nevertheless, for Extended Payment Initiation Services or Extended Account Information Services this might still apply for many subservices due to the current regulation.

Actors and roles of related parties in a scenario where the extended service is defined via a TPP are described in the underlying basic operational rules for the framework documents, see [oFA-OR-Com] and [oFA-OR-FW].

3.2 Direct access scenario

The openFinance API Framework now develops specification to reuse the TPP – ASPSP openFinance API also as an PSU – ASPSP interface, at least for the corporate case, where the broad functionality like multi-signing etc. applies. The direct access potentially addresses all (business) services defined in the openFinance API Framework. The technical API client system in this case is either a client software hosted by the corporate itself or by e.g. an ERP cloud provider.

For simplicity reasons, if no further distinction is necessary, the party accessing the API will be referred to as "API Client" in this document.

4 API Services supported for the Establish Consent Services

The current version of the Consent Services supports the following API services. The Service Type is indicating a bundling of several API services to a service family. This document introduces the following service types:

- Establish consent on account information at the XS2A interface for accounts/account types regulated by the [PSD2] (CONS-AIS),
- Establish consent on extended account information at the openFinance API for accounts and account types, that are not (necessarily) regulated by the [PSD2] (CONS-XAIS),
- Establish consent for confirmation of funds at the openFinance API (CONS-COF),
- Establish consent on user parameter information at the openFinance API (CONS-USP), and
- Establish consent for document services at the open Finance API (CONS-DOC).

In addition, the openFinance API will support technical use cases within the RESTful API approach that are not necessarily used within the above-mentioned use cases, e.g. to read details on payment objects or other created resources.

The following table gives an overview of the API services:

API Services	Service Type	Technical Functionalities	PSU directly involved
Establish consent on account information (regulated by the [PSD2])	CONS-AIS	Establishes consent for retrieving account information for addressed account(s), Defines the corresponding access rights to the addressed account(s), If a recurring access to the account already exists, it will get deleted automatically and replaced by the recurring one.	yes
Establish consent on (extended) account information (not regulated by the [PSD2])	CONS-XAIS	Establishes consent for accessing account information for addressed account(s), Defines access rights to the addressed account(s), If a recurring access to the account already exists, it will get deleted	yes

API Services	Service Type	Technical Functionalities	PSU directly involved
		automatically and replaced by the recurring one	
Establish consent for confirmation of funds	CONS-COF	Establishes consent for confirmation of funds for the specified account(s)	yes
Establish consent on user parameter information	CONS-USP	Establishes consent for the access to technical user parameters related to given accounts by a PSU, or related to a given PSU respectively. If a recurring access to the user parameters already exists, it will get deleted automatically and replaced by the recurring one	yes
Establish consent for document services	COS-DOC	Establishes consent which allows the API Client to send documents for an account of the PSU and to retrieve potential PSU related information data	yes
Cancellation of a consent	CONS-AIS, CONS-XAIS, CONS-COF, CONS-USP, CONS-DOC	Terminates the addressed consent explicitly	no

Table 2: API services of the Consent Services

In addition, the openFinance API will support technical use cases within the RESTful API.

Remark: PSUs are always directly involved in establishing the consent. After the consent is recorded, the API Client may use the consent resource on its own in order to retrieve the allowed information within the limits set during the authentication.

4.1 API Service: Establish consent on account information

The support of this API Service at the XS2A interface is mandatory and the support at the openFinance API is recommended.

In the case of the core account information service at the XS2A interface, a TPP can only request information about payment accounts and card (reconciliation) accounts. On the other hand, in the case of the premium account information service, an API Client may request access information about payment accounts, cards, card-accounts, savings accounts, loans accounts, and/or securities accounts.

Note: If the API Client requests a recurring access to the account data and there exists already a former consent for recurring access to account information for the addressed PSU and API Client, then the former consent automatically expires as soon as the new consent request is authorised by the PSU. This has been introduced to manage consent token and account access in a coherent way.

An API Client may execute the consent service to receive the right to execute account information services in the future without necessarily needing the PSU's direct involvement for this. Subject to consent of the PSU, the API Client can obtain the rights for following services (of the account information service) for payment accounts, single card entries, card accounts, savings accounts, loans accounts, or securities accounts:

- Get the list of addressable accounts of the PSU once,
- Get details for a list of accounts/cards once or multiple times,
- Get the balances for a list of accounts once or multiple times,
- Get payment transaction information for a list of accounts once or multiple times.

Note: For the services at the XS2A interface the rights for services may only be obtained for payment accounts and cards, as the other account types are not supported at the interface.

In addition to the rights for services listed above, the following further rights for services can be obtained by the API Client at the openFinance API for the respective accounts:

- Payment and securities accounts: Get transaction details for a list of addressable accounts once or multiple times,
- Securities accounts: Get account positions for a list of addressable accounts once or multiple times,
- Securities accounts: Get securities order list for a list of addressable accounts once or multiple times,

If the API Client is granted the right to access balance or payment transaction information for certain accounts, this will include automatically the right to retrieve detailed information about the related payment accounts.

