THE Berlin GROUP

A EUROPEAN STANDARDS INITIATIVE

openFinance API Framework

Operational Rules for Extended Services

Push Account Information Services

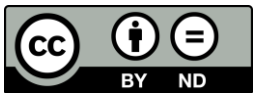
Version 2.3

31 March 2026

Berlin Group License Notice

Berlin Group Specifications are 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.
- Any OpenAPI files offered by Berlin Group are bound by “Creative Commons Attribution 4.0 International Public License” and are allowed to be changed. However, please note that the normative reference still is the Implementation Guidelines document.
- 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.

Contents

1 Introduction 1
1.1 From Core XS2A Interface to openFinance API 1
1.2 Push Account Information Services as Extended Services 2
1.3 Documents 4
2 Extended Services of the openFinance API 6
2.1 Services of the openFinance API 6
2.2 Extended services covered in this document 6
3 API Services supported for the Push AIS 8
3.1 Supported Service Types 8
3.2 API Service: Establish a Push Account Information Service Subscription 11
3.2.1 Subscription model for natural persons 12
3.2.2 Subscription model for corporates 13
3.2.3 Technical differentiation 15
3.3 API Service: Update an existing Push Account Information Service Subscription 15
3.4 API Service: Cancel existing subscription via the openFinance API 17
3.5 API Service: Notification of a subscription/ subscription entry cancellation 18
3.6 API Service: Push account information of a dedicated category and of a dedicated account 18
3.7 API service: Push account information of a dedicated category and of a dedicated account to a fall-back URI 19
3.8 API Service: Push account triggers of a dedicated account 20
4 Key concepts of the openFinance API for the Push Account Entry Service 22
4.1 Confirmation of the consent of the PSU 22
5 Operational rules 24
5.1 Client Identification 24
5.2 Client Authorisation 24
5.3 Coding of business data 24

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

5.4 Uniqueness of a subscription for push account entries for personal subscription 24
5.5 Uniqueness of a subscription for push account entries in specific corporate banking APIs 25
5.6 Consent of the PSU 25
5.7 Decision about strong customer authentication 25
5.8 Signing baskets 26
5.9 Avoiding 90 days rule by GDPR consent 26
6 Message and data model 27
6.1 Protocol Level 27
6.2 Authorisation Data Model 27
6.3 Subscription data model 27
6.3.1 Establish subscription transaction 27
6.3.2 Push account information transaction 33
6.3.3 Push account information trigger transaction 34
7 Annex 36
7.1 Glossary 36
7.2 References 37
7.3 List of figures 38
7.4 List of tables 38

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

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.

The XS2A Framework is now planned to be extended to premium 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 API 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 API Client's bank for KYC processes.

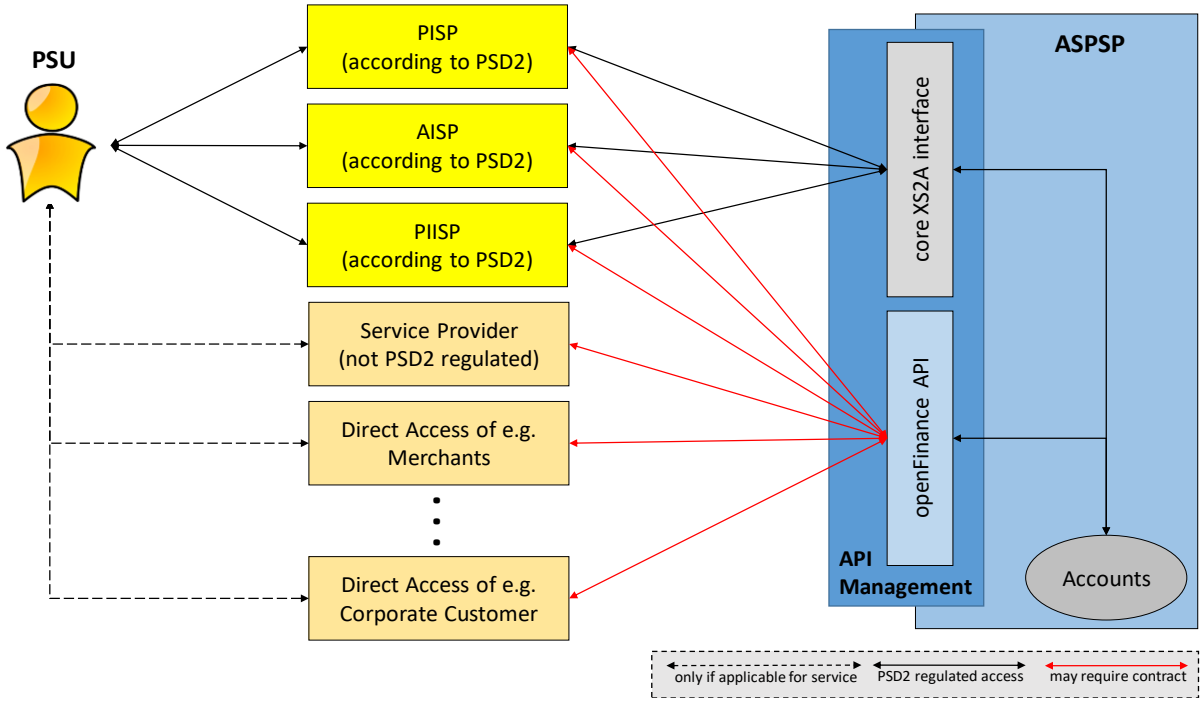
Note: The notions of API Client and ASPSP are used because of the technical standardisation perspective of the openFinance API. These terms are analogous to "asset broker" and "asset holder" resp. in the work of the ERPB on a SEPA API access scheme.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Note: In implementations, several ASPSPs might offer their services in 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:



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 the services offered at the openFinance API are out of scope of this document. These requirements will be described in a dedicated operational rules document [oFA-OR-Adm].

1.2 Push Account Information Services as Extended Services

The core XS2A Interface as introduced above is for Account Information Services (AIS) in general based on a "pull philosophy": The TPP can **pull** AIS related information from the ASPSP once the PSU has given his/her consent and the PSU is authenticated by SCA methods provided by the ASPSP. Triggers for pulling can be the PSU as such (PSU present at the TPP's application asking for a refresh) or a 4 times offline (PSU not present) access of the TPP to the PSU's accounts per day.

A first "push functionality" was introduced in the core XS2A interface for informing the TPP about technical status changes of major API resources which have been submitted by the TPP itself, e.g. an information that a payment has been authorised by the PSU in a decoupled SCA process. The core XS2A interface does not support pushing new financial account related information like new account entries for e.g. incoming instant payments.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

It is now planned to introduce a service to enable ASPSPs to inform PSUs via API Client systems about account information into the openFinance API Framework. This service might require a contract between the API Client and the ASPSP. It is called in the following **Push Account Information Service**. The openFinance API Framework will support several subservices from the beginning: Varying categories of account information like

- account entries,
- account statements,
- account balances, and
- incoming RTPs

will lead to different subservices.

As a first subservice, the pushing of account entries is standardised. This subservice is called **Push Account Entry Service**. The Push Account Entry service will be available in two variants:

A next subservice is a service to push account statements. The statements are containing more information as the transaction reports as provided today within the PSD2 NextGenPSD2 API. This subservice is called **Push Account Statement Service**. This service will allow to transport account statements either in MT94x or in camt.05x format. Starting with version 2.0, also JSON based statements will be supported by this standard.

A third subservice is pushing balance related information, e.g. inform the API Client in case a certain balance threshold is met. This subservice is called **Push Balance Information Service**.

A fourth subservice is pushing incoming RTP Requests, e.g. forward all incoming RTP Requests which are due in a certain short time frame or warn about RTP Request reaching the expiry date. This subservice is called **Push RTP Information Service**.

All these subservices come in two variants:

- Push an account information directly to an **entry URI** provided by the API Client, or
- Push a trigger (a hyperlink or static text) for the related account information to a **trigger URI** provided by the API Client. In a next step, the API Client would need to pull the account information data, using an underlying PSU consent.

Please note that all these services come with a subscription by the PSU, which is initiated by the API Client. Each subservice might be triggered by events, pre-defined timeslots etc. The **trigger criteria**, or "**criteria**" in short in this text, are agreed on during the subscription process.

All these subservices are technically separated for the different categories of account information: a subscription is always authorised for **one** account information category only.

