THE Berlin GROUP

A EUROPEAN STANDARDS INITIATIVE

openFinance API Framework

Implementation Guidelines for Extended Services

Resource Status Notification Service

Version 1.2

24 September 2021

License Notice

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

Contents

1 Introduction

1.1 From Core XS2A Interface to openFinance API

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

- Payment Initiation Service (PIS) to be operated by a Payment Initiation Service Provider (PISP) TPP as defined by article 66 of [PSD2],
- Account Information Service (AIS) to be operated by an Account Information Service Provider (AISP) TPP as defined by article 67 of [PSD2], and
- Confirmation 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:

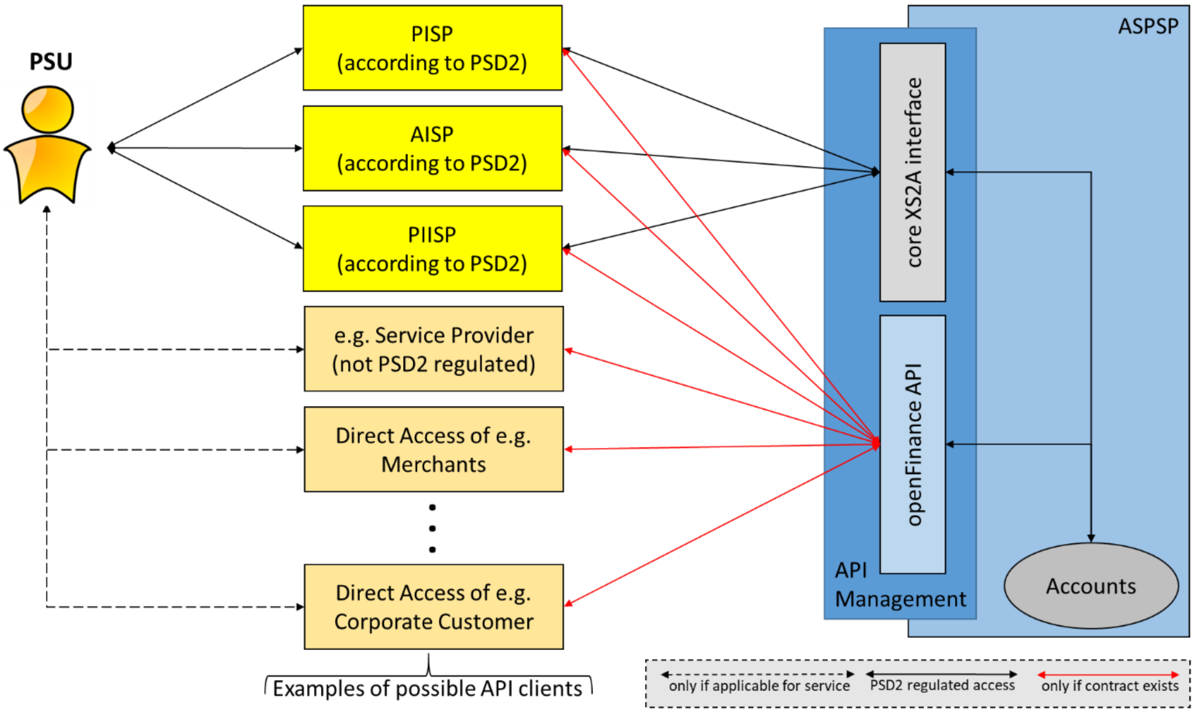

PSU
PISP (according to PSD2)
AISP (according to PSD2)
PIISP (according to PSD2)
e.g. Service Provider (not PSD2 regulated)
Direct Access of e.g. Merchants
Direct Access of e.g. Corporate Customer
Examples of possible API clients
ASPSP
core XS2A interface
openFinance API
API Management
Accounts
only if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

Figure 1: Core XS2A interface and openFinance API

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

1.2 Resource Status Notification Service

This document is an extended service applicable to the core XS2A interface as well as to the openFinance API as displayed above.

The transport layer for the notification services defined in this document are determined by webhooks, where the TPP is registering implicitly an URI when submitting resource data to the ASPSP XS2A interface. This URI can then be used by ASPSPs to post e.g. payment initiation or consent status information or a callback request. Technically this is provided by posting resource based information about resources created through the TPP in the XS2A interface by submitting e.g. payment initiation or consent related data. For this reason, these notification services are called "Resource Status Notification Services" throughout this document.

