THE Berlin GROUP

A EUROPEAN STANDARDS INITIATIVE

openFinance API Framework

Implementation Guidelines for Administrative Services

First version without definition of data structure for price lists

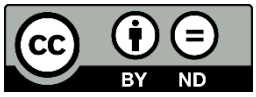
Version 1.1

9 September 2024

License Notice

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

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

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

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

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

Contents

1 Introduction

1.1 From Core XS2A Interface to openFinance API

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

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

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

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

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

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

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

The following account access methods are covered by this framework:

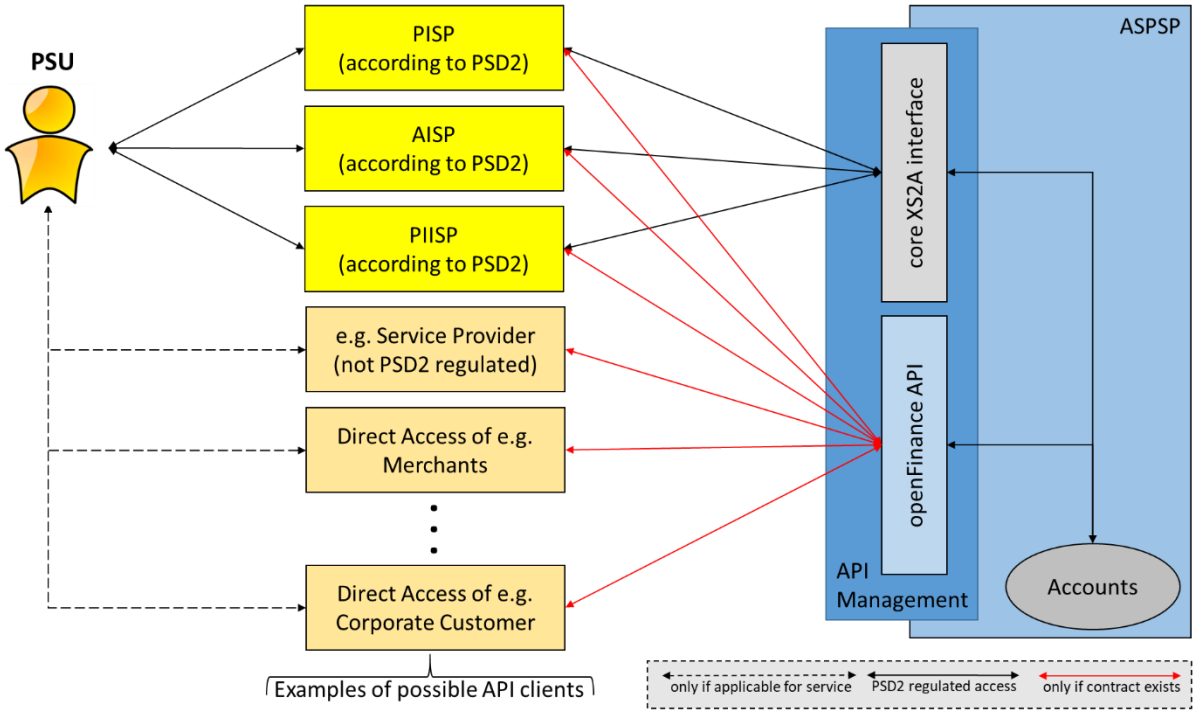


Figure 1: Core XS2A interface and openFinance API

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

1.2 Administrative Services

In contrast to the compliance API of an ASPSP the access to the openFinance API of an ASPSP with the extended services supported by the ASPSP is not open to all possible API clients. Instead an API Client has to conclude corresponding contracts (either with a scheme or bilaterally with an ASPSP) before he may access the extended services supported by an ASPSP. Which contracts are necessary will be determined either by an access scheme or the ASPSP itself (if no access scheme exists).

After conclusion of necessary contracts an API Client has to execute a technical onboarding at the openFinance API of an ASPSP. As part of the technical onboarding the ASPSP can check if all necessary contractual steps have been finished before. If everything is ok the ASPSP will assign a unique apiContractId to the API Client and will return this to the API Client as response of the technical onboarding. The API Client has to use this apiContractId for all further accesses to the extended services at the openFinance API of the ASPSP.

Currently these implementation guidelines for Administrative Services describe only the services for the technical onboarding of an API Client at the openFinance API of an ASPSP.

Please notice that [oFA-OR-ADM] contains more services which are currently not contained in these implementation guidelines.

For the technical onboarding a "one step process" and a "two step process" are distinguished.

The one step process shall be used if all conditions of the access to the extended services have been determined finally as part of the contracts concluded before. In this case nothing has to be accepted or confirmed by the API Client as part of the technical onboarding.

In contrast the two step process allows to ask the API Client for an explicit confirmation of special conditions determined by the ASPSP as part of the technical onboarding. Special price conditions for using the extended services of an ASPSP may be an example.

Remark:

> If a scheme determines the rules and conditions for using the extended services of its participants it is up to the scheme to determine if only the one step process shall be supported for the technical onboarding or if also the two step process will be possible. If no scheme exists this has to be decided by the ASPSP.