Remark: Even if the subscription model followed within this service is a GDPR like consent between ASPSP and PSU on forwarding account information data to a third party, PSD2

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

requirements might still apply on e.g. SCA related periods. This is under review and up to API Access Scheme definitions or bilateral commercial contracts between API Client and ASPSP. Anyhow, variants where just e.g. hyperlinks pointing to new account information or static trigger text are pushed to the API Client might be easier implemented from a legal/contractual point of view.

Note: The Push Account Information Service for the openFinance API is built in such a way that it can also be used in a direct PSU – ASPSP interface. Some remarks specific to such an implementation are integrated in this document.

1.3 Documents

The document at hand gives an overview on the functional model of the Push Account Information Service, where details are only provided for the subcategory Push Account Entry Service. The technical specification of the Push Account Entry Service in form of an API specification is not part of this document. Please refer to the corresponding Implementation Guidelines [oFA-IG-Push] for this technical specification.

1.4 Document History

Version	Change/Note	Approved
0.9	First Version for market consultation	03 February 2021, openFinance TF
0.91	Clarifications added after first feedback from the advisory board	17 February 2021, openFinance TF
1.0	More sub services added after consideration on Implementation Guidelines level.	17 December 2021, openFinance TF
1.1	Adaption of the abstract data model by a review and adding a model for the Add Subscription Entries Request in Section 6.3.1	15 February 2022, openFinance TF
2.0	Adaption to Version 2.	08 February 2024, openFinance TF
2.1	Adaption to openFinance API Framework Version 2.1	31 July 2024 openFinance TF
2.2	Adaption to the new version of the Implementation Guidelines of the Push Account Information Services and to the openFinance API Framework Version 2.3	11 December 2025

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Version	Change/Note	Approved
2.3	Adaption to the new version of the Implementation Guidelines of the Push Account Informaton Services and to the openFinance API Framework Version 2.4, by adding a note in Section 3.3 and updating references. Correction of Service Types in Section 3.1.	19 March 2026

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

2 Extended Services of the openFinance API

2.1 Services of the openFinance API

The openFinance API supports different services. It is distinguished between core services and extended services. According to PSD2 requirements an ASPSP must support all core services at its XS2A interface towards TPPs. These core services could also be offered in the openFinance API to other access clients, e.g. own corporate customers.

The ASPSP is further free to decide which extended service to support in its implementation of the openFinance API in accordance with its own market needs.

2.2 Extended services covered in this document

An ASPSP supporting extended services at its openFinance API may limit access to these extended services to a special group of TPPs or other API Clients. If requested by the ASPSP a contractual relationship regulating the usage of the extended service shall be established between the ASPSP and the TPP, resp. more generally the ASPSP and the API Client.

It might also be that the PSU needs to agree with the ASPSP on such a service explicitly, e.g. when ASPSP costs for the PSU are involved for the support of this service. Such a potential agreement might be concluded during the redirect or decoupled SCA and is out of scope of this document.

The following **subservices** of Pushing Account Information Services are addressed in this document within the openFinance API Framework:

Services addressed	**Usage**
Push Account Entry Service	Service which supports to push single account entries **or** a trigger for account entries to a URI of the AISP. This service needs to be subscribed to by the PSU via the API Client as a communication channel. Within this subscription, • the categories of account entries, • the elements of an account entry to be pushed and • the dedicated URI of the API Client where the account entries shall be pushed to are communicated to the ASPSP. The subscription is secured towards the ASPSP by a SCA of the PSU.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Push Account Statement Service	Service which supports to push account statements or a trigger for statements to a URI of the AISP. This service needs to be subscribed to by the PSU via the API Client as a communication channel. Within this subscription, • the preferred format of the account statements, • the events triggering the pushing of account statements, • the dedicated URI of the API Client where the account statements shall be pushed to are communicated to the ASPSP. The subscription is secured towards the ASPSP by a SCA of the PSU.
Push Balance Statement Service	Service which supports to push balances or a trigger to a URI of the AISP. This service needs to be subscribed to by the PSU via the API Client as a communication channel. Within this subscription, • the addressed balance types, • the events triggering the pushing of balances, • the dedicated URI of the API Client where the balances shall be pushed to are communicated to the ASPSP. The subscription is secured towards the ASPSP by a SCA of the PSU.
Push RTP Service	Service which supports to push incoming Requests to Pay (RTP) or a trigger to a URI of the AISP. This service needs to be subscribed to by the PSU via the API Client as a communication channel. Within this subscription, • the preferred format of the RTPs, • the events triggering the pushing of RTPs, • the dedicated URI of the API Client where the RTPs shall be pushed to are communicated to the ASPSP. The subscription is secured towards the ASPSP by a SCA of the PSU.

Table 1: Extended services scheduled for further detailed work

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

3 API Services supported for the Push AIS

3.1 Supported Service Types

This document introduces the following Service Types

- Extended Account Information Pushing Service with the four subservices
  - for Account Entries (PAIS-AE)
  - for Account Balances (PAIS-BI)
  - for Account Statements (PAIS-AS)
  - for incoming RTP (PAIS-RTP).

These service types are subsuming the following API Services, where XX needs an instantiation of the four related subservices AE, AS, BI, RTP:

API Services	Service Type	Functionalities	PSU directly involved
Establish a push account information subscription	PAIS-XX	Authorisation of the submitted subscription entries in a dedicated sub service by the PSU	yes
Change an existing push account information subscription	PAIS-XX	Deletion of certain subscription entries in a dedicated sub service	no
		Authorisation of adding a subscription entry in a dedicated subservice	yes
Cancellation of a service subscription via the openAPI interface	PAIS-XX	Deletion of the full related subscription	no
Cancellation of a service subscription/ subscription entry via the ASPSP	PAIS-XX	Deletion of the subscription/ of some subscription entries in a dedicated subservice via the ASPSP (outside the API)	yes
		Information of the status change of the related subscription/subscription entry in the dedicated sub service	no
Pushing an account entry	PAIS-AE	Push the actual account entry to the URI provided by the API Client	no
Pushing an account entry to a fallback URI	PAIS-AE	Push the actual account entry to the fallback URI, if provided by the API Client	no

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

API Services	Service Type	Functionalities	PSU directly involved
Pushing an account entry trigger	PAIS-AE	Push a trigger to a API Client or direct access client that relevant data can be pulled from the related interface, e.g. the core XS2A interface	no
Pushing an account statement	PAIS-AS	Push the actual account statement to the URI provided by the API Client	no
Pushing an account statement to a fallback URI	PAIS-AS	Push the actual account statement to the fallback URI, if provided by the API Client	no
Pushing an account statement trigger	PAIS-AS	Push a trigger to a API Client or direct access client that relevant data can be pulled from the related interface, here the premium extension of the core interface for account statements.	no
Pushing an account balance	PAIS-BI	Push the actual account balance to the URI provided by the API Client	no
Pushing an account balance to a fallback URI	PAIS-BI	Push the actual account balance to the fallback URI, if provided by the API Client	no
Pushing an account balance trigger	PAIS-BI	Push a trigger to a API Client or direct access client that relevant data can be pulled from the related interface, e.g. the core XS2A interface	no
Pushing an incoming RTP	PAIS-RTP	Push the actual incoming RTP to the URI provided by the API Client	no
Pushing an incoming RTP to a fallback URI	XPAIS-RTP	Push the actual incoming RTP to the fallback URI, if provided by the API Client	no

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

API Services	Service Type	Functionalities	PSU directly involved
Pushing an incoming RTP trigger	XPAIS-RTP	Push a trigger to a API Client or direct access client that relevant data can be pulled from the related interface, here the premium extension of the core interface for account statements.	no

Table 2: API services for the Push Account Information Service

In addition, the openFinance API will support technical API services/micro services within the RESTful API approach which are not necessarily used within the above mentioned API services, e.g. to read details on subscription objects or other created resources. Further details on the technical API services are defined in [oFA-IG-Push].

Not all of the above API services have to be supported at the openFinance API of an ASPSP for the Push Account Information Service.

The execution of any transaction at the openFinance API is subject to the consent of the PSU. Some API services require direct involvement of the PSU while others do not. This is specified in column "PSU directly involved".

If a subscription transaction based on an API service requires direct involvement of a PSU, strong customer authentication of the PSU may be necessary. Please refer to [oFA-OR-FW] for details about strong customer authentication. One of the purposes of a strong customer authentication is to prove that the subscription transaction is executed with the consent of the PSU.

If a transaction is based on an API service that does not require direct involvement of the PSU, strong customer authentication is not possible. In this case the PSU has to give consent by other means prior to the transaction. A longer time period may elapse between the PSU giving the consent and the actual execution of the transaction by the API Client. The steps necessary for giving and proving the consent of the PSU depend on the API service and will be explained in the following sections and in section 4.1.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

3.2 API Service: Establish a Push Account Information Service Subscription

The support of this API service for the Push Account Information Service at the openFinance API is mandatory. It might be restricted to one or more sub-services for different categories of account information.