If the API Client is granted the right to execute an access to account information multiple times, the validity period of the right in days or the maximal period offered by the ASPSP are defined. It is furthermore possible to define the permitted frequency of corresponding services (per day). For the core services at the XS2A interface, the requirements of [PSD2] and [RTS] shall be observed during the entire validity period granted and for the allowed frequency of services. For the premium services, this may be regulated by restrictions given by the ASPSP and general regulations.

The ASPSP can offer optionally that the API Client submits only the account information type which has been agreed on between API Client and PSU. The ASPSP then will involve the PSU into the selection of the corresponding accounts. This option is not supported in the embedded SCA approach, cp. [oFA-OR-FW].

In case of multicurrency accounts, the account access can be granted on multicurrency account level as well as on sub-account level. The API Client is steering this by submitting the corresponding request, addressing the multicurrency account or sub-account level.

The following figure shows only the very top level information flow:

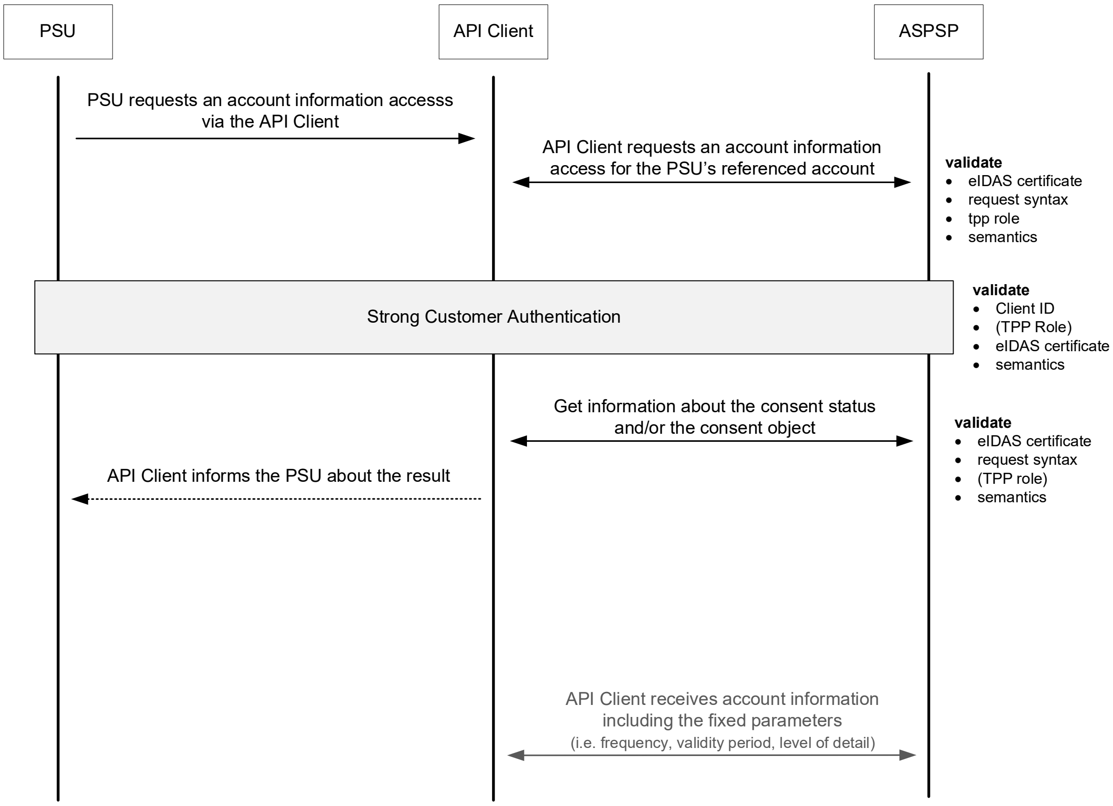
PSU
API Client
ASPSP
PSU requests an account information accesss via the API Client
API Client requests an account information access for the PSU's referenced account
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
Strong Customer Authentication
validate
• Client ID
• (TPP Role)
• eIDAS certificate
• semantics
Get information about the consent status and/or the consent object
validate
• eIDAS certificate
• request syntax
• (TPP role)
• semantics
API Client informs the PSU about the result
API Client receives account information including the fixed parameters
(i.e. frequency, validity period, level of detail)

Figure 2: Establish consent for account information

In general, while the service at the API is initiated by the API Client, it must first be initiated by the PSU at the PSU – API Client interface. The PSU – API Client interface is not within the scope of this document. However, the API Client has to inform the PSU clearly about the rights for which the PSU has to confirm its consent.

The API Client then sends an account information access request to the ASPSP. Within the request, the following is included:

- The requested access rights on accounts, where the account information service is aimed to be submitted,

  - for AIS at the XS2A interface the access for the requested payment account(s) or card (reconciliation) accounts or

  - for XAIS at the openFinance API the access for the requested account(s) which may be payment accounts, (single) cards, savings accounts, loans accounts, and/or securities accounts.