Remark on version 1.1 of this document:

> The new endpoint "onboardings/{onboardingId}/certificates" has been added.
>
> All POST request messages used for the administrative services have to be signed http messages using the certificate (QsealC) of the API Client.
>
> If the API Client will change this certificate for message signing after his onboarding with an ASPSP has been finished, he has to inform the ASPSP about this using this new endpoint.

1.3 Document Structure

This document specifies the Administrative Services in detail.

Section 2 is providing specific, but still abstract information on the application layer of this service like service API access methods or service specific error codes.

Section 3 defines the API for Administrative Services in detail.

1.4 Document History

Version	Change/Note	Approved
28.12.2023	Final draft version for internal consultation	
21.02.2024	Final Version	21.02.2024 openFinance TF
17.07.2024	Draft ve 1.1: new endpoint has been added "onboardings/{onboardingId}/certificates" has Confirmation of a price list change from PUT method to POST method	
9.09.2024	Version 1.1 final: No further changes, only references adapted	9.09.2024 BWG

2 Application Layer: Guiding Principles

2.1 Signing Messages at Application Layer

For the administrative services all POST request messages according to section 3.1, section 3.3 and section 3.7 have to be signed HTTP request messages as described in section 6 of [oFA-ProtSec].

For these request messages the certificate (QSealC) of the API Client has to be included into the JWS Protected Header (element x5c), if the certificate has not already been exchanged with the ASPSP before by some other means.

If the API Client will change his certificate after the onboarding has been finished, he has to inform the ASPSP by a POST request message according to section 3.7. This POST request message has to be signed by the API Client using his **new** certificate. The new certificate of the API Client has to be included into the JWS Protected Header (element x5c) of the POST request message, if the new certificate has not already been exchanged with the ASPSP before by some other means.

2.2 Overview: Transaction flow for the onboarding process

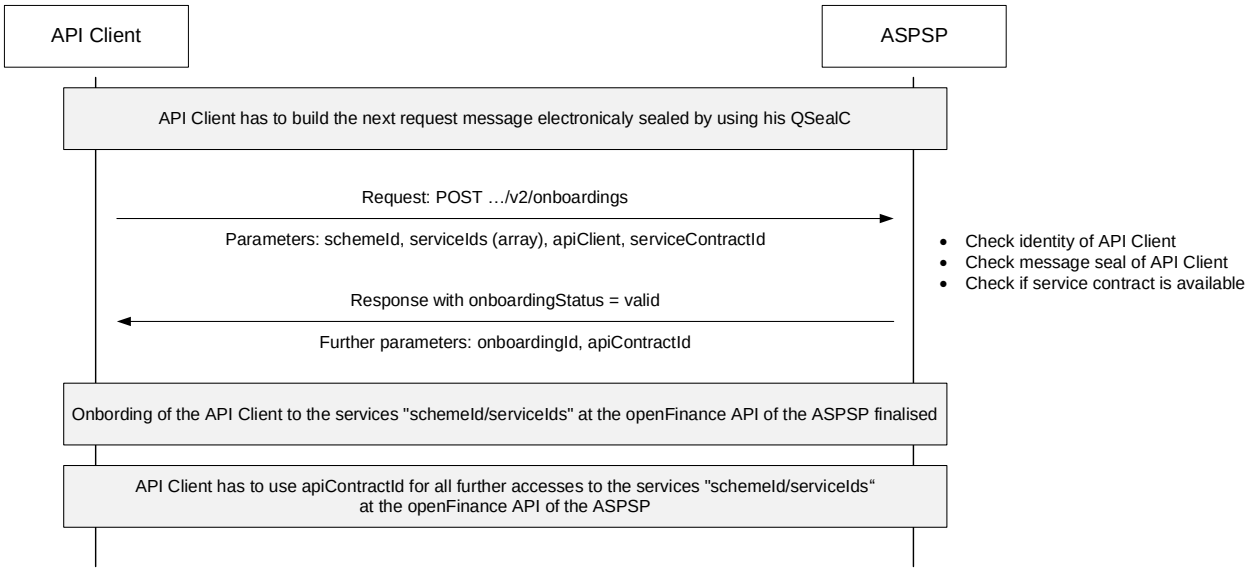
For the onboarding process of an API Client at the openFinance API of an ASPSP a "one step process" and a "two step process" can be distinguished and are described by this specification.

It will be decided by a scheme if the one step process or the two step process has to be used by an API Client for onboarding to an openFinance API of an ASPSP as participant of the scheme. If no scheme is involved this will be decided by the ASPSP.

The API Client is informed by the response body to the first POST request message if a one step process or a two step process has to be used.

The following figures will show a high-level overview of the processes for the two cases.

One step process without confirmation of (price) conditions at the API



For this use case the onboarding process consist only of one simple POST request sent by the API Client to the onboarding endpoint of the openFinance API of the ASPSP.