The API service of establishing a Push Account Information Service subscription is separated in the openFinance API of any potential consent process in the XS2A core interface which might be mandated to the related TPP in addition.

The subscription to be established contains the following information:

- Generic subscription information:
  - Preference on application layer encryption for the push service
  - Encryption certificate if applicable
  - Duration of the subscription
- One or more so called **subscription entries** for push services within one account information category e.g. for different accounts or different triggering criteria.

Each subscription entry contains the following information

- Related account reference
- A subscription entry name for displaying on the ASPSP site to the PSU
- Account Information related criteria, which will trigger a push account information service in the related category of account information, see below.
- Additional parameters for the push account information services, like preferred formats etc., see below.
- A preference of the API Client to receive the information only as a hyperlink or with a static text as defined by the API Client. This preference might be ignored by the ASPSP if not supported.
- URI information for the API Client entry interface, where related account information is pushed to.

The account information category specific data are defined in more detail in Section 6.3.1.5 (trigger criteria) and Section 6.3.1.6 (additional parameters)..Details of these definitions are given in [oFA-IG-Push].

NOTE: The "subscription entry" level within the subscription data model allows the API client also to handle multi agent scenarios, where an API Client is offering connectivity services to its agents in a non-transparent way. Several subscription entries may e.g. be used for the same account within one subscription. This then could lead to pushing account information to different URIs by different subscription entries within the same subscription. The notion of a unique subscription e.g. per PSU in a retail banking context or e.g. per corporate in a corporate banking context is needed to guarantee an optimised data keeping, cp. also Section 3.2.1 and Section 3.2.2.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

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

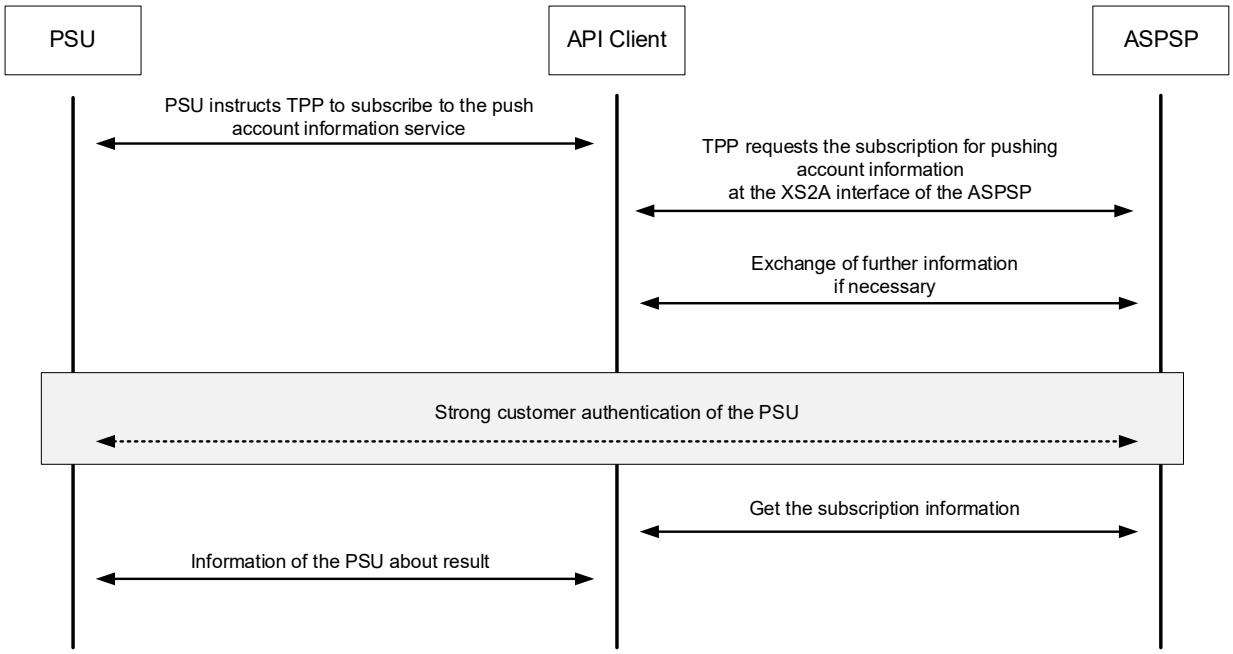


Figure 1: API Service Establish push account entry subscription

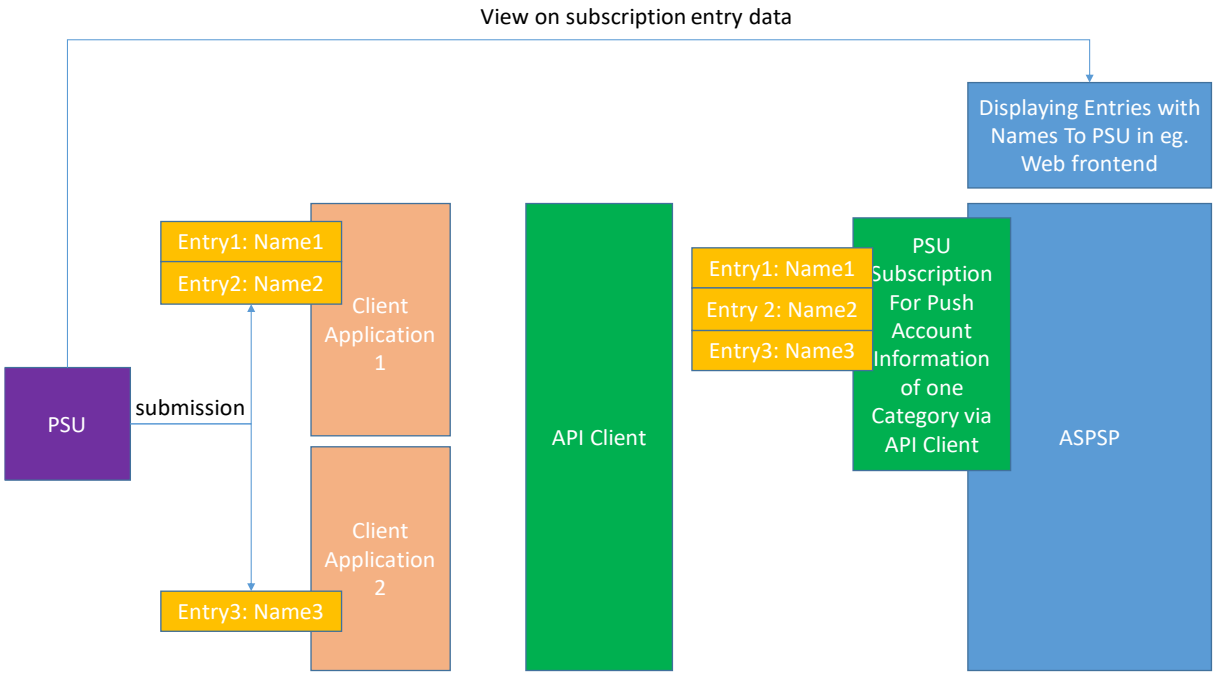
3.2.1 Subscription model for natural persons

If a subscription for a push account information service for the addressed account information category does already exist for the API Client/PSU/ASPSP triple, a further full subscription for pushing an account information cannot be submitted in a retail payment context within the same account information category, i.e. will be rejected with a dedicated error code. This rejection can only be performed when the PSU ID is known by the ASPSP, i.e. it is recommended to submit the PSU ID with the first call. After the rejection, the API client needs to use the related update procedure to extend the **existing** subscription, cp. Section 3.3.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

The following picture gives an overview on the roles of API clients, client application, PSUs, entries etc. in a retail banking context:



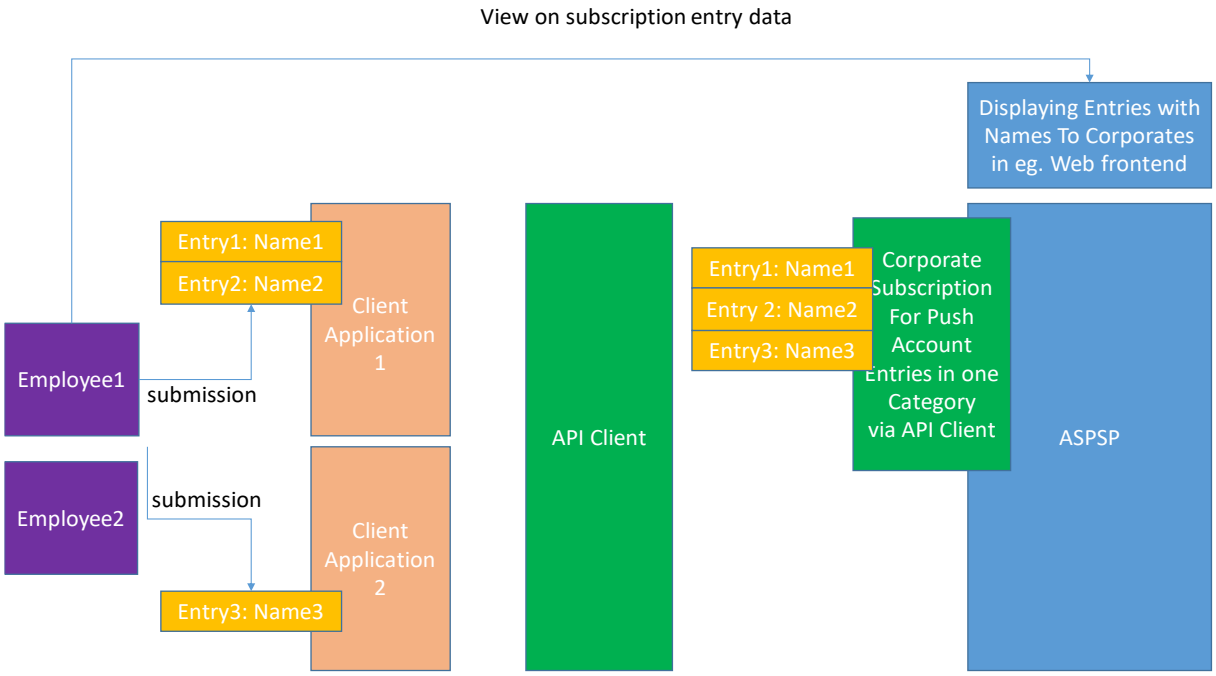
3.2.2 Subscription model for corporates

If a subscription for a push account information service for the addressed account information category does already exist for the API Client/Corporate/ASPSP triple, a further full subscription for pushing an account information cannot be submitted in a corporate payment context within the same account information category, i.e. will be rejected with a dedicated error code. This rejection can only be performed when the addressed corporate is known by the ASPSP, i.e. it is recommended to submit the PSU ID and Corporate ID (where applicable) with the first call. The API Client will only be able to submit one subscription per corporate and account information category and needs to update it later in case of any changes. In that case, the subscription entry will contain the PSU identification of the employee, to be transparent in the interface e.g. when entries would be cancelled after a corporate asked the ASPSP to do so, or where a API Client is asked by a corporate customer to cancel all entries authorised by a dedicated employee, identified by a PSU ID.

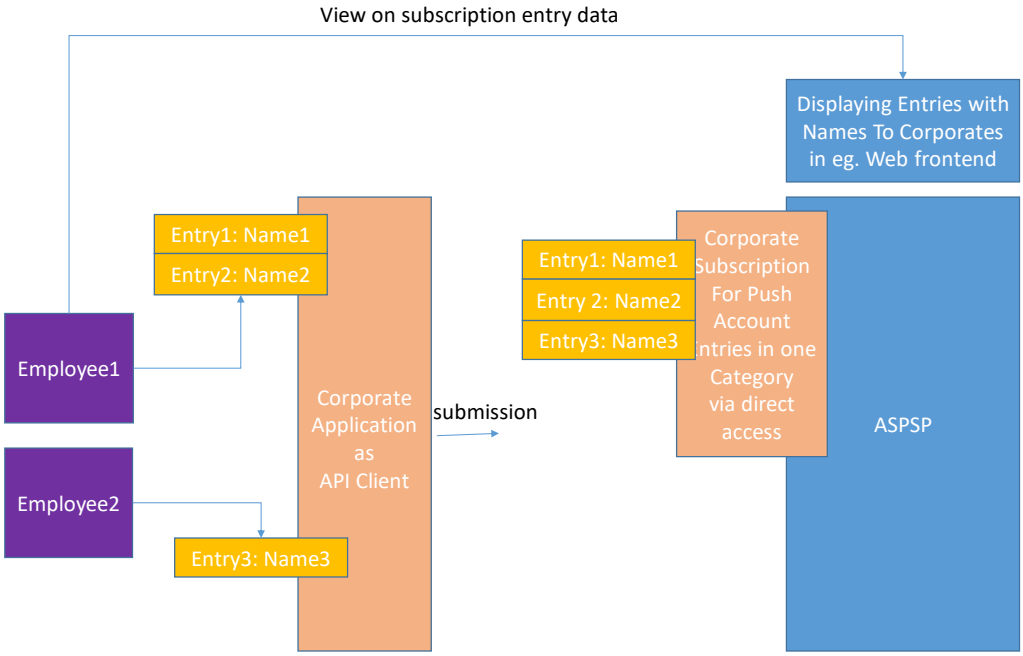
Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

The following picture gives an overview on the roles of API Clients, client application, PSUs, entries etc. in a corporate banking context (with separate API Client e.g. TPP):



In addition to the above picture, the context for direct access to the openFinance API by a corporate is shown.



(ref. License Notice for full license conditions)

3.2.3 Technical differentiation

A corporate banking situation is always given in the case where the API Client needs e.g. to add the Corporate ID when using the openFinance API. Anyhow, in the response to the subscription process to inform the API Client the ASPSP will always deliver the information on whether the response relates to a corporate subscription or to a retail banking subscription. This enables the API Client to distinguish the subscription models to be applied.

3.3 API Service: Update an existing Push Account Information Service Subscription

The support of this API service for the Push Account Information Service at the openFinance API is mandatory.

This API service only applies where a subscription for a push account information service already exists for the addressed category of account information.

The update can consist of

- the cancellation of existing subscription entries, and/or
- addition of new subscription entries.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

The following figure shows only the very top level information flow for the cancellation of existing subscription entries:

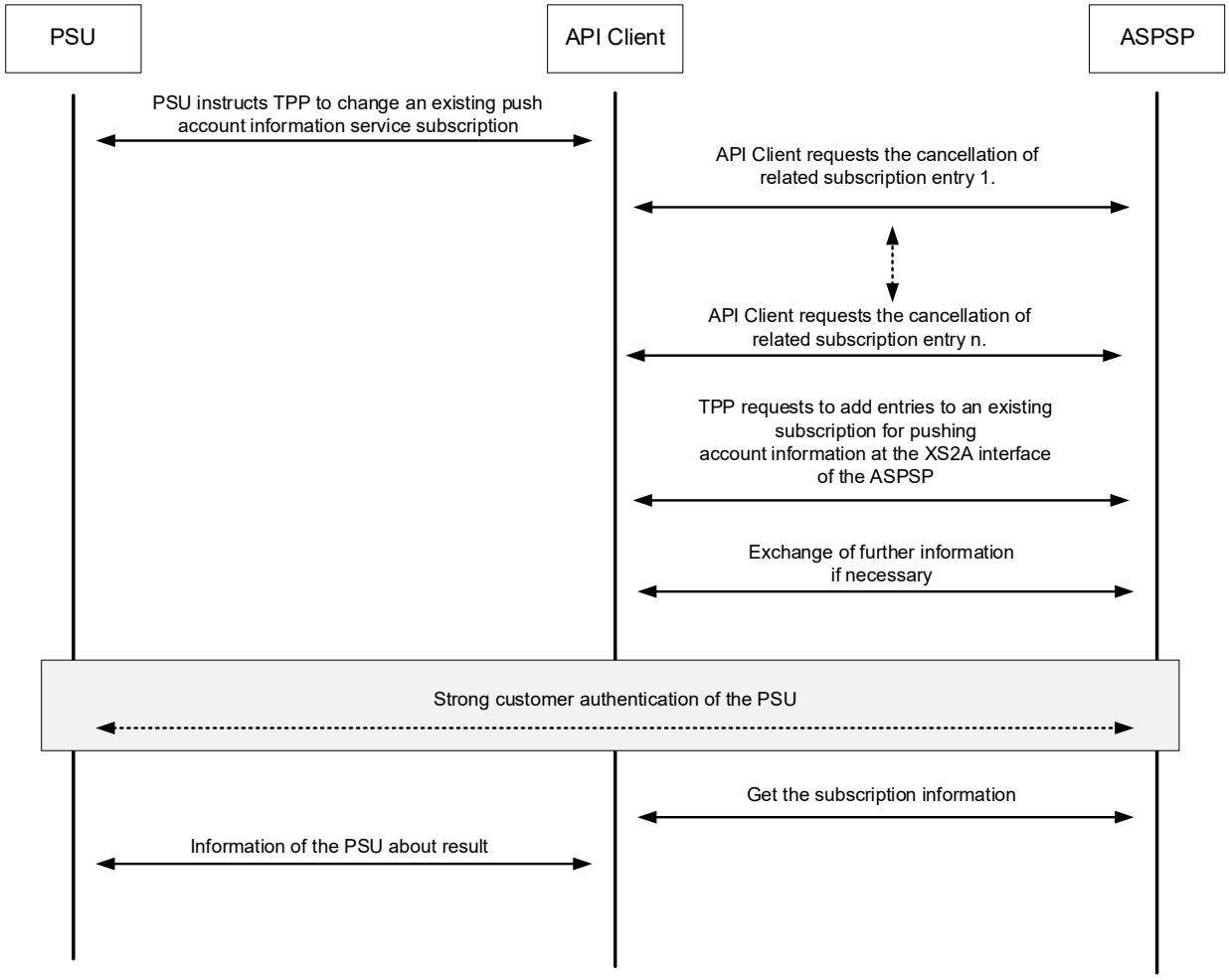


Figure 2: API Service Update push account information subscription

In case where only cancellations apply, the second part of the service would not apply. Especially, there is no SCA needed in this case.

In case where only additions of entries apply, the first service would not apply.

NOTE: Please note that the new openFinance API Framework comes with a change methodology for existing resources, which is applied in [oFA-IG-Push] to the function of adding new subscription entries. The cancellation of subscription is managed as before, since this function does not require SCA. Since the change methodology mentioned above comes with building new version of resource representation, the API Client needs to be careful in applying the functions of cancelling subscriptions to the correct version of the resource representation.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

3.4 API Service: Cancel existing subscription via the openFinance API

The support of this API service for the Push Account Information Service at the openFinance API is mandatory.

The following figure shows only the very top level information flow of a cancellation of a subscription:

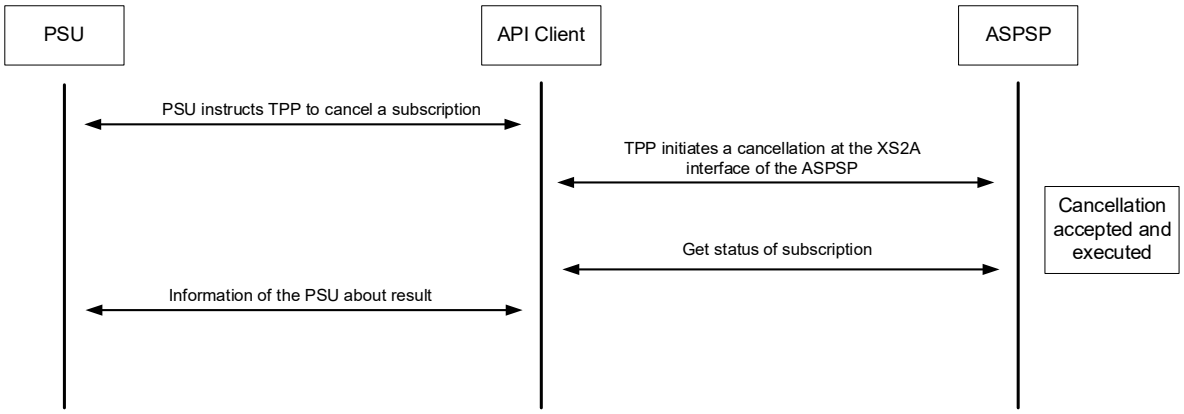


Figure 3: API Service Cancellation of a subscription

Note: The ASPSP might offer the PSU the possibility to cancel the subscription or a subscription entry directly in the PSU – ASPSP interface. This is not a dedicated API service for the openFinance API, but would be communicated to the API Client only by a change of the status of the subscription resource in the openFinance API. This status change should be pushed via a Lean Push Service by the ASPSP as defined in Section 3.5 if requested by the API Client.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

3.5 API Service: Notification of a subscription/ subscription entry cancellation

The support of this API service for the Push Account Information Service at the openFinance API is mandatory if the ASPSP offers the PSU the service to cancel a subscription/subscription entry in the PSU – ASPSP interface.

The following figure shows only the very top level information flow of a cancellation of a subscription:

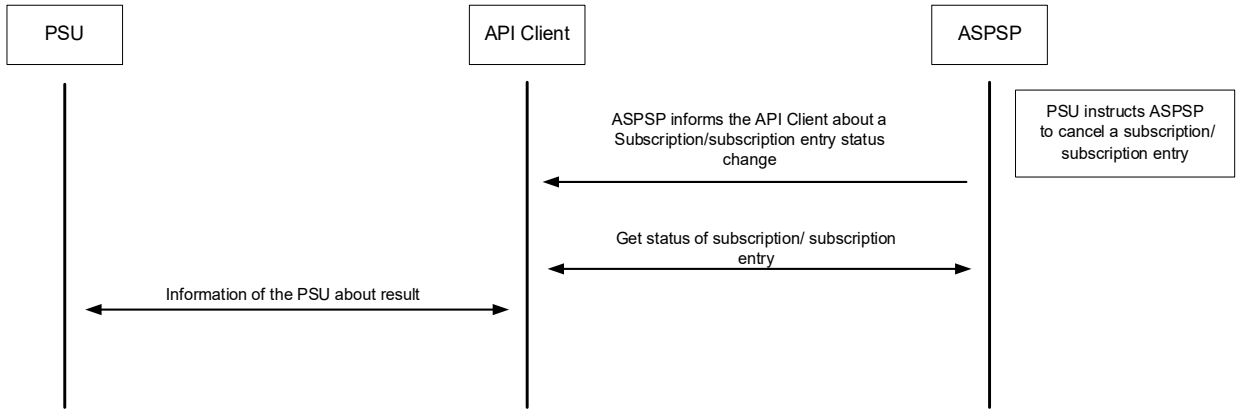


Figure 4: API Service Notification of subscription/subscription entry cancellation

This service is a generic resource status notification function as defined in [oFA-PFSM]. This service requires a registration of this service for the related subscription when submitting the subscription.

3.6 API Service: Push account information of a dedicated category and of a dedicated account

The support of this API service at the openFinance API is mandatory, if the Push Account Information Service for the related category of account information is supported as such.

Push account information instances according to this API service can be used to push all sorts of information to a PSU. The related accounts and the related account information of the addressed category are defined by the underlying subscription entry and trigger criteria resp., cp. Section 3.2.

The ASPSP is pushing all addressed account information to the API Client. If certain additional parameters are supported by the ASPSP like e.g. the attribute "attribute selection preference", then e.g. only attributes of the entry which are defined in the attribute selection preference are pushed to the API Client.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

The push transaction at the openFinance API is initiated by the ASPSP either

- as soon as a pre-defined event applies or
- a certain pre-defined timeslot applies
- or in the case of the account entry category as soon as a new account entry available as booked or pending element at the related account.

It does not have to be initiated by the PSU at the PSU – API Client interface previously. However, the PSU must still have granted its consent during the subscription phase.

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

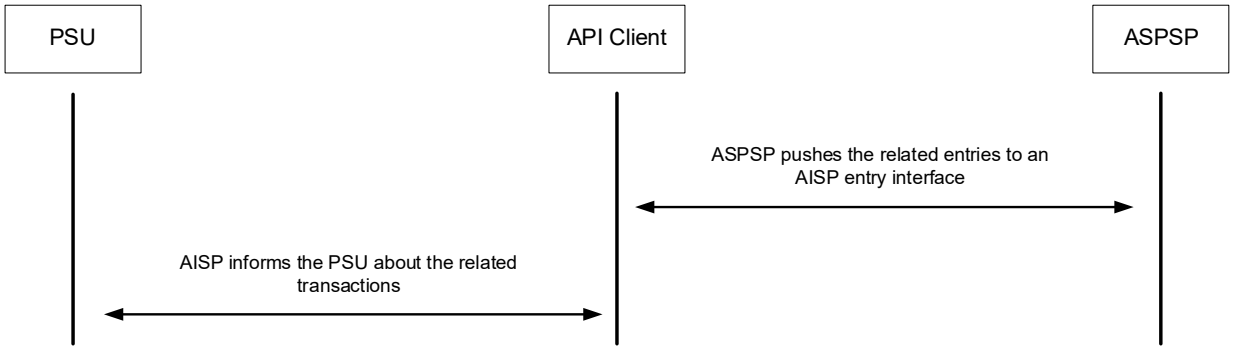


Figure 5: API Service Push account entries of a dedicated account

3.7 API service: Push account information of a dedicated category and of a dedicated account to a fall-back URI

The support of this API service at the openFinance API is optional.