- The consent type that describes the level of information the API Client requests to access with this service request. The different possible consent types are present in detail in Subsection 6.2.1.1.1,

- The API Client may request the consent for a recurring access or for a one-time access, the maximum frequency per day for an access is included (after that the PSU's involvement is needed to access the account again), and in the case that a one-time access is requested, this frequency is set to one, and

- The date until which the consent is valid.

The ASPSP checks if the request is valid and might adjust the validity period and/or the frequency to access the account information per day. The ASPSP will reject the service if the API Client cannot be identified correctly API. In the case of a TPP at the XS2A interface, the ASPSP will reject the service if the TPP does not have the role AISP.

If the requested consent type is not supported by the ASPSP, the ASPSP will use a dedicated message code in the error messaging.

In the next step, the PSU authorises the access towards the ASPSP by SCA. The strong customer authentication might be exempted by the ASPSP. The API Client might request information about the consent status, in order to supervise the process of the SCA and conduct further information about the consent resource, in particular if the ASPSP adjusted parameters or not. If the initiation process of the account information access was successful, the API Client informs the PSU about it.

After the authorisation process is completed, a consent token is created which must be saved by the API Client and can be used to access the validated account information within the fixed parameters without further involvement of the PSU. The usage of the consent is no longer part of the consent-establishing service but a different service.

4.2 API Service: Establish consent for confirmation of funds

With this service, the API Client and the ASPSP establish consent which enables the API Client to access information about confirmation of funds of the specified account(s).

The following figure shows only the very top level information flow:

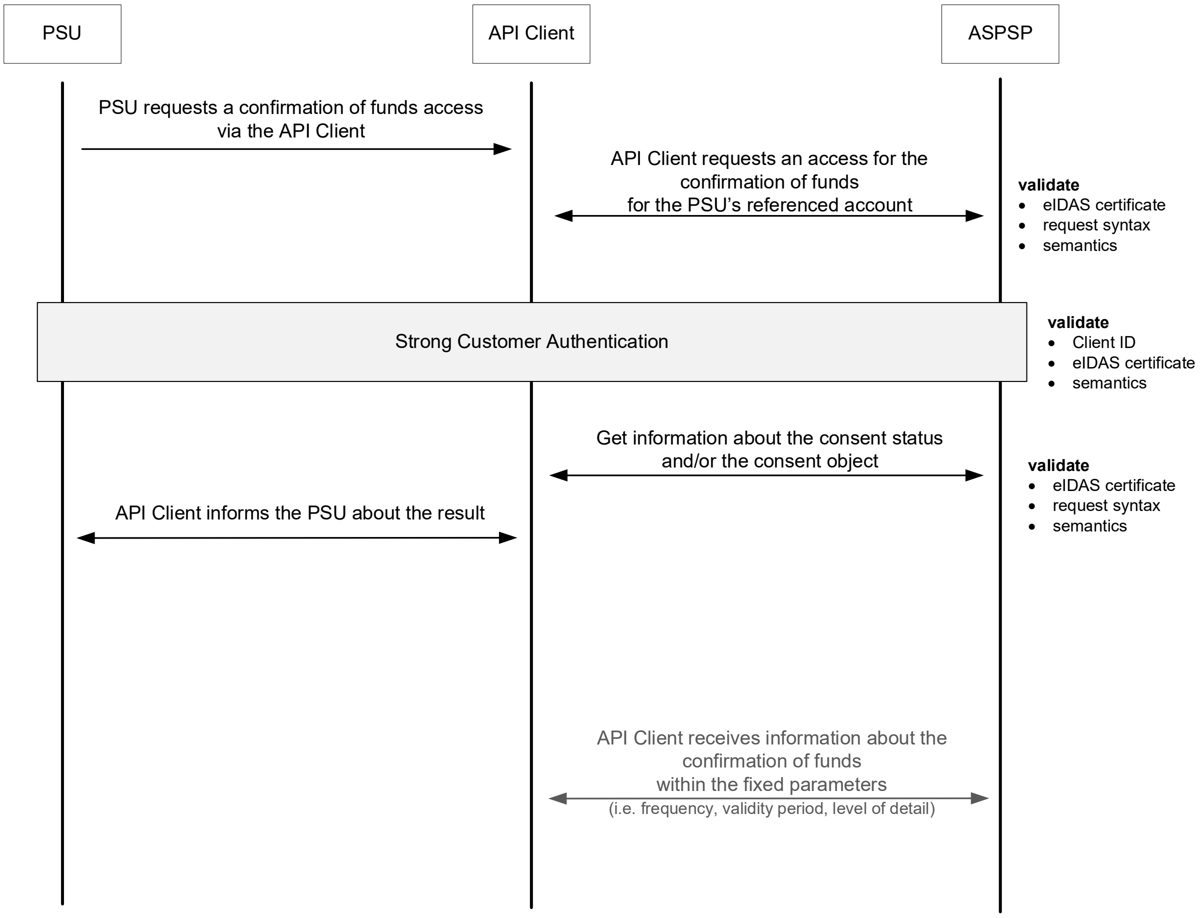
PSU
API Client
ASPSP
PSU requests a confirmation of funds access via the API Client
API Client requests an access for the confirmation of funds for the PSU's referenced account
validate
• eIDAS certificate
• request syntax
• semantics
Strong Customer Authentication
validate
• Client ID
• eIDAS certificate
• semantics
Get information about the consent status and/or the consent object
validate
• eIDAS certificate
• request syntax
• semantics
API Client informs the PSU about the result
API Client receives information about the confirmation of funds within the fixed parameters
(i.e. frequency, validity period, level of detail)