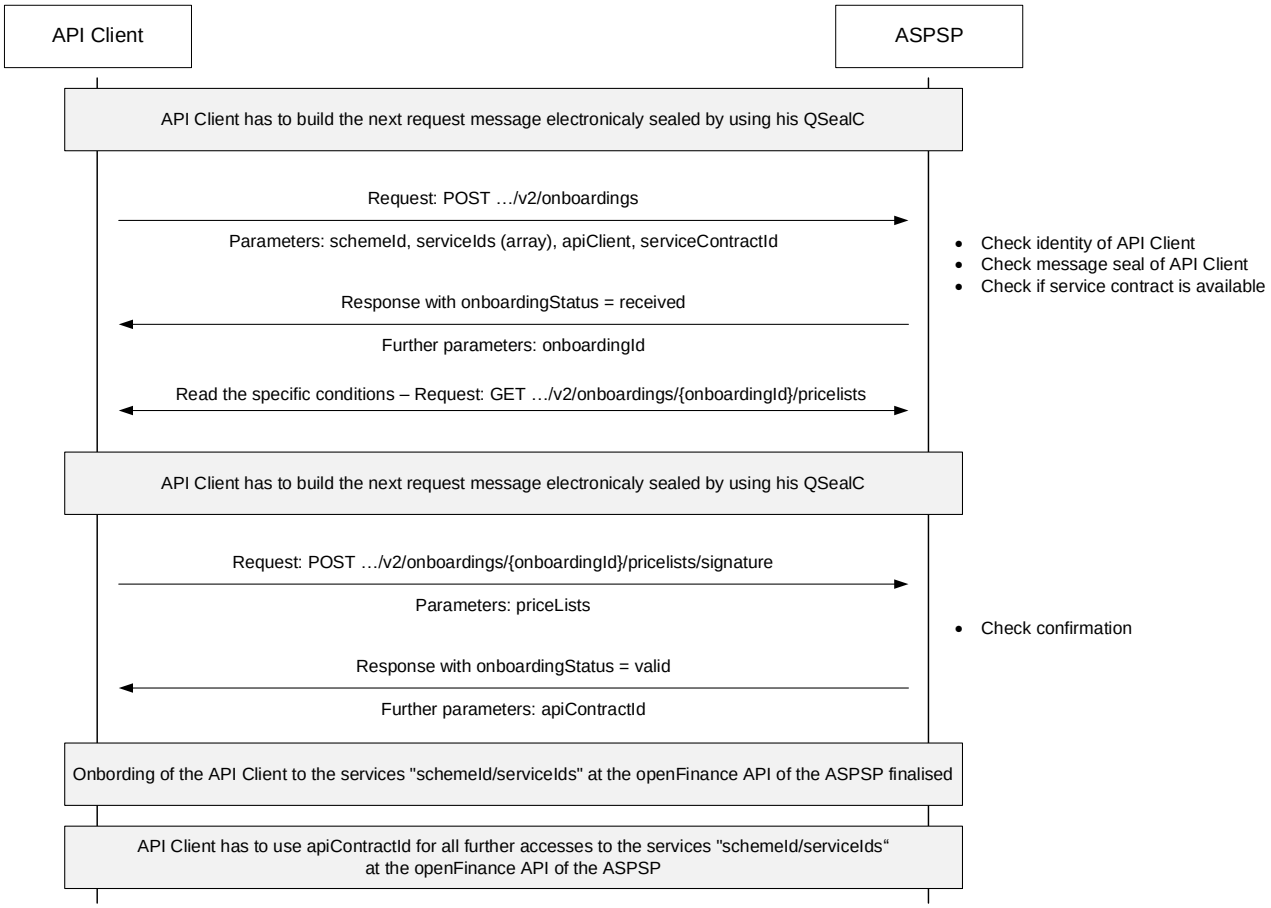
If the onboarding is successful an onboarding resource is created and the ASPSP will respond with status `valid`. The attribute `validTo` of the onboarding resource will be set depending on the validity of the service contract. In this case the ASPSP will also provide an UUID as `apiContractId`. The API Client shall use this UUID for further accesses to the corresponding services at the openFinance API of this ASPSP.

Remark:

> If the service contract has got an unlimited validity the attribute `validTo` will be set to 9999-12-31.

If the onboarding is not successful no onboarding resource will be created and the ASPSP will respond with a HTTP Response Code according to [oFA-ProtSec].

Two step process with confirmation of (price) conditions at the API



For this use case the onboarding process consist of two POST requests sent by the API Client to the onboarding endpoint of the openFinance API of the ASPSP.

If the first POST is successful, an onboarding resource is created and the ASPSP will respond with resource status `received`. In addition links indicating the next steps are returned by the ASPSP. The value of the attribute `validTo` of the onboarding resource is irrelevant at this time.

If the POST is not successful no onboarding resource will be created and the ASPSP will respond with a HTTP Response Code according to [oFA-ProtSec]. In this case this process is finished.