1.3 Document History

Version	Change/Note	Approved
1.0	Initial public version	2019-03-01
1.1	Final draft for market consultation. The introduction was adapted by the openFinance API Framework context. The service was extended to all resource created within the openFinance API Framework in Section 5.1. A remark was added that a Resource Status Notification Service might fail if the API provided by the TPP webhook is not addressable by the ASPSP just by using its QWAC certificate in Section 5.2. Extended status information attributes were added in the http content of the push transaction defined in Section 6.1.1	2021-05-26
1.2	Introduce the notion of the openFinance API Framework, where the service also applies. Rename the TPP into the more general openFinance term of API Client for that reason, also in attributes. Enhance the documentation by introducing "choice" variants for attribute definitions by following OpenAPI capabilities.	2021-09-24

2 Character Sets and Notations

For definition on character Sets and Notations as well as for request and response notations refer to Chapter 2 of [XS2A-IG].

2.1 Additional Notations

As an extension of the notations in Chapter 3 in [XS2A-IG], the following conditions may be used, where these additional conditions apply to both, requests from the client to the server as well as responses from the server to the client:

Attribute	Type	Condition	Description
		{Or	
		Or	
		Or}	
		{Or – Optional	
		Or – Optional	
		Or – Optional}	

- {Or: The **first** element in a sequence of elements of which **exactly one** has to be included.
- Or: An element in a sequence of elements of which **exactly one** has to be included. The element is **neither the first nor the last** within this sequence.
- Or}: The **last** element in a sequence of elements of which **exactly one** has to be included.
- {Or – Optional: The **first** element in a sequence of elements of which **at most one** may be included.
- Or – Optional: An element in a sequence of elements of which **at most one** may be included. The element is **neither the first nor the last** within this sequence.
- Or – Optional}: The **last** element in a sequence of elements of which **at most one** may be included.

3 Transport Layer

For details on the transport Layer, please refer to Chapter 3 in [XS2A-IG].

> **Note:** The ASPSP is required to use the same web site certificate as client certificate towards the TPP as used as web site certificate in the corresponding TPP – ASPSP XS2A interface as defined in [XS2A-IG] or more generally in the Client – ASPSP API as it is defined in the openFinance API Framework.

4 Application Layer: Guiding Principles

4.1 Additional Error Information

No additional error information is provided for this simple service. Error information is transported by HTTP response codes only, cp. 4.4 for permitted codes.

4.2 TPP Interface API Structure

This specification makes no requirements on the local endpoint structure of the API Client, i.e. the API Client is free to define host, service and transaction identifiers within the Client-Notification-URI implementation. The only restriction is that the domain within the URI equals the domain as contained in the API Client eIDAS web site certificate used for identification towards the ASPSP, cp. Section 5.2. Every notification is done as a POST command towards the address

```
https://<Client-Notification-URI>
```

using additional content parameters {parameters} defined in JSON encoding.

Example1: https://notificationsgateway.tpp-name.eu, where tpp-name.eu is the domain of the TPP.

Example2: https://notificationsgateway.tpp-name.eu/transaction123/notifications, where "transaction123" is a unique transaction reference used internally within the TPP.

The structure of the request/response is described according to the following categories

- Path: Attributes encoded in the Path (not applicable here)
- Query Parameters: Attributes added to the path after the ? sign as process steering flags or filtering attributes for GET access methods. Query parameters of type Boolean shall always be used in a form query-parameter=true or query-parameter=false. Not used for the current specification of the API Client Notification API.
- Header: Attributes encoded in the HTTP header of request or response
- Request: Attributes within the content parameter set of the request
- Response: Attributes within the content parameter set of the response, encoded in JSON

The HTTP response codes which might be used in this XS2A interface are specified in Section **4.4.** This is not repeated for every API call definition.

4.3 API Access Methods

The following table gives an overview on the HTTP access methods supported by the API endpoints.

4.3.1 Notification Endpoint

Endpoint	Method	Condition	Description
<Client-Notification-URI>	POST	Conditional	Notification initiated by ASPSP, endpoint provided by the API Client. This command posts notification content to the provided endpoint. This access method shall be supported by the API Client if a Client-Notification-URI is provided by the API Client in a previous call to the XS2A Interface or openFinance API of the ASPSP.