Figure 3: Establish consent for confirmation of funds service

The PSU requests that the API Client gains access to the confirmation of funds service. Then, the API Client sends a consent request to the ASPSP. In this request, the following is included:

- Access: the requested access rights on payment accounts, for which the confirmation of funds service is aimed to be submitted,
- The corresponding consent types yields the level of detail regarding information that may be accessed with this request. A list which consent types are supported for the Confirmation of Funds service is in Subsection 6.2.1.1.2,
- Indicator if the access is recurring or a one-time access to the account data, and

- Validity period: given by a valid-to-date, until which the access is requested.

The ASPSP checks if the request is valid and might adjust the validity period. If the requested consent type is not supported by the ASPSP, the ASPSP will use a dedicated message code in the error messaging.

In the next step, the PSU authorises the access towards the ASPSP by SCA. The strong customer authentication might be exempted by the ASPSP. The API Client might request information about the consent status, in order to supervise the process of the SCA and conduct further information about the consent resource, in particular if the ASPSP adjusted parameters or not. If the initiation process of the access to confirmation of funds was successful, the API Client informs the PSU about it.

After the authorisation process is completed, a consent token is created and transmitted to the API Client. The API Client must store the token for its whole lifespan and use it within requests asking for confirmation of funds. Then the service corresponding to the validated account and within the validity period is executed without further involvement of the PSU. The usage of the token is not part of the service of establishing consent on confirmation of Funds but a separated service on its own.

4.3 API Service: Establish consent on user parameter information

User parameters are set to be user related technical access data like access rights and payment authorisation limits which are needed for the API Client to 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 of corporates. It can also be used for retrieving this sort of information on PSUs for Third Parties, e.g. in account check services.

Note: If the API Client requests a recurring access to the user parameter data of an account and there exists already a former consent for recurring access to the user parameter of this account for the addressed PSU and API Client, then the former consent automatically expires as soon as the new consent request is authorised by the PSU.

The following figure shows only the very top level information flow:

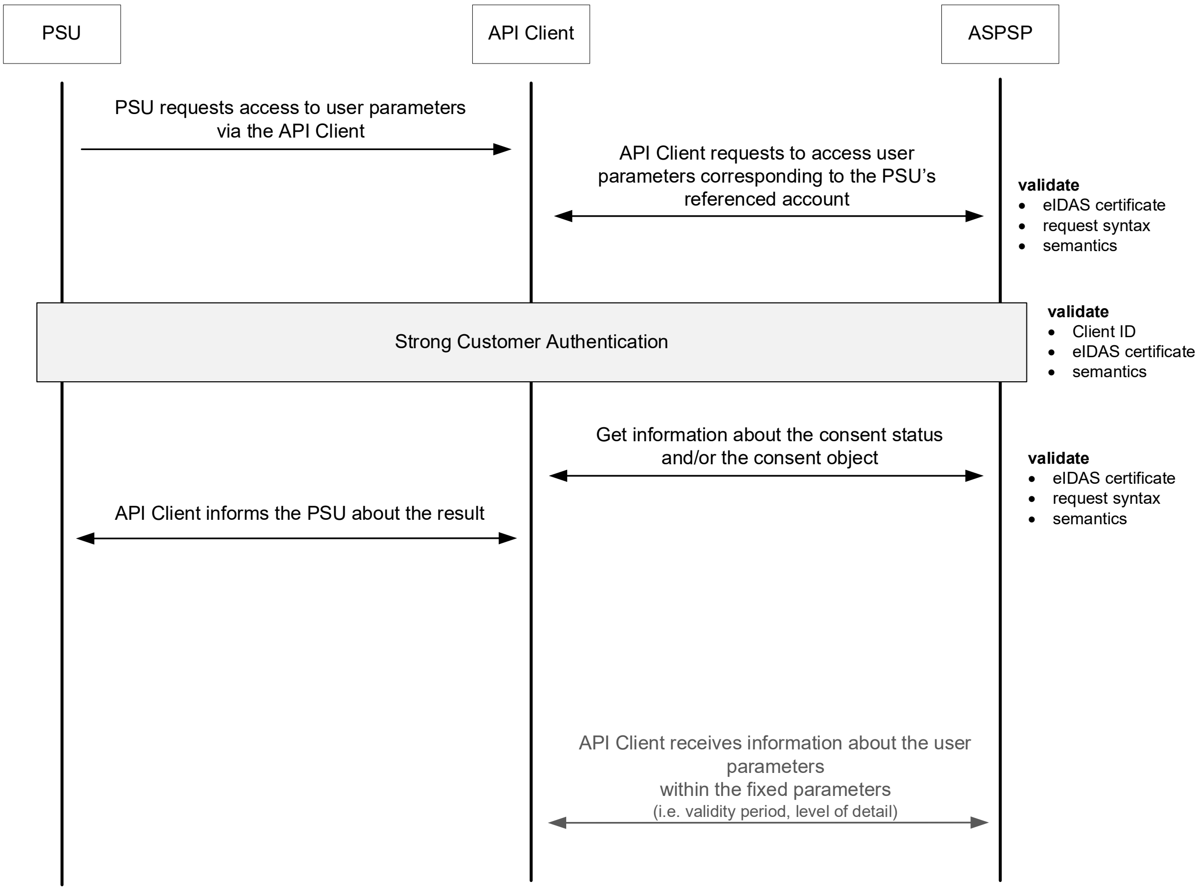
PSU
API Client
ASPSP
PSU requests access to user parameters via the API Client
API Client requests to access user parameters corresponding to the PSU's referenced account
validate
• eIDAS certificate
• request syntax
• semantics
Strong Customer Authentication
validate
• Client ID
• eIDAS certificate
• semantics
Get information about the consent status and/or the consent object
validate
• eIDAS certificate
• request syntax
• semantics
API Client informs the PSU about the result
API Client receives information about the user parameters within the fixed parameters
(i.e. validity period, level of detail)