If the POST returns with status `received` the API Client can read the special (price) conditions (JSON data structure) from the onboarding resource created by the POST request. The API Client can check the conditions. If OK he has to send the second POST request to confirm his acceptance to the ASPSP. This second POST request also contains a signature calculated for the HTTP request message as described by [oFA-ProtSec].

If the second POST (confirmation step) is successful the ASPSP will respond with resource status `valid`. The attribute `validTo` of the onboarding resource will be set according to the validity of the price conditions confirmed by the second POST (see section 3.3). In this case the ASPSP will also provide an UUID as `apiContractId`. The API Client shall use this UUID for further accesses to the corresponding services at the openFinance API of this ASPSP.

If the second POST (confirmation step) is not successful, the ASPSP will respond with the resource status `rejected`. This could be for example the case if the signature of the HTTP request message of the API Client or further content of the second POST message could not be verified successfully by the ASPSP. The API Client needs to start the onboarding process again after a potential error in his implementation has been fixed.

2.3 Overview: Status changes for an onboarding resource

If the first POST command of an onboarding process is executed successfully, an onboarding resource is created. Beside other attributes this onboarding resource can contain the attributes validTo (indicating the validity period of the corresponding onboarding) and apiContractId (indicating the id to be used by the API Client to access the corresponding services at the openFinance API of the ASPSP).

The possible values for the status of an onboarding resource (see [oFA DaD]) and the possible events triggering a transition of the status are different depending on the case of the onboarding process, i.e. with or without confirmation of special price conditions, and the business logic given by the contracts. The transition of the status and of the value of the `validTo` attribute can be triggered for example by

- a second POST command received for this onboarding resource at the openFinance API to confirm the price list (section 3.3),
- a DELETE command received for this onboarding resource at the openFinance API (section 3.6),
- an external event to be reflected by the onboarding resource, i.e. termination of the corresponding service contract or replacement of the special price conditions by the ASPSP, or

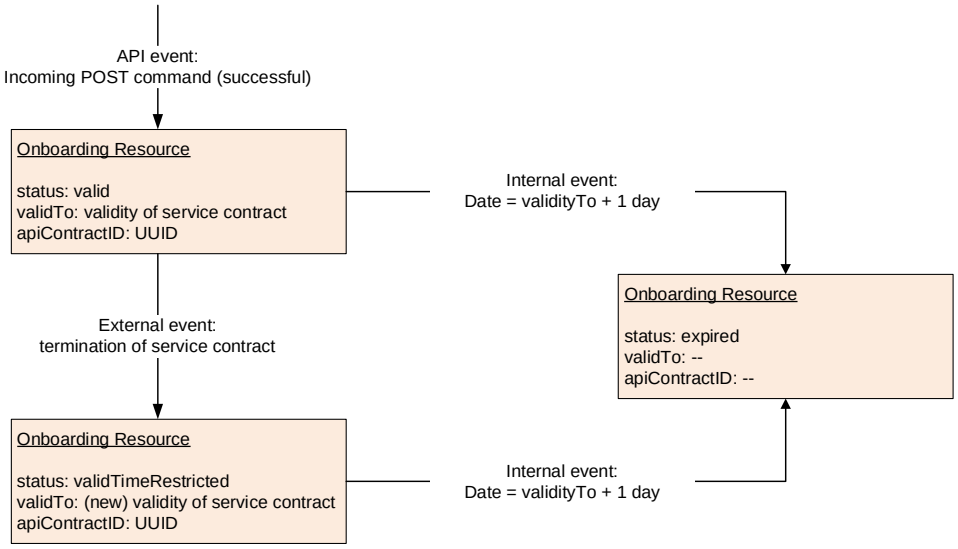
- an internal event of the implementation, i.e. the termination of the resource due to the expiration of its validity.

Possible external events depend on the business logic which is define by ASPSP and API Client within the contracts. These external events are not the topic of this implementation guidelines. The same holds for the usage of the validTo parameter. The conditions for changing the value of this parameter depend on the business logic.