This API service is an extension of the API service described in Section 3.6. It can only be supported by the ASPSP if the API Client has provided a fallback URI within the subscription process.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

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

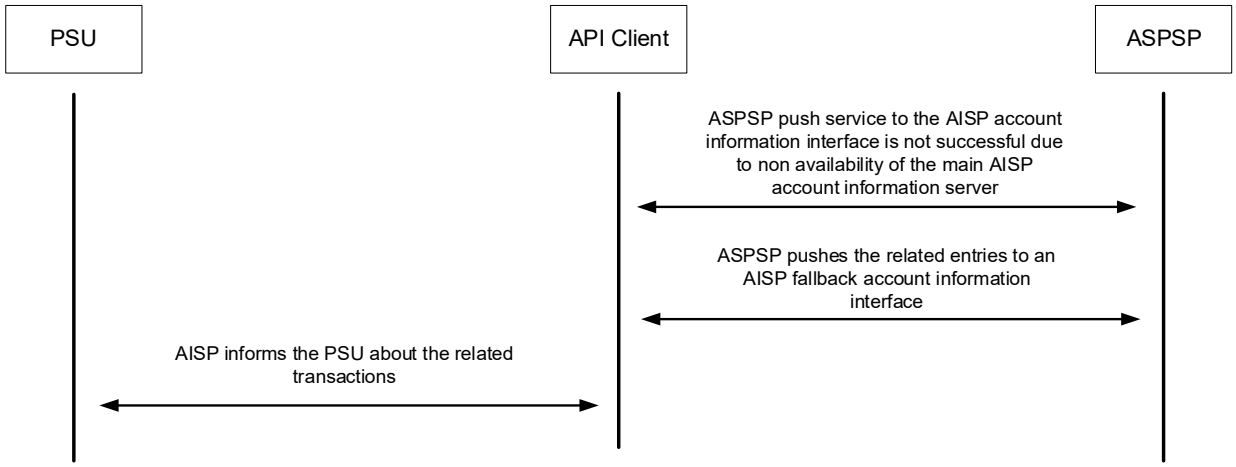


Figure 6: API Service Push account information of a dedicated account to a fallback URI

The ASPSP might use the AISP fallback entry interface for a certain time period, before trying to push account information instances again to the main AISP entry interface.

3.8 API Service: Push account triggers of a dedicated account

The support of this API service at the openFinance API is optional. This is a service applicable especially in the TPP access scenario.

Push account information instances according to this API service can be used to push all sorts of account information triggers to an AISP, such that the AISP can retrieve the information via the core PSD2 or premium extension services in a second step. The related accounts and the related account information of the addressed category are defined by the underlying subscription entry and trigger criteria resp., cp. Section 3.2.

The ASPSP is pushing triggers for all addressed account information instances to the AISP.

The push trigger transaction at the openFinance API is initiated by the ASPSP either

- as soon as a pre-defined event applies or
- a certain pre-defined timeslot applies
- or in the case of the account entry category as soon as a new account entry available as booked or pending element at the related account.

It does not have to be initiated by the PSU at the PSU – AISP interface previously. However, the PSU must still have granted its consent of the AISP being triggered during the subscription phase.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

In a second step the AISP needs to retrieve the related account information via the services defined in the core or extended openFinance API of the ASPSP. For retrieving this information from the ASPSP PSD2 or premium extended interface, the AISP needs a valid PSD2 compliant PSU consent.

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

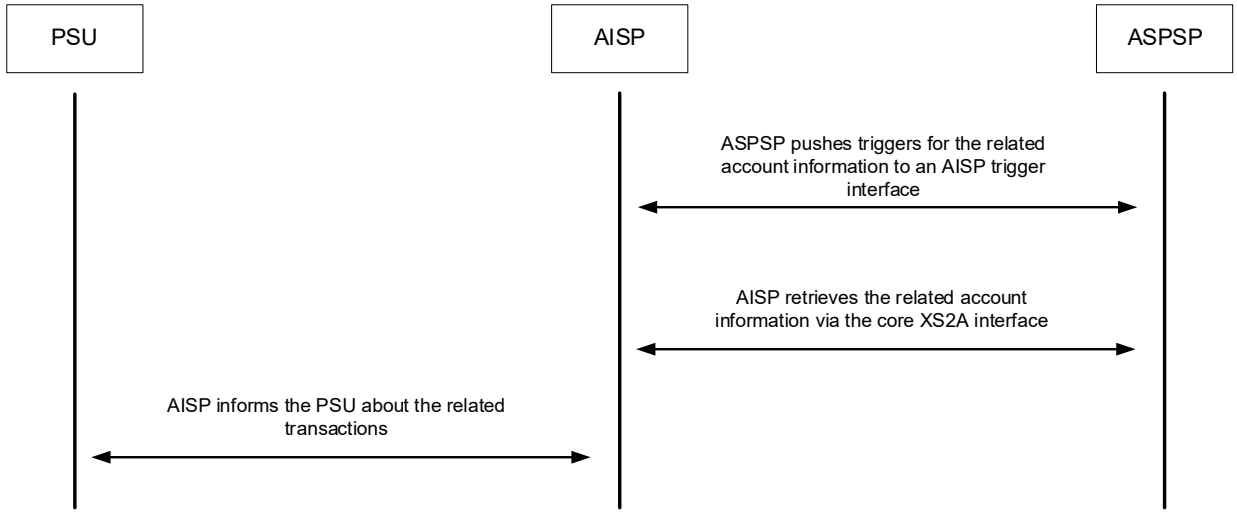


Figure 7: API Service Push account triggers of a dedicated account

(ref. License Notice for full license conditions)

4 Key concepts of the openFinance API for the Push Account Entry Service

This section contains an overview of key concepts of the openFinance API specific to the Push Account Information Service. For basic key concepts please refer to [oFA-OR-FW]. For the detailed specification of this extended service please refer to the document [oFA-IG-Push].

4.1 Confirmation of the consent of the PSU

Each transaction at the openFinance API is subject to the consent of the PSU. How consent of the PSU is confirmed during a subscription or push transaction specific to this extended service depends on the API Service as shown in the following table:

API Service	How the PSU grants consent	How consent of the PSU is verified
Establish subscription	Identifies itself as part of the subscription transaction, if necessary by strong customer authentication.	Verification of the identity of the PSU, if necessary by strong customer authentication.
Update Subscription	This process is defined by a set of potential cancellations of subscription entries and an addition of a potential set of new subscription entries respectively. The requirements for the PSU consent on cancelling/adding subscription entries equal the requirements for the respective processes for subcriptions.	See above and below.
Cancel Subscription	PSU grants consent only to API Client, if the cancellation is triggered through the API Client.	No explicit verification of the PSU consent is applied.
	PSU grants consent to the ASPSP if the cancellation is triggered at the PSU-ASPSP interface.	Verification of the identity of the PSU in the PSU-ASPSP interface, if needed by strong customer authentication.
Push Account Information	The consent on pushing the account information was consented within the subscription process.	No further verification needed, since pushing is triggered by the ASPSP.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

API Service	How the PSU grants consent	How consent of the PSU is verified
Push Account Information Trigger	The consent on pushing the account information was consented within the subscription process.	No further verification needed, since pushing is triggered by the ASPSP.
	The consent on retrieving the actual account information was consented in an underlying PSD2 or premium API consent between PSU and AISP.	Access token given to the API Client was verified as result of a previously executed Establish account information consent transaction of the underlying PSU AIS consent, cp. [oFA-OR-FW].

Table 3: Consent of the PSU within API Service

A subscription or push transaction at the openFinance API may only be executed if the consent of the PSU can be confirmed where needed. Otherwise the ASPSP will reject the transaction.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

5 Operational rules

This section summarises the operational rules specifically to be observed by each API Client accessing the openFinance API for the Push Account Information Service and each ASPSP providing the Push Account Information Service at an openFinance API. In addition, the general operational rules as specified in [oFA-OR-FW] apply.

Not all of these rules are enforced by technical means of the Extended Service within the openFinance API.

5.1 Client Identification

Accessing clients are identified based on eIDAS certificates in analogy to the Core XS2A Interface, but without mandating PSD2 specific attributes in the certificate.

5.2 Client Authorisation

ASPSPs might refuse access to the openFinance API even if the related eIDAS certificate is valid, if not all related contract conditions are fulfilled. This might be controlled e.g. via a scheme directory.