Figure 4: Establish consent on Access to user parameter service

In the first step, the PSU requests that the API Client gains access to the user parameters of the respective account(s).

The API Client requests to access the user parameters for the account(s) of the PSU. The request includes the following details:

- the requested access for payment accounts, for which the access to user parameters service is aimed to be submitted,
- The Consent type yields the corresponding level of information for the access of user parameters service request. A list which consent types are supported is included in subsection 6.2.1.1.3,
- Indicator if the access is recurring or a one-time access to the account data, and
- A requested expiry date, until which the access is requested.

The ASPSP checks if the request is valid and might adjust the validity period. If the requested consent type is not supported by the ASPSP, the ASPSP will use a dedicated message code in the error messaging.

In the next step, the PSU authorises the access towards the ASPSP by SCA. The strong customer authentication might be exempted by the ASPSP. The API Client might request information about the consent status, in order to supervise the process of the SCA and conduct further information about the consent resource, in particular if the ASPSP adjusted parameters or not. If the initiation process of the access to the user parameters was successful, the API Client informs the PSU about it.

After the authorisation process is completed, the API Client receives a consent token allowing it to access the user parameters of the corresponding account and within the validity period without further involvement of the PSU. This is usage of the token is not part of consent establishing service, but a service on its own.

4.4 API Service: Establish Consent on document service

Using this service, the API Client gains consent of the PSU which allows the management of documents for the PSU, i.e. the API Client is allowed to send via a mailbox associated to an account of the PSU and to retrieve potentially PSU related information data. The latter feature might ensure the API Client that the documents are sent to the correct addressee. An example for this are Request-to-pay Requests including an invoice document which can be send by the API Client to the corresponding receiver.

The following figure shows only the very top level information flow:

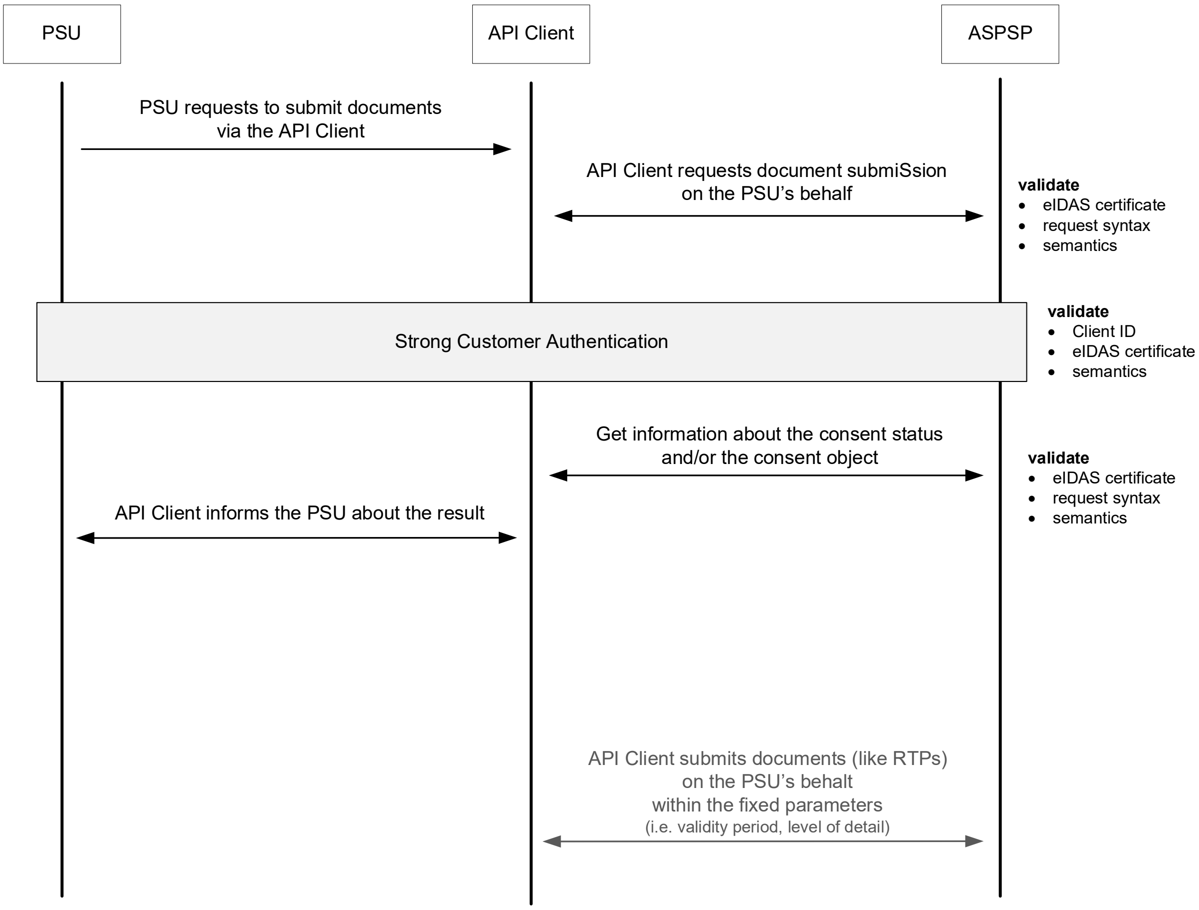
PSU
API Client
ASPSP
PSU requests to submit documents via the API Client
API Client requests document submiSsion on the PSU's behalf
validate
• eIDAS certificate
• request syntax
• semantics
Strong Customer Authentication
validate
• Client ID
• eIDAS certificate
• semantics
Get information about the consent status and/or the consent object
validate
• eIDAS certificate
• request syntax
• semantics
API Client informs the PSU about the result
API Client submits documents (like RTPs) on the PSU's behalt within the fixed parameters
(i.e. validity period, level of detail)

Figure 5: Establish Consent for Document service

In the first step, the PSU requests the API Client to send documents corresponding to a payment account of the PSU.

The API Client requests to access the user parameters for the account(s) of the PSU. This includes the following details:

- Access: the requested access rights on payment accounts, for which the access to the document service is aimed to be submitted,
- The consent type yields the level of detail regarding information that may be accessed with this request. A list which consent types are supported for the Documents service is in Subsection 6.2.1.1.4,
- Details of the documents which shall be submitted: information about the sender of a document or the creditor of the related payments, and information about the type of documents which are supposed to be submitted, and
- Validity period: given by a valid-to-date, until which the access is requested.

The ASPSP checks if the request is valid and might adjust the validity period. If the requested consent type is not supported by the ASPSP, the ASPSP will use a dedicated message code in the error messaging.

In the next step, the PSU authorises the access towards the ASPSP by SCA. The strong customer authentication might be exempted by the ASPSP. The API Client might request information about the consent status, in order to supervise the process of the SCA and conduct further information about the consent resource, in particular if the ASPSP adjusted parameters or not. If the initiation process of the document service was successful, the API Client informs the PSU about it.

After the authorisation process is completed, the API Client may submit documents on the PSU's behalf within the validity period without further involvement of the PSU.

After the authorisation process is completed, a consent token is created and transmitted to the API Client. The API Client may store the token but it is not necessary for a documents service execution or for other technical reasons.. Then this service corresponding to the validated account and within the validity period is executed without further involvement of the PSU.

4.5 API Service: Cancellation of a Consent

The following figure only shows the very top level information flow:

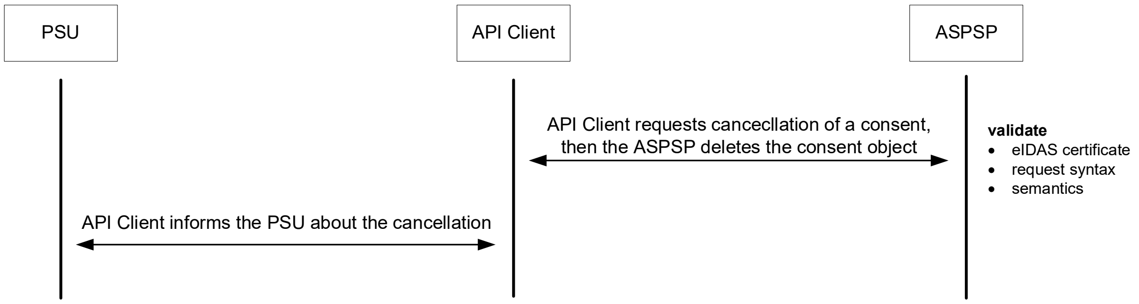
PSU
API Client
ASPSP
API Client requests cancecllation of a consent, then the ASPSP deletes the consent object
validate
- eIDAS certificate
- request syntax
- semantics
API Client informs the PSU about the cancellation

Figure 6: Cancellation of a consent

The API Client can cancel a consent object if needed. In that case, the API Client requests the deletion of a consent resource. The ASPSP checks the request and if it is valid, this terminates the given consent. For this, no SCA or other involvement of the PSU is necessary. The API Client receives a response message confirming the cancellation.