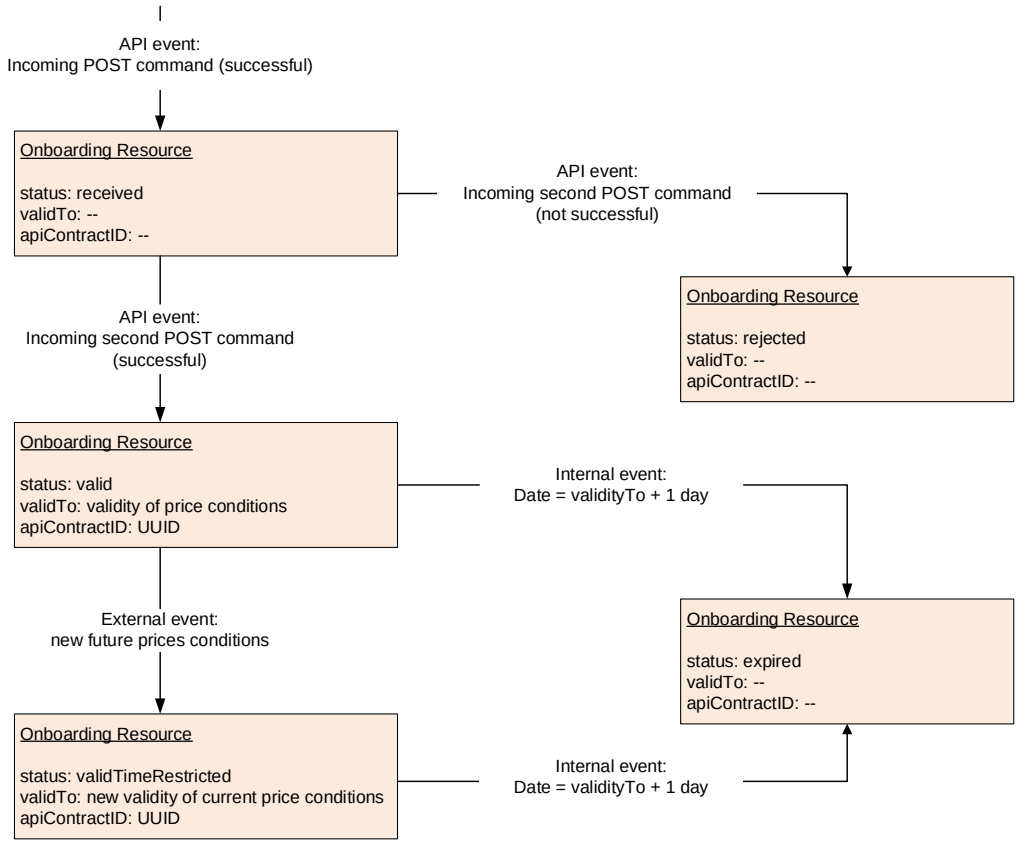
If a POST command is received (at the endpoint `onboardings/{onboardingId}/certificates`) for an onboarding resource at the openFinance API to inform about a new certificate (QSealC) of the API Client (see section 3.7), the status of the onboarding resource will not change. If this POST command is successful, the ASPSP will store the new certificate for the API Client. If this POST command is not successful (i.e. if the signature or the new certificate cannot be verified successfully), the ASPSP will reject this message. In both cases the status of the onboarding resource will not be changed.

The following figures give an overview of possible transitions of an onboarding resource:

One step process without confirmation of (price) conditions at the API



Two step process with confirmation of (price) conditions at the API



Remark:

The validity of an onboarding resource starts always on the day its status is set to valid. It is not possible to execute an onboarding with a validity starting in the future.

Nevertheless, the response body to a POST request may contain an optional validFrom parameter for implementation reasons. This parameter indicates the date from which the API Client may access the corresponding services at the openFinance API of the ASPSP.

2.4 Endpoints and API Access Methods

The following table gives an overview on the HTTP access methods supported by the API endpoints for the Administrative Services. The conditions are relative and are provided for the case that the related parent endpoint exists.

Endpoints/Resources	Method	Condition	Description
onboardings	POST	Mandatory	Create an onboarding resource addressable under {onboardingId}. **Note:** All POST request messages sent to this endpoint shall be sealed by the API Client using his QSealC. See section 3.1.
onboardings/{onboardingId}/pricelists	GET	conditional	Read the data structure containing one or more price lists valid for using the corresponding extended services by the given API Client. Shall be supported if the two step approach is required by the business definitions. See section 3.2.
onboardings/{onboardingId}/pricelists/signature	POST	conditional	POST a signature to confirm the price lists contained in the data structure read before. Shall be supported if the two step approach is required by the business definitions. See section 3.3.
onboardings/{onboardingId}/certificates	POST	Optional	POST a signature (based on a new certificate (QSealC)) to inform about a new certificate used by the API Client for http-message signing. See section 3.7.
onboardings/{onboardingId}/status	GET	Mandatory	Read the status of the onboarding resource. See section 3.4.
onboardings/{onboardingId}	GET	Mandatory	Read the onboarding resource See section 3.5.
onboardings/{onboardingId}	DELETE	Optional	Cancel the onboarding. See section 3.6.

The support of the endpoint `onboardings/{onboardingId}/certificates` is optional for an ASPSP. If an ASPSP supports this endpoint the following requirements hold:

- The ASPSP shall also support to receive signed http messages where the JWS Protected Header does not contain the certificate in the element x5c but contains only a hash value of the certificate in the element x5t#S256. See section 6 of [oFA-ProtSec].

- The ASPSP informs about the support of this endpoint using his Discovery API.

- The ASPSP can require that an API Client has to include the hash value of the certificate (using the element x5t#S256) instead of the certificate itself (using the element x5c) into the JWS Protected Header if he sends a signed http message. In this case a signed http request message containing the certificate in the JWS protected header will be rejected by the ASPSP.

- The ASPSP will inform using his Discovery API if he accepts for signed http messages, that the certificate is included in the JWS Protected Header (using the element x5c) or if the API Client has to include the hash value of the certificate using the element x5t#S256 instead.