4.4 HTTP Response Codes

The HTTP response code is communicating the success or failure of an API Client request message. The 4XX HTTP response codes should only be given if the current request cannot be fulfilled, e.g. the syntax of the body content is not correct.

This specification supports the following HTTP response codes for the Client Notification API:

Status Code	Description
200 OK	POST for a notification
400 Bad Request	Validation error occurred. This code will cover malformed syntax in request or incorrect data in payload.
401 Unauthorized	The API Client or the PSU is not correctly authorized to perform the request. Retry the request with correct authentication information.
403 Forbidden	Returned if the resource that was referenced in the path exists but cannot be accessed by the ASPSP. This code should only be used for non-sensitive id references as it will reveal that the resource exists even though it cannot be accessed.
404 Not found	Returned if the endpoint that was referenced in the path does not exist or cannot be referenced by the ASPSP. When in doubt if a specific id in the path is sensitive or not, use the HTTP response code 404 instead of the HTTP response code 403.
405 Method Not Allowed	This code is only sent when the HTTP method (PUT, POST, DELETE, GET etc.) is not supported on a specific endpoint.
408 Request Timeout	The server is still working correctly, but an individual request has timed out.
415 Unsupported Media Type	The ASPSP has supplied a media type which the API Client does not support.
500 Internal Server Error	Internal server error occurred.
503 Service Unavailable	The API Client server is currently unavailable. Generally, this is a temporary state.

5 Implicit Subscription for Resource Status Notification Service

The NextGenPSD2 XS2A Interface supports several requests for creating resources, e.g. the Payment Initiation Request, the Establish Account Information Request or the Signing Basket Request. For all the related POST commands as defined in [XS2A-IG] or future developments of the specification, this section describes how a TPP or more generally an API Client can implicitly subscribe for a Resource Status Notification Service by extending these commands.

NOTE: The notification services will also be available for cancellation processes which require SCA based authentication of PSUs. These services will then be supported by the ASPSP if requested before by the API Client for the related resource initiation process. So, the "notification service" support function is stored within the created resource.

NOTE: The Resource Status Notification Service is an extended service of the openFinance API Framework. This specification makes no assumption whether a contract may be needed for the ASPSP to offer this service to API Clients.

5.1 Communicate Notification URI of API Clients

Call

```
Any POST command
  • creating a payment, signing basket or consent resource in the ASPSP
    server as defined in [XS2A-IG] or
  • creating a resource within the extended services of the openFinance
    API Framework.
```

Creates a corresponding resource in the ASPSP server.

Path Parameters

No specific requirements.

Query Parameters

No specific requirements

Request Header

The following table contains only the request headers which have to be supported by the TPP or the API Client more generally in addition to headers defined for the corresponding resource creation request.

Attribute	Type	Condition	Description
Client-Notification-URI	String	Optional	URI for the Endpoint of the Client API to which the status of the resource should be sent. This header field may by ignored by the ASPSP.
Client-Notification-Content-Preferred	String	Optional	The string has the form status=X1, …, Xn where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated.

Attribute	Type	Condition	Description
			The usage of the constants supports the following semantics: SCA: A notification on every change of the scaStatus attribute for all related authorisation processes is preferred by the API Client. PROCESS: A notification on all changes of resource status attributes is preferred by the API Client. LAST: Only a notification on the last resource status as available in the XS2A interface or openFinance API is preferred by the API Client. This header field may be ignored, if the ASPSP does not support resource notification services for the related API Client.

Request Body

No specific requirements.

Response Code

No specific requirements

Response Header

The following table contains only the response headers which have to be supported by the ASPSP in addition to headers defined for the corresponding resource creation response if the Resource Status Notification Service is supported.

Attribute	Type	Condition	Description
ASPSP-Notification-Support	Boolean	Conditional	true if the ASPSP supports resource status notification services. false if the ASPSP supports resource status notification in general, but not for the current request. Not used, if resource status notification services are generally not supported by the ASPSP. Shall be supported if the ASPSP supports resource status notification services.