The API Client might inform the PSU about the cancellation of the consent resource.

5 Operational rules

This section summarises the operational rules for the following two cases:

- The operational rules to be observed by each TPP accessing the XS2A interface and each ASPSP providing an XS2A interface. Not all of these rules are enforced by technical means of the XS2A interface. In addition, the general operational rules as specified in [oFA-OR-FW] and [oFA-OR-Com] apply.
- The operational rules specifically to be observed by each API Client accessing the openFinance API for the Extended Account Information Service and each ASPSP providing the Extended Account Information Service at the openFinance API. Not all of these rules are enforced by technical means of the Extended Service within the openFinance API. In addition, the general operational rules as specified in [oFA-OR-FW] apply.

The order of the rules does not represent an order of importance.

5.1 Client Identification and Authorisation

The API Client needs to identify itself when first accessing the API. The ASPSP will reject an API Client if the identification is not completely correct. The process of identifying and authorising an API Client is described in detail in Section 5.3 in [oFA-OR-FW].

5.2 Consent of the PSU

A PSU is always involved in the execution of an establishing consent service. It must be made transparent towards the PSU, which of its accounts/account types are involved and which kinds of information are made accessible towards the API Client with this consent. The PSU's verification must be obtained by executing strong customer authentication.

5.3 Expiry date for consents

A consent is valid directly after the respective consent resource is successfully authorised by the PSU using SCA. The maximum time frame supported by an ASPSP is defined by the ASPSP or an API access scheme. The API Client may request an expiry date within the first request message, the ASPSP may adjust this date to shorten the validity period and must inform the API Client about this change. By requesting an expiry date in the far future (e.g. 9999-01-01), the ASPSP will limit the date to its maximum expiry date.

In general, the consent for a service may be cancelled by the PSU using an ASPSP channel directly without involving the API Client. The API Client may also terminate the consent of a service, if it is needed.

For the services CONS-AIS, CONS-XAIS and CONS-USP, if a recurring service is already consented for a PSU's account with the API Client and the same API Client requests the same service as recurring service for the same PSU, then, as soon as the new consent is authorised by the PSU, the consent for the former service is automatically expired.

5.4 Establish Consent Request/Response messages

The Establish Consent Request and Response messages are Transaction Initiation Request and Response messages, respectively. The operational rules for these categories of messages apply, cf. [oFA-OR-FW].

6 Message and data model

In the following, an abstract data model is presented for the specific usage of the consent services within the openFinance API. The basic abstract data model for the XS2A Interface and for the openFinance API is defined in [oFA-OR-Com] and [oFA-OR-FW], respectively.

The data model for these services is further detailed in the implementation guidelines [oFA-CO] and [oFA-IG-XAIS].

6.1 Protocol Level

Within both the XS2A Interface and the openFinance API, an establish consent service always starts with the Establish Consent Request and the Establish Consent Response. These belong to the categories of Transaction Initiation Request and Transaction Initiation Response messages, respectively. The message and data model of these type of messages is presented in detail in [oFA-OR-FW]. In this document, only the additional parameters will be established.

6.2 Consent data model

6.2.1 Establish consent

6.2.1.1 Establish Consent Request

In addition to the parameters of a Transaction Initiation Request, the following data attributes are supported for all kinds of Establish Consent Requests:

- Account Access (mandatory)

  This is a data structure describing the requested access to the PSU's accounts, i.e. access to payment accounts, cards, card accounts, savings accounts, loans accounts, or securities accounts. This data structure will refer different account information types like account details, balances and payment transaction information. The API Client then can address for all of these account information types the exact list of accounts.

  In addition, the ASPSP can optionally offer to support the submission of the requested account information types without addressing specific accounts. The ASPSP then will

  - either agree with the PSU within a Redirect SCA Approach or OAuth2 SCA Approach on possible restrictions of these account accesses to certain of the PSU accounts

  - or grant access to all available payment accounts of the PSU.

- Consent Type (mandatory)

The technical consent type describing the level of information to be accessed with this request. In general, the following consent types may be the supported at the API: detailed, global, aspspManaged and accountList. The consent type does affect the attribute conditions in the account access parameter, thus a detailed description is contained in the respective subsections below.

Optionally, the ASPSP might accept specific access rights, or a command where only requested access rights are inserted without mentioning other addressed account(s). This is handled between the PSU and the ASPSP directly.

- Recurring Indicator (mandatory)

  Specifies if the access to the account within this consent is for recurring usage or for a one-time access usage

- Validity (mandatory)

  The end date of the validity of the consent. For services at the XS2A interface, this is restricted by [RTS] to maximal 90 days. For a consent for a one off access, the current date is to be used.

6.2.1.1.1 Establish Consent on Account Information Request

The parameters which may occur in an Establish Consent on Account Information Request in addition to the ones listed above, are the following:

- Consent Type (mandatory)

  - 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.