- The last two points do not hold for the POST request messages defined as part of the Administrative Services defined in this document. For these signed http request messages, the requirements as defined by section 2.1 hold.

3 Administrative Service

For all access methods defined in this document further applicable header parameters and error handling are defined in [oFA-ProtSec]. In addition, also the authentication of the API Client is done based on a QWAC as described in the above document. Data structures are used as defined in [oFA DaD].

3.1 Initiation of the onboarding process

Call

`POST …/v2/onboardings`

Creates the onboarding resource at the ASPSP.

Path Parameters

No Path Parameters

Query Parameters

No Query Parameter

Request Header

No special Request Header parameter

Request Body

Attribute	Type	Condition	Description
schemeId	Max70Text	Mandatory	Identification of the corresponding scheme, in case of no scheme involved use constant `BILATERAL` Example: giroAPI-payment The schemeIds supported by the ASPSP should be provided in the Discovery API [oFA Discov].
serviceIds	Array of Service Type	Mandatory	Identification of the extended services for which the onboarding will be done. Possible values to identify a service have to be defined by the service owner (scheme or ASPSP). The service types supported by the ASPSP should be provided in the Discovery API [oFA Discov].

Attribute	Type	Condition	Description
apiClientContacts	Array of Contact	Optional	Contacts (of different categories) of the API Client. Only to be given if not already available by e.g. a directory service.
serviceContractId	Max70Text	Mandatory	Identification of the service contract concluded between the service owner (scheme or ASPSP) and the API Client to use these services.

Response Code

The HTTP response code equals 201.

Response Header

No special Response Header parameter

Response Body

Attribute	Type	Condition	Description
onboardingId	Max70Text	Mandatory	Unique identification of the onboarding resource created by this initiation.
onboardingStatus	Onboarding Status	Mandatory	Status of the onboarding process.
apiContractId	UUID	Conditional	Unique identification of the successful onboarding of the API Client for the list of services. Shall be used by the API Client if accessing the services after the onboarding. The attribute apiContractId shall only be provided if the onboarding resource has been created and the resource status is set to `valid`. Otherwise this attribute shall be missing.
validTo	ISODate	Conditional	Date until when the successful onboarding is valid. The attribute validTo shall only be provided if the onboarding resource has been created and the resource status is set to `valid`. Otherwise this attribute shall be missing.

Attribute	Type	Condition	Description
			If the validity of the created resource is unlimited, the value 9999-12-31 shall be provided.
validFrom	ISODate	Optional	Indicates the date after which the API Client can access the services at the openFinance API of the ASPSP using the apiContractId returned with this response.
_links	Links	Conditional	A list of hyperlinks to the next steps to read a price list and to confirm a price list. The attribute _links shall only be provided if the technical onboarding requires the execution of the two step process according to section 2.2. Otherwise this attribute shall be missing. If this attribute is provided it shall contain the following link: • readConditions • confirmConditions **Remark about future versions:** Future versions of the data dictionary [oFA DaD] will contain the definition of these href types.

Remark:

The parameters onboardingId and apiContractId are two identifiers. The onboardingId is a technical resource identifier, which will always be provided for the first step if successful (and which is recommended to be a UUID). The apiContractId is a UUID provided only after successful completion of the whole onboarding process, which might need more than one API interaction in the future.

Example

Request

```
POST https://api.testbank.com/openfinance/v2/onboardings
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Thu, 28 Dec 2023 15:02:37 GMT
Digest:                 see section 6.2.2.1 of [oFA-ProtSec]
x-jws-signature:        see section 6.2.2.2 of [oFA-ProtSec]

{
```

```
  "schemeId": "giroAPI-Payment",
  "serviceIds": ["XFPIS", "XDFPIS", "XMDFPIS"],

  "serviceContractId": "giroAPI-Contract-ab10cd19ef58-20231229"
}
```

Response for a one step onboarding process

```
HTTP/1.x 201 Created
X-Request-ID:          99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                  Thu, 28 Dec 2023 15:02:42 GMT
Location:
      https://www.testbank.com/openfinance/v2/onboardings/3d9a81b3-a47d-
4130-8765-a9c0ff861100
Content-Type:          application/json

{
  "onboardingId": "3d9a81b3-a47d-4130-8765-a9c0ff861100",
  "onboardingStatus": "valid",
  "apiContractId": "24a990c2-e990-42f0-8cc4-56c822d4e2ec",
  "validTo": "2025-12-31",
  "validFrom": "2024-01-01"
}
```

Response for a two step onboarding process

```
HTTP/1.x 201 Created
X-Request-ID:          99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                  Thu, 28 Aug 2023 15:02:42 GMT
Location:
      https://www.testbank.com/openfinance/v2/onboardings/3d9a81b3-a47d-
4130-8765-a9c0ff861100
Content-Type:          application/json

{
  "onboardingId": "3d9a81b3-a47d-4130-8765-a9c0ff861100",
  "onboardingStatus": "received",
  "_links": {
     "readConditions":
           {"href": "/openfinance/v2/onboardings/3d9a81b3-a47d-4130-8765-
a9c0ff861100/pricelists"},
     "confirmConditions":
           {"href": "/openfinance/v2/onboardings/3d9a81b3-a47d-4130-8765-
a9c0ff861100/pricelists/signature"}
            }
}
```

3.2 Reading price lists

Call

```
GET …/v2/onboardings/{onboardingId}/pricelists
```

The secure authentication of the API Client allows the ASPSP to present the (potentially confidential) price list applicable exactly to the API Client.