Attribute	Type	Condition	Description
ASPSP-Notification-Content	String	Conditional	The string has the form status=X1, …, Xn where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated. The usage of the constants supports the following semantics: SCA: Notification on every change of the scaStatus attribute for all related authorisation processes is provided by the ASPSP for the related resource. PROCESS: Notification on all changes of consentStatus or transactionStatus attributes is provided by the ASPSP for the related resource. LAST: Notification on the last consentStatus or transactionStatus as available in the XS2A interface is provided by the ASPSP for the related resource. This field must be provided if the ASPSP-Notification-Support =true. The ASPSP might consider the notification content as preferred by the TPP, but can also respond independently of the preferred request.

Response Body

No specific requirements.

5.2 Requirements on the Notification URI

For security reasons, it shall be ensured that the Client-Notification-URI as introduced above is secured by the API Client eIDAS QWAC used for identification of the API Client. The following applies:

URIs which are provided by API Clients in Client-Notification-URI shall comply with the domain secured by the eIDAS QWAC certificate of the API Client in the field CN or SubjectAltName of the certificate. Please note that in case of example-TPP.com as certificate entry TPP-Notification-URI like

- www.example-TPP.com/xs2a-client/v1/ASPSPidentifcation/mytransaction-id/notifications or

- notifications.example-TPP.com/xs2a-client/v1/ASPSPidentifcation/mytransaction-id/notifications

would be compliant.

Wildcard definitions shall be taken into account for compliance checks by the ASPSP.

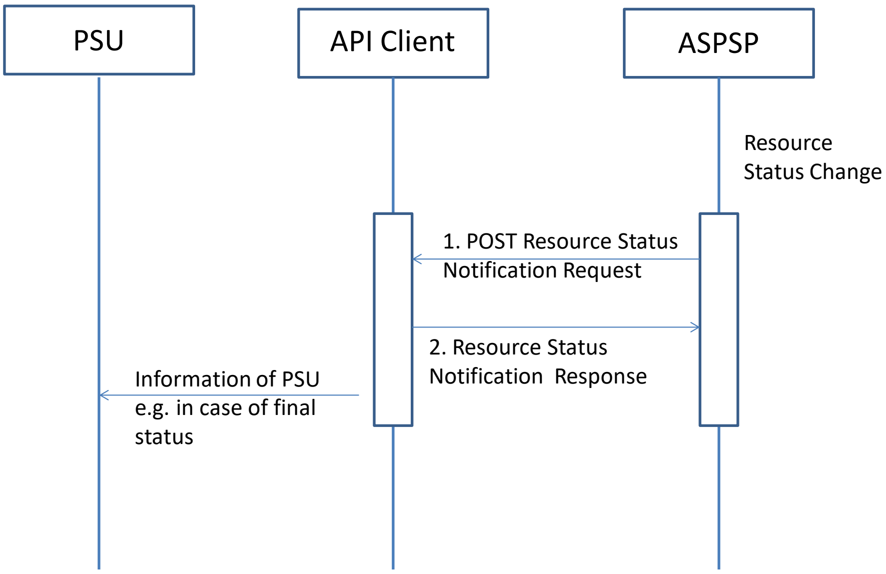
ASPSPs may respond with ASPSP-Notification-Support set to `false`, if the provided URIs do not comply.

NOTE: The URI should be addressable by the ASPSP just by presenting the related QWAC. No further pre-steps of any kind should be mandated by the API Client. If so, the Resource Status Notification Service could fail.

6 Resource Notification Push Service

6.1 Resource Notification Push Message Flow

The following flow shows the simple request and response flow for a resource status notification service:



Remark: In case, where the ASPSP is only pushing a status hyperlink to the API Client, the API Client needs to check the resource status after step 2.) before informing e.g. the PSU.

6.1.1 Push Resource Status with JSON encoding

Call

```
POST <Client-Notification-URL>
```

Creates a Resource Notification on the API Client server.

Path Parameters

No Path Parameter

Query Parameters

No Query Parameter

Request Header

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

Request Body

Attribute	Type	Condition	Description
paymentId	String	{Or	This shall be contained, if the push notification is about a payment or RTP initiation.
consentId	String	Or	This shall be contained if the push notification is about establishing a consent.
subscriptionId	String	Or	This shall be contained if the push notification is about establishing a subscription.
basketId	String	Or}	This shall be contained if the push notification is about signing a basket.
entryId	String	{Or - Optional	This may be used if the status relates to an entry of an RTP bulk.
subscriptionEntryId	String	Or - Optional}	This may be used if the status relates to an entry of a subscription.
authorisationId	String	{Or - Optional	This attribute should be contained if the push notification is about a specific SCA status.
cancellationId	String	Or - Optional}	This attribute should be contained if the push notification is about a specific SCA status of a cancellation authorisation sub-resource.