5.3 Coding of business data

For the pushing of account information the ASPSP shall support one or more of the following coding of the account data to be delivered:

- camt.05x,
- JSON,
- MT94x.

The JSON structure to be used for transferring the account information are defined by the data dictionary [oFA-DaD].

In any case it is up to the ASPSP to decide what kind of coding will be supported for pushing account information at its openFinance API. The ASPSP will inform the API Client about its decision as part of the documentation of this extended service in its openFinance API.

5.4 Uniqueness of a subscription for push account entries for personal subscription

At most one active subscription instance is supported for every guadruple (API client, PSU, ASPSP, account information category) in the openFinance API, if the ASPSP system is a retail banking system.

If an additional subscription for a push account information service for a dedicated account information category is submitted to the ASPSP by the API Client following the request of a PSU, where a related valid subscription for a push account information service for the related

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

account information category exists already for this PSU, then this submission will be rejected with a dedicated error code.

5.5 Uniqueness of a subscription for push account entries in specific corporate banking APIs

At most one active subscription instance is supported for every quadruple (API client, Corporate, ASPSP, account information category) in the openFinance API, if the ASPSP system is a corporate banking system. This is always the case where a Corporate ID is needed as identification of the related corporate in the openFinance API. If this corporate identification is not used in the openFinance API, then the information whether the related account is a corporate account shall be communicated to the API client via the response to the subscription initiation.

If an additional subscription for a push account information service for a dedicated account information category is submitted to the ASPSP by the API Client following the request of a PSU of a certain corporate, where a related valid subscription for a push account information service exists already for this corporate and the related account information category, then this submission will be rejected with a dedicated error code.

5.6 Consent of the PSU

An API Client may execute a transaction at the openFinance API of an ASPSP if it has the necessary consent of the PSU. The consent for initiating subscriptions and of pushing account information towards the API Client will be given by the PSU directly towards the ASPSP by performing an SCA procedure within subscription authorisation. In cases of exemptions, authorisation by the PSU is omitted or alternative authentication procedures might be agreed between PSU and ASPSP and might be integrated into the subscription authorisation flow.

An ASPSP will reject any transaction at the openFinance API if the consent of the PSU cannot be proven and was required at the same time by the ASPSP.

5.7 Decision about strong customer authentication

The ASPSP has to decide

- if SCA has to be executed as part of a subscription transaction at the openFinance API for this extended service,
- which method and personalised credentials have to be used for SCA, where the PSU will be involved in a selection process if several SCA procedures are available, and
- which approach has to be used for executing SCA, taking into account the redirection preference of the API Client.

The API Client has to follow the decision of the ASPSP.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

This rule explicitly applies to the decision of the ASPSP whether and for which content an Update Subscription Entry transaction needs to be authorised by the PSU via SCA.

5.8 Signing baskets

In addition to operational rules on signing baskets in [oFA-OR-FW], the authorisation of an Establish Subscription or an Update Subscription Entry transaction might be signed by the PSU in a signing basket transaction if offered by the ASPSP.

5.9 Avoiding 90 days rule by GDPR consent

Following [RTS] and [RTS-180], a PSU needs to re-authenticate with SCA after every 180 days to retrieve Account Information Data. It is up to the implementation and compliance evaluation of the ASPSP

- to provide the pushing services solely on GDPR basis, not applying the 180 days rule, or
- to check that a related AIS consent authorisation of the PSU with SCA was provided in the openFinance API for the related AISP within the period of the last 180 days and is still valid (within a TPP context), or
- to check that a PSU authentication with SCA was provided in the related PSU-ASPSP interface within the period of the last 180 days (within a direct access context), or
- to let PSUs re-authorise subscriptions at the same time when PSUs are authorising online channel access by using SCA, or
- to use push account information services by pushing triggers for related account information.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

6 Message and data model

In the following, an abstract data model is presented for the specific usage of the Extended Service within the XS2A Interface. The basic abstract data model for the XS2A Interface is defined in [oFA-OR-FW].

A detailed data model for this Extended Service is defined in [oFA-IG-Push].

6.1 Protocol Level

There are no specific requirements on data modelling on protocol level. The following data elements are used independently of the semantic of the related messages, building an abstract basic protocol level.

6.2 Authorisation Data Model

For transactions specific to this Extended Service which mandate authorisation as establish subscription, a uniform authorisation data model is used. To enable such a uniform authorisation, to each transaction to be authorised an authorisation sub-resource is associated. In cases where several authorisations are needed, this sub-resource is repeated.

6.3 Subscription data model

The Push Account Information Service in the XS2A Interface is divided in two different steps – first the establishment of a subscription (initiated by the API Client using the API of the ASPSP) and second the pushing of the data as such (initiated by the ASPSP using the API of the AISP).

Remark: These two steps are implemented through different APIs – the /subscriptions as part of the API of the ASPSP and the entry API or the entry trigger API of the AISP, cp. [oFA-IG-Push].

6.3.1 Establish subscription transaction

Within the XS2A Interface, an Establish subscription transaction always starts with the Establish Subscription Request and the Establish Subscription Response. For the Decoupled, Redirect or OAuth2 SCA Approach, there must be at least a second message pair Subscription Status Request and Subscription Status Response within the openFinance API to retrieve the information whether the SCA method was successful. In all cases, the ASPSP may ask the API Client to update the subscription resource created after the Establish Subscription Request with additional data via an Update Data Request.

In case of the Embedded SCA Approach, a dedicated message pair consists of the Transaction Authorisation Request and the Transaction Authorisation Response for processing PSU credentials directly within the openFinance API. This message pair is conditional, depending on the result of the ASPSP's risk management on SCA necessity. It can be repeated in case of a non-successful SCA of the PSU.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

6.3.1.1 Establish Subscription Request

Remark: Only the requests are modelles, since the responses are only transporting the equivalent resource and authorisation related data as described for the authorisation of other resources in [oFA-OR-FW].

6.3.1.1.1 PSU Data

- PSU Identification (conditional, only if mandated by parameters published by the ASPSP)
- PSU Corporate Identification and Type (conditional, only if mandated by parameters published by the ASPSP and only if PSU is a corporate)
- PSU Risk Management Data. If not included in the message the ASPSP will take this into account in its risk management.
  - IP Address PSU (mandatory)
  - PSU Device and Application Software Information (operating system, browser etc.) (optional),
  - GEO Location PSU (optional)

6.3.1.1.2 API Client data

- Redirect Preferred Indicator (optional)

  With this indicator, the API Client can set its priority for a re-direct based SCA Approach (Redirect SCA Approach or OAuth2 SCA Approach) vs. a SCA Approach without a re-direction to a bank site (Embedded SCA Approach or Decoupled SCA Approach, depending on the authentication method).

- Explicit Authorisation Preferred Indicator (optional)

  With this indicator, the API Client can set its priority to create an authorisation sub-resource explicitly. This should be used by the API Client in cases where the corresponding transaction is put into a signing basket later.

- Redirect URL-API Client (conditional, only mandated if the Redirect Preferred Indicator equals true or if this Indicator is not contained)

  This data element defines an URL to which the ASPSP shall redirect the PSU browser session once the SCA on bank websites is performed.

6.3.1.1.3 Subscription Data

- General subscription information:

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

- Account Information Category (i.e. which of the account information categories account entries, account statements, account balances or incoming RTPs to be pushed)
- Preference on application layer encryption for the push service (optional)
- Encryption certificate (optional)

- One or more so called subscription entries for push services e.g. for different accounts or resulting from different trigger criteria.

  Each subscription entry contains the following information

  - Related account reference (mandatory)
  - A subscription entry name for displaying on the ASPSP site to the PSU. (optional)
  - Callback Preference Hyperlink (Optional)

    A preference of the AISP to receive the information only as a hyperlink. This preference might be ignored by the ASPSP if not supported.
  - Callback Preference Static (Optional)

    A preference of the AISP to receive the information only with a static text as defined by the AISP. This preference might be ignored by the ASPSP if not supported.
  - Callback Static Text (Optional)

    The static text to be used by the ASPSP for the upper preference.
  - Primary Push URI (Mandatory)

    Information for the AISP entry interface, where related account information shall be pushed to. (mandatory)
  - Secondary Push URI (Optional)

    Information for the AISP entry interface, where related account information shall be pushed to, if the Primary Push URI is not reachable.
  - Account Information Category Parameter (Mandatory)

    Parameters steering the content of the actual push service. These parameters consist of account information category related trigger criteria, cp. Section 6.3.1.5, as well as additional parameters, cp. Section 6.3.1.6.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

- Validity (optional)

  The end date of the validity of the subscription. If not provided it will be valid till cancellation of this subscription entry.

