THE Berlin GROUP

A EUROPEAN STANDARDS INITIATIVE

NextGenPSD2 XS2A Framework

Implementation Guidelines

Extended Services

Resource Status Notification Service

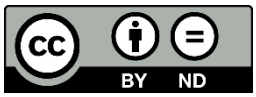
Version 1.0

01 March 2019

License Notice

This Specification has been prepared by the Participants of the Joint Initiative pan-European PSD2-Interface Interoperability* (hereafter: Joint Initiative). 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.
- 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 'Joint Initiative pan-European PSD2-Interface Interoperability' brings together participants of the Berlin Group with additional European banks (ASPSPs), banking associations, payment associations, payment schemes and interbank processors.

Contents

1 Introduction....2
1.1 Background....2
1.2 XS2A Interface Specification....3
2 Character Sets and Notations....4
3 Transport Layer....4
4 Application Layer: Guiding Principles....4
4.1 Additional Error Information....4
4.2 TPP Interface API Structure....4
4.3 API Access Methods....5
4.3.1 Notification Endpoint....5
4.4 HTTP Response Codes....6
5 Implicit Subscription for Resource Status Notification Service....6
5.1 Communicate Notification URI of TPPs....7
5.2 Requirements on the Notification URI....9
6 Resource Notification Push Service....10
6.1 Resource Notification Push Message Flow....10
6.1.1 Push Payment Status with JSON encoding....10
7 References....14

1 Introduction

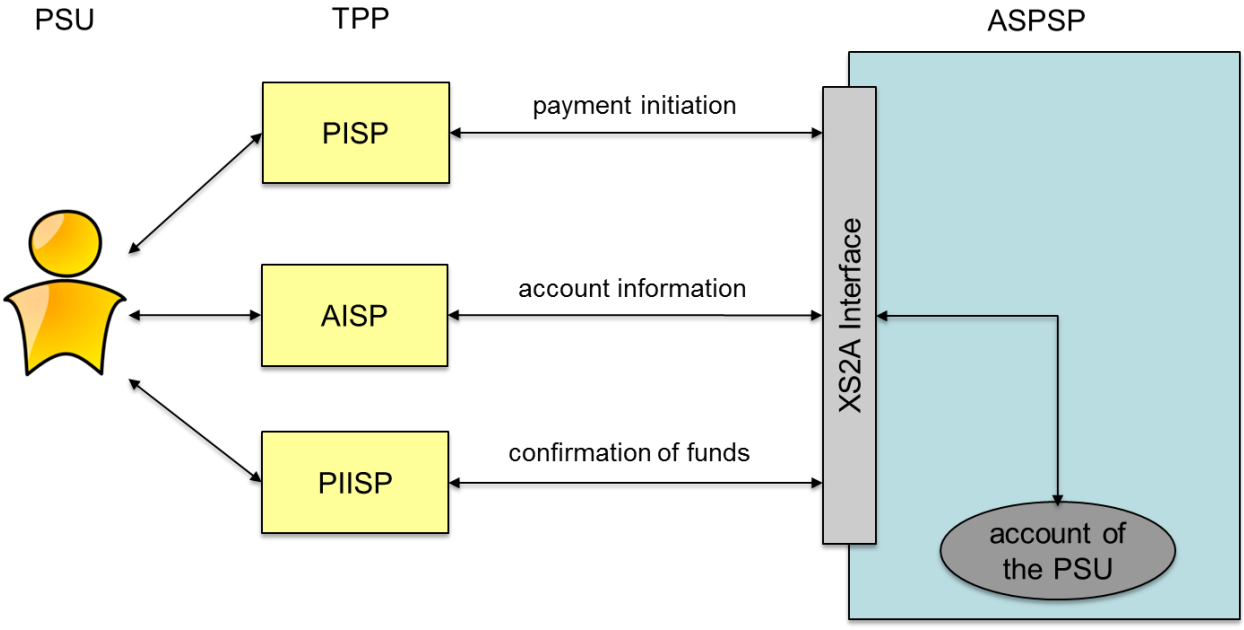
1.1 Background

With [PSD2] the European Union has published a new directive on payment services in the internal market. Member States had to adopt this directive into their national law until 13th of January 2018.