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	{Or Optional	This attribute might be contained if the related resource contains a transaction status which has changed.
consentStatus	Consent Status	Or Optional	This attribute might be contained if the consent status of the addressed resource has changed.
subscriptionStatus	Subscripton Status	Or Optional	This attribute might be contained if the subscription status of the addressed resource has changed.
subscriptionEntryStatus	Subscription Entry Status	Or Optional}	This attribute might be contained if the subscription entry status of the addressed resource has changed.
scaStatus	SCA Status	Optional	This attribute might be contained if the authorisation status of the addressed authorisation resource has changed.
requestStatus	Request Status	Optional	The status of the related request to pay transaction. To be delivered by the API Server if not agreed otherwise.
reasonCode	Status Reason Code	{Or Optional	Additional information on the reason for e.g. rejecting the request
reasonProprietary	Max35Text	Or Optional}	Proprietary additional information on the reason for e.g. rejecting the request.
debtorDecision DateTime	ISO Date Time	Optional	The date and time when the PSU has decided on accepting/rejecting the related request.
acceptedAmount	Amount	Optional	Contained only if the accepted amount deviates from the instructed amount.
acceptanceDateTime	ISODateTime	Optional	Contained only if the agreed requested execution date deviates from the requested execution date in the request.
acceptedPaymentInstrument	Max105Text	Optional	"SCT" or "SCT inst" as default values.
statusIdentification	Max35Text	Optional	Reference added by the debtor.
_links	Links	Optional	The following link types are supported.

Attribute	Type	Condition	Description
			scaStatus This shall be contained if the related SCA status is not reported at the same time by the scaStatus attribute. The API Client then needs to get the scaStatus by a GET command using this hyperlink.
			status This shall be contained if the related consent or transaction status is not reported at the same time. The API Client then needs to get the resource status by a GET command using this hyperlink.

HTTP Response Code

200

Remark: All response codes which do not equal 200 are ignored by the ASPSP. The notification will not be repeated.

Response Header

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

Response Body

No Response Body

Example

Request

```
POST https://notifications.testclient.com/v1/transaction-12345
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:02:37 GMT

{
  "payment-ID": "12345-23454-123123",
  "transactionStatus": "ACFC"
}
```

Response

```
HTTP/1.x 200
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
```

7 References

[XS2A-OR] NextGenPSD2 XS2A Framework, Operational Rules, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, version 1.3, published 21 December 2018

[XS2A-IG] NextGenPSD2 XS2A Interoperability Framework, Implementation Guidelines, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, version 1.3.11, published 24 September 2021

[oFA-OR-Adm] openFinance API Framework, Operational Rules, Administrative Services, version 0.9, Consultation Draft, 7 June 2021

[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

[signHTTP] Signing HTTP messages, Network Working Group, Internet Draft version 10, https://datatracker.ietf.org/doc/draft-cavage-http-signatures/

[HAL] Kelley, M., "HAL - Hypertext Application Language", 2013-09-18, http://stateless.co/hal_specification.html

[RFC2426] Dawson, F. and T. Howes, T., "vCard MIME Directory Profile", September 1998, https://tools.ietf.org/html/rfc2426

[RFC3230] Mogul, J. and A. Van Hoff, "Instance Digests in HTTP", RFC 3230, DOI 10.17487/RFC3230, January 2002,https://www.rfc-editor.org/info/rfc3230

[RFC4648] Josefsson, S.," The Base16, Base32, and Base64 Data Encodings", October 2006, https://tools.ietf.org/html/rfc4648

[RFC5843] Bryan, A, "Additional Hash Algorithms for HTTP Instance Digests", RFC 5843, DOI 10.17487/RFC5843, April 2010, https://www.rfc-editor.org/info/rfc5843

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", October 2012, https://tools.ietf.org/html/rfc6749

[RFC7807] M. Nottingham, Akamai, E. Wilde, „Problem Details for HTTP APIs“, March 2016, https://tools.ietf.org/html/rfc7807