6.3.1.2 Add Subscription Entries Request

The PSU and API Client Data are identical to the related data defined in Sections 6.3.1.1.1 and 6.3.1.1.2.

To match the content of this request with existing subscriptions, the following parameter needs to be transported:

- Subscription Identification (Mandatory)

  A reference to the related subscription.

In addition, the following data needs to be transmitted:

6.3.1.2.1 Subscription Entry Data

- One or more so called subscription entries for push services e.g. for different accounts or resulting from different trigger criteria.

  Each subscription entry contains the following information

  - Related account reference (mandatory)
  - Subcription Entry Name

    A subscription entry name for displaying on the ASPSP site to the PSU. (optional)
  - Callback Preference Hyperlink (Optional)

    A preference of the AISP to receive the information only as a hyperlink. This preference might be ignored by the ASPSP if not supported.
  - Callback Preference Static (Optional)

    A preference of the AISP to receive the information only with a static text as defined by the AISP. This preference might be ignored by the ASPSP if not supported.
  - Callback Static Text (Optional)

    The static text to be used by the ASPSP for the upper preference.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

- Primary Push URI (Mandatory)

  Information for the AISP entry interface, where related account information shall be pushed to. (mandatory)

- Secondary Push URI (Optional)

  Information for the AISP entry interface, where related account information shall be pushed to, if the Primary Push URI is not reachable.

- Account Information Category Parameter (Mandatory)

  Parameters steering the content of the actual push service. These parameters consist of account information category related trigger criteria, cp. Section 6.3.1.5, as well as additional parameters, cp. Section 6.3.1.6.

- Validity (optional)

  The end date of the validity of the subscription. If not provided it will be valid till cancellation of this subscription entry.

6.3.1.3 Subscription 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 in request or response.

6.3.1.4 Subscription Details Request and Response

This request is addressed on a created resource and requesting to retrieve the details of the subscription 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 subscription (partially or implicitly) via the PSU ASPSP interface. The request contains no specific data elements.

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

6.3.1.5 Account information category related trigger criteria

The following trigger criteria are supported for the various account information categories:

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

Account entries

- A so called account entry filter for filtering the account entries to be pushed. (mandatory)

  Filter criteria can be a.o. bank transaction codes, amount limits, partner accounts, entry status, debted or credited transactions.

- A so called attribute selection preference, restricting the attributes of the ASPSP to be sent to the attributes contained in the selection preference. The aim of this is to reduce data to be transmitted. This restriction preference might be ignored by the ASPSP if not supported.

Account Statements

A so called account statement trigger criteria, for defining the account statements to be pushed

- Criteria can be the required report or statement type related to camt.05x or MT94x messages (mandatory)

- Criteria can be an event entry, e.g. stating that reports should be delivered "after generation" in the ASPSP backend. (optional)

- Criteria can also be a time slot, asking for statements at a certain pre-defined time. (optional)

Account Balance

A so called **balance trigger criteria** for defining the balances to be pushed

- criteria can be balance related data like balance amounts with related operators like less or greater than a predefined amount, balance types (optional)

- criteria can also be a time slot, asking for balance at a certain pre-defined time (optional).

Incoming RTPs

A co called **RTP trigger criteria**, for defining the incoming RTPs to be pushed

- Criteria can be an event entry, e.g. stating that reports should be delivered "after reception" in the ASPSP system. (optional)

- Criteria can also be a restriction only to push RTPs with a short term, pre-defined validity period (optional)

- Criteria can also be a reminder for long-lasting RTPs with a pre-defined period before the validity of the RTP expires. (optional)

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

6.3.1.6 Additional Parameters related to account information categories

For the following account information categories there are additional parameters defined:

Account entries

- Accepted format for transaction reports. (Mandatory)
- A so called attribute selection preference, restricting the attributes of the ASPSP to be sent to the attributes contained in the selection preference. The aim of this is to reduce data to be transmitted. This restriction preference might be ignored by the ASPSP if not supported (optional)
- A preference to receive also documents attached to an account entry, might be ignored by the ASPSP. (optional)

Incoming RTPs

- Accepted format for transaction reports. (Mandatory)
- A preference to receive also documents attached to an account entry, might be ignored by the ASPSP. (optional)

6.3.2 Push account information transaction

Within the XS2A Interface a Push account information transaction is usually only one pair of a Push account information request and response.

The Push Account Information Request is addressing the related resource in the entry URI of the AISP. The modelling for the JSON based reporting is separating generic push data and account information category specific data as described below. The MT94x and camt.05x reports to be pushed contain the related account information data within the related formats.

6.3.2.1 Generic push data (JSON)

The modelling of the generic push data for the JSON based reporting is as follows:

- Subscription-ID (mandatory)

  The identification of the underlying subscription resource.

- Subscription Entry ID (mandatory)

  The identification of the underlying subscription entry.

- Account (mandatory)

  Account where the related entries where performed in the ASPSP system.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

- Date and Time of the Last Push (mandatory)

  Date and Time of the last push which has been triggered by the related subscription.

6.3.2.2 Account information category specific push data (JSON)

The modelling of the account information category specific push data for the JSON based reporting is as follows:

Account entries

- A debit accounting flag (optional)

  If not supporting, debit account is default (i.e. `false`, i.e. debted entries are negative.)

- Transactions (mandatory)

  One or more entries to be pushed to the API Client.

Account Statements

- Account Statement (mandatory)

  Account statement related to a dedicated account.

Account Balance

- Account Balance (mandatory)

Incoming RTPs

- Transactions (mandatory)

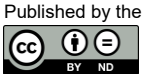
  One or more entries with a specific booking status and related RTP details to be pushed to the API Client.

6.3.3 Push account information trigger transaction

Within the XS2A Interface a Push account information transaction is usually only one pair of a Push account information request and response.

The Push Account Information Request is addressing the related resource in the entry URI of the AISP.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License


(ref. License Notice for full license conditions)

The modelling for the JSON based reporting is as follows:

- Subscription-ID (mandatory)

  The identification of the underlying subscription resource.

- Subscription Entry ID (mandatory)

  The identification of the underlying subscription entry.

- Account (conditional)

  Account where the related entries where performed in the ASPSP system. Only used when a trigger text is sent to the API Client.

- Trigger Text (conditional)

  A static text defined in the related Submit Subscription Entry Request.

- Hyperlink to the related account endpoint (conditional)

  A hyperlink to the account where new entries have been performed in the ASPSP system.

- Date and Time of the Last Push (mandatory)

  Date and Time of the last push which has been triggered by the related subscription.

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

7 Annex

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].

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].

QTSP

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

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

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 openFinance API 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.

7.2 References

[oFA-IG-Push] openFinance API Framework, Push Account Information Services, Implementation Guidelines Version 2.3, 31 March 2026

[oFA-OR-FW] openFinance API Framework, Operational Rules, Basic Operational Rules of the Framework, Version 2.1, 31 March 2026

[oFA-OR-Adm] openFinance API Framework, Administration Services, Operational Rules, Version 0.9, published for market consultation 26 May 2021

[oFA-PFSM] openFinance API Framework, Implementation Guidelines, Protocol Functions and Security Mechanisms, Version 2.4, 31 March 2026

[oFA-DaD] openFinance API Framework, Data Dictionary, Version 2.4, 31 March 2026

[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

[RTS-180] 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

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)

[EBA-OP] Opinion of the European Banking Authority on the implementation of the RTS on SCA and CSC, EBA-Op-2018-04, 13.06.2018

[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, 4 June 2020

[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

[TS 119 495] Draft ETSI TS 119 495, Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Qualified Certificates Profiles and TSP Policy Requirements under the Payment Service Directive 2015/2366/EU, V1.1.2 (2018-07)

[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

Figure 1: API Service Establish push account entry subscription 12
Figure 2: API Service Update push account information subscription 16
Figure 3: API Service Cancellation of a subscription 17
Figure 4: API Service Notification of subscription/subscription entry cancellation 18
Figure 5: API Service Push account entries of a dedicated account 19
Figure 6: API Service Push account information of a dedicated account to a fallback URI ... 20
Figure 7: API Service Push account triggers of a dedicated account 21

7.4 List of tables

Table 1: Extended services scheduled for further detailed work 7
Table 2: API services for the Push Account Information Service 10
Table 3: Consent of the PSU within API Service 23

Published by the Berlin Group under Creative Commons Attribution-NoDerivatives 4.0 International Public License

(ref. License Notice for full license conditions)