- Access Frequency (mandatory)

  Maximal access to the account within this consent per day and without PSU involvement.

6.2.1.1.2 Establish Consent on Confirmation of Funds Request

The parameters which may occur in an Establish Consent on Confirmation of Funds Request in addition to the ones listed for a general Establish Consent Request, are the following:

- Consent Type (mandatory)

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

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

- Card Information (optional)

- Registration Information (optional)

  Additional information about the registration process of the PSU

6.2.1.1.3 Establish Consent on User Parameters Access Request

The parameters which may occur in an Establish Consent on User Parameter Access Request in addition to the ones listed for a general Establish Consent Request, are the following:

- Consent Type (mandatory)

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

  - aspspManaged: 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.

6.2.1.1.4 Establish Consent on Document Services Request

The parameters which may occur in an Establish Consent on Document Services Request in addition to the ones listed for a general Establish Consent Request, are the following:

- Consent Type (mandatory)
  - 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.
  - aspspManaged: 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.
- Registration Information (optional)

  Additional information about the registration process of the PSU

6.2.1.2 Establish Consent Response

Besides the established parameters of a Transaction Initiation Response, only one additional data element is supported:

- ASPSP Multiple Consents Support (conditional)

  Indicator if the ASPSP supports multiple consent service. This is a service where the ASPSP does not let a consent resource be automatically terminated if a new recurring consent token for the same PSU is requested, i.e. the status of the old consent resource may not automatically change to "replaced by API Client".

6.2.1.3 Consent Status Request and Response

This request is used, when a status of the authentication of the PSU is needed by the API Client, e.g. in the Redirect, OAuth2 or decoupled SCA Approach. This request can be sent as long as the resource is accessible.

No specific data elements are supported in request or response.

6.2.1.4 Consent Details Request and Response

This request is addressed on a created resource and requesting to retrieve the details of the consent resource. This request can be sent as long as the resource is accessible. This request might be needed for the API Client if the PSU has withdrawn the consent (partially or implicitly) via the PSU – ASPSP interface. The request contains no specific data elements.

The corresponding response contains in its payload the current consent object, the detailed data structure is defined in [oFA-CO].

7 Annexe

7.1 Glossary

AIS

Account Information Service according to article 4 (16) of [PSD2] and as regulated by article 67 of [PSD2].

AISP

Payment service provider offering an AIS to its customer. See article 4 (19) of [PSD2].

API Client

Service provider accessing the openFinanceAPI of an ASPSP. A TPP is a possible example of an API client.

ASPSP

Account Servicing Payment Service Provider providing and maintain a payment account for a payer. See article 4 (17) of [PSD2].

PIISP

Payment Instrument Issuer Service Provider according to article 4 (14) and 45) of [PSD2]. A PIISP can use the service "Confirmation on the availability of funds" as regulated by article 65 of [PSD2].

PIS

Payment Initiation Service according to article 4 (15) of [PSD2] and as regulated by article 66 of [PSD2].

PISP

Payment service provider offering a PIS to its customer. See article 4 (18) of [PSD2].

PSP

Payment service provider according to article 4 (11) of [PSD2].

PSU

Payment Service User according to article 4 (10) of [PSD2].

openFinance API

Premium access to account interface – interface provided by an ASPSP to API clients for accessing extended services offered by the ASPSP.

QTSP

Qualified Trust Service Provider, e. g. a trust centre issuing qualified certificates

SCA

Strong Customer Authentication – authentication procedure based on two factors compliant with the requirements of [PSD2] and [RTS].

TPP

Third Party Provider – generic term for AISP/PIISP/PISP.

TSP/QTSP

Trust Service Provider according to [eIDAS]. Within the context of the XS2A interface specification only qualified TSPs (QTSPs) according to section 3 of [eIDAS] issuing qualified certificates for electronic seals and/or qualified certificates for website authentication which are compliant with the requirements of [RTS] are relevant.

XS2A interface

Access to account interface – interface provided by an ASPSP to TPP for accessing accounts for [PSD2] regulated services.

7.2 References

[oFA-IG-ADM] openFinance API Framework, Implementation Guidelines, Administrative Services, Version 1.1, 09 September 2024

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

[oFA-IG-XAIS] openFinance API Framework, Implementation Guidelines, Extended Account Information Services, Version 2.2, 17. April 2025

[oFA-IG-Com] openFinance API Framework. Implementation Guidelines, Compliance Service, Version 2.4, 31 October 2025

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

[oFA-OR-FW] openFinance API, Operational Rules, Basic Operational Rules of the Framework, version 2.0, 31 October 2025

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

[oFA DaD] openFinance API Framework, Data dictionary for V2 services, Version 2.2, 31 July 2024

[RTS] Commission Delegated Regulation (EU) 2018/§() 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, L69/23, Official Journal of the European Union, 13.03.2018

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

[PSD2] Directive (EU) 2015/2366 of the European Parliament and of the Council on Payment Services in the Internal Market, published 25 November 2016

7.3 List of figures

7.4 List of tables