Path Parameters

Attribute	Type	Description
onboardingId	UUID	Onboarding ID as returned with the response message to the initiation of the onboarding process according to section 3.1.

Query Parameters

No Query Parameter

Request Header

No special Request Header parameter

Request Body

No Request Body

Response Code

The HTTP response code equals 200.

Response Header

No special Response Header parameter

Response Body

Attribute	Type	Condition	Description
priceLists	Price List	Mandatory	JSON data structure containing information about fees to be paid for using the services at the openFinance API of the ASPSP **Remark:** This data type is a dummy for this current version of these implementation guidelines. It will be defined and added for a later version.

Example

Request

```
GET https://api.testbank.com/openfinance/v2/onboardings/3d9a81b3-a47d-4130-8765-a9c0ff861100/pricelists
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7725
Date:                   Thu, 28 Dec 2023 15:03:15 GMT
```

3.3 Confirmation of price lists

Call

```
POST …/v2/onboardings/{onboardingId}/pricelists/signature
```

Changes the status of the onboarding resource at the ASPSP, which has been created by the first POST call according to section 3.1.

This POST command has to be secured by HTTP message signing. In case of signature failure an http Response Code 401 with a message code SIGNATURE_INVALID will be returned according to error handling defined in [oFA-ProtSec]. See also [oFA DaD] for related message codes.

Path Parameters

Attribute	Type	Description
onboardingId	UUID	Onboarding ID as returned with the response message to the initiation of the onboarding process according to section 3.1.

Query Parameters

No Query Parameter

Request Header

No special Request Header parameter

Request Body

To confirm the price list read according to section 3.2 the price list has to be return in the Request Body without any changes.

Attribute	Type	Condition	Description
priceLists	Price List	Mandatory	Copy of the price lists read before without any changes.

Response Code

The HTTP response code equals 200.

Response Header

No special Response Header parameter

Response Body

Attribute	Type	Condition	Description
onboardingStatus	Onboarding Status	Mandatory	Status of the onboarding process.
apiContractId	UUID	Mandatory	Unique identification of the successful onboarding of the API Client for the list of services. Shall be used by the API Client if accessing the services after the onboarding.
validTo	ISODate	Mandatory	Date until when the successful onboarding is valid.
validFrom	ISODate	Optional	Indicates the date after which the API Client can access the services at the openFinance API of the ASPSP using the apiContractId returned with this response.

Example

Request

```
POST https://api.testbank.com/openfinance/v2/onboardings/3d9a81b3-a47d-4130-8765-a9c0ff861100/pricelists/signature
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7727
Date:                   Thu, 28 Dec 2023 15:33:05 GMT
Digest:                 see section 6.2.2.1 of [oFA-ProtSec]
x-jws-signature:        see section 6.2.2.2 of [oFA-ProtSec]

{
  "priceLists": "json structure to be defined",
}
```

Response

```
HTTP/1.x 200 ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7727
Date:                   Thu, 28 Dec 2023 15:33:35 GMT
Content-Type:           application/json

{
  "onboardingStatus": "valid",
  "apiContractId": "24a990c2-e990-42f0-8cc4-56c822d4e2ec",
  "validTo": "2025-12-31",
  "validFrom": "2024-01-01"
}
```

3.4 Reading the status of the onboarding

Call

```
GET …/v2/onboardings/{onboardingId}/status
```

Path Parameters

Attribute	Type	Description
onboardingId	UUID	Onboarding ID as returned with the response message to the initiation of the onboarding process according to section 3.1.

Query Parameters

No Query Parameter

Request Header

No special Request Header parameter

Request Body

No Request Body

Response Code

The HTTP response code equals 200.

Response Header

No special Response Header parameter

Response Body

Attribute	Type	Condition	Description
onboardingStatus	Onboarding Status	Mandatory	Current status of this onboarding resource.

Example

Request

```
GET https://api.testbank.com/openfinance/v2/onboardings/3d9a81b3-a47d-4130-8765-a9c0ff861100/status
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7729
Date:                   Fri, 29 Dec 2023 15:03:15 GMT
```

3.5 Reading the onboarding resource

Call

```
GET …/v2/onboardings/{onboardingId}
```

Path Parameters

Attribute	Type	Description
onboardingId	UUID	Onboarding ID as returned with the response message to the initiation of the onboarding process according to section 3.1.

Query Parameters

No Query Parameter

Request Header

No special Request Header parameter

Request Body

No Request Body

Response Code

The HTTP response code equals 200.

Response Header

No special Response Header parameter

Response Body

Attribute	Type	Condition	Description
onboardingStatus	Onboarding Status	Mandatory	Current status of this onboarding resource.
schemeId	Max70Text	Mandatory	Identification of the corresponding scheme, in case of no scheme involved use constant `BILATERAL` Example: giroAPI-payment
serviceIds	Array of Service Type	Mandatory	Identification of the extended services for which the onboarding will be done. Possible values to identify a service have to be defined by the service owner (scheme or ASPSP).
serviceContractId	Max70Text	Mandatory	Identification of the service contract concluded between the service owner (scheme or ASPSP) and the API Client to use these services.
apiContractId	UUID	Conditional	Unique identification of the successful onboarding of the API Client for the list of services contained in serviceIds of the request body. Shall be used by the API Client if accessing the services after the onboarding. Has to be returned if the apiContractId is set within the onboarding resource.
validTo	ISODate	Conditional	Date until when the successful onboarding is valid.

Attribute	Type	Condition	Description
			Has to be returned if the attribute validTo is set within the onboarding resource.
validFrom	ISODate	Conditional	Indicates the date after which the API Client can access the services at the openFinance API of the ASPSP using the apiContractId. Has to be returned if the attribute validFrom is set within the onboarding resource.

Example

Request

```
GET https://api.testbank.com/openfinance/v2/onboardings/3d9a81b3-a47d-4130-8765-a9c0ff861100
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7731
Date:                   Fri, 29 Dec 2023 15:03:15 GMT
```

3.6 Cancel the onboarding

Call

```
DELETE …/v2/onboardings/{onboardingId}
```

Path Parameters

Attribute	Type	Description
onboardingId	UUID	Onboarding ID as returned with the response message to the initiation of the onboarding process according to section 3.1.

Query Parameters

No Query Parameter

Request Header

No special Request Header parameter

Request Body

No Request Body

Response Code

The HTTP response code equals 200.

Response Header

No special Response Header parameter

Response Body

No Response Body.

Example

Request

```
DELETE https://api.testbank.com/openfinance/v2/onboardings/3d9a81b3-a47d-
4130-8765-a9c0ff861100
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7732
Date:                   Fri, 29 Dec 2023 15:03:15 GMT
```

3.7 Information about a new QSealC of the API Client

Call

```
POST …/v2/onboardings/{onboardingId}/certificates
```

With this POST command the API Client informs the ASPSP that he will use for future calls a new certificate (QSealC) for message signing.

This POST command has to be secured by HTTP message signing using already the new certificate. See section 2.1.

In case of signature failure an http Response Code 401 with a message code SIGNATURE_INVALID will be returned according to error handling defined in [oFA-ProtSec]. See also [oFA DaD] for related message codes.

The validity of the old certificate is not affected by this call. The old certificate stays valid as stated in the old certificate.

Path Parameters

Attribute	Type	Description
onboardingId	UUID	Onboarding ID as returned with the response message to the initiation of the onboarding process according to section 3.1.

Query Parameters

No Query Parameter

Request Header

No special Request Header parameter

Request Body

No Request Body

Response Code

The HTTP response code equals 200.

Response Header

No special Response Header parameter

Response Body

No Response Body

Example

Request

```
POST https://api.testbank.com/openfinance/v2/onboardings/3d9a81b3-a47d-
4130-8765-a9c0ff861100/certificates
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7727
Date:                   Thu, 28 Dec 2023 15:33:05 GMT
Digest:                 of empty body, see section 6.2.2.1 of [oFA-ProtSec]
x-jws-signature:        see section 6.2.2.2 of [oFA-ProtSec]
```

Response

```
HTTP/1.x 200 ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7727
Date:                   Thu, 28 Dec 2023 15:33:35 GMT
Content-Type:           application/json
```

4 References

4.1 Informative References

[XS2A-OR] NextGenPSD2 XS2A Framework, Operational Rules, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, version 1.3., published December 2019

[oFA-ComV2] openFinance API Framework, XS2A API as PSD2 Interface, Implementation Guidelines, Version 2.1, 31 July 2024

4.2 Normative References

[oFA-OR-EPIS] openFinance API Framework, Operational Rules for Extended Services, Extended Payment Initiation Services, Version 2.1, 31 July 2024

[oFA-ProtSec] openFinance API Framework, Implementation Guidelines, Protocol Functions and Security Measures, Version 2.1, 31 July 2024

[oFA DaD] openFinance API Framework, Data Dictionary, Version 2.2, 31 July 2024

[oFA Discov] openFinance API Framework, Implementation Guidelines, Discovery Services, Version 1.0, 15 February 2024

[oFA-OR-ADM] openFinance API Framework, Operational Rules for Extended Services, Administrative Services, Version 0.91, 26 August 2021, draft version after market consultation (not published yet)

[EBA-RTS] Commission Delegated Regulation (EU) 2018/389 of 27 November 2017 supplementing Directive 2015/2366 of the European Parliament and of the Council with regard to Regulatory Technical Standards for Strong Customer Authentication and Common and Secure Open Standards of Communication, C(2017) 7782 final, published 13 March 2018

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

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