Among others [PSD2] contains regulations of new services to be operated by so called Third Party Payment Service Providers (TPP) on behalf of a Payment Service User (PSU). These new services are

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

For operating the new services a TPP needs to access the account of the PSU which is usually managed by another PSP called the Account Servicing Payment Service Provider (ASPSP). As shown in the following figure, an ASPSP has to provide an interface (called "PSD2 compliant Access to Account Interface" or short "XS2A Interface") to its systems to be used by a TPP for necessary accesses regulated by [PSD2]:



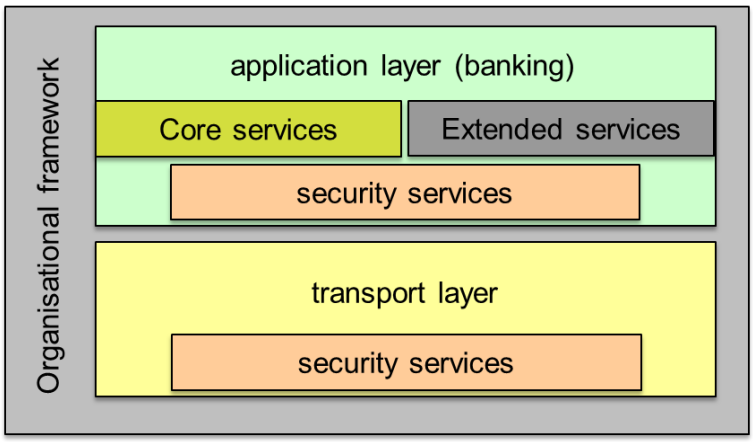
1.2 XS2A Interface Specification

This document is an extension of the NextGenPSD2 XS2A Specification which defines a standard for an XS2A Interface and by this reaching interoperability of the interfaces of ASPSPs at least for the core services defined by [PSD2].

The core XS2A Interface is designed as a B2B interface between a TPP server and the ASPSP server. The core NextGenPSD2 XS2A Specification as defined in [XS2A-IG] is a pure client-server protocol, assuming the TPP server being the client, i.e. all API calls are initiated by the TPP. The Interoperability Framework defines operational rules, requirements on the data model and a process description in [XS2A-OR].

This document defines Notification Services for the NextGenPSD2 XS2A Interface, where the ASPSP server is the client and the TPP server is the server. These push services are restricted to the case where the TPP has created resources in the ASPSPs server e.g. for payment initiations or establishing account information consent. The push services will be offered to inform the TPP e.g. about status changes of the resources either by directly communicating the new status or request a callback for a dedicated resource.

This document details the standard in defining messages and detailed data structures for these extended services of the XS2A Interface. For the specification the two layers shown in the following figure are distinguished:



.

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.

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].

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].

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 TPP, i.e. the TPP is free to define host, service and transaction identifiers within the TPP-Notification-URI implementation. The only restriction is that the domain within the URI equals the domain as contained in the TPP 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://<TPP-Notification-URI>
```

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

Example: https://notificationsgateway.tpp-name.eu, where tpp-name.eu is the domain of 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 TPP 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
<TPP-Notification-URI>	POST	Conditional	Notification initiated by ASPSP, endpoint provided by the TPP. This command posts notification content to the provided endpoint. This access method shall be supported by the TPP if a TPP-Notification-URI is provided by the TPP in a previous call to the XS2A Interface of the ASPSP.

4.4 HTTP Response Codes

The HTTP response code is communicating the success or failure of a TPP 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 TPP 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 TPP 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 TPP does not support.
500 Internal Server Error	Internal server error occurred.
503 Service Unavailable	The TPP 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 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 TPP 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 NextGenPSD2 XS2A interface framework. This specification makes no assumption whether a contract may be needed for the ASPSP to offer this service to TPPs.

5.1 Communicate Notification URI of TPPs

Call

```
Any POST command creating a payment, signing basket or consent resource in
the ASPSP server as defined in [XS2A-IG].
```

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 in addition to headers defined for the corresponding resource creation request.

Attribute	Type	Condition	Description
TPP-Notification-URI	String	Optional	URI for the Endpoint of the TPP-API to which the status of the payment initiation should be sent. This header field may by ignored by the ASPSP.
TPP-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. The usage of the constants supports the following semantics:

Attribute	Type	Condition	Description
			SCA: A notification on every change of the scaStatus attribute for all related authorisation processes is preferred by the TPP. PROCESS: A notification on all changes of consentStatus or transactionStatus attributes is preferred by the TPP. LAST: Only a notification on the last consentStatus or transactionStatus as available in the XS2A interface is preferred by the TPP. This header field may be ignored, if the ASPSP does not support resource notification services for the related TPP.

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.
ASPSP-Notification-	String	Conditional	The string has the form

Attribute	Type	Condition	Description
Content			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 TPP-Notification-URI as introduced above is secured by the TPP eIDAS QWAC used for identification of the TPP. The following applies:

URIs which are provided by TPPs in TPP-Notification-URI shall comply with the domain secured by the eIDAS QWAC certificate of the TPP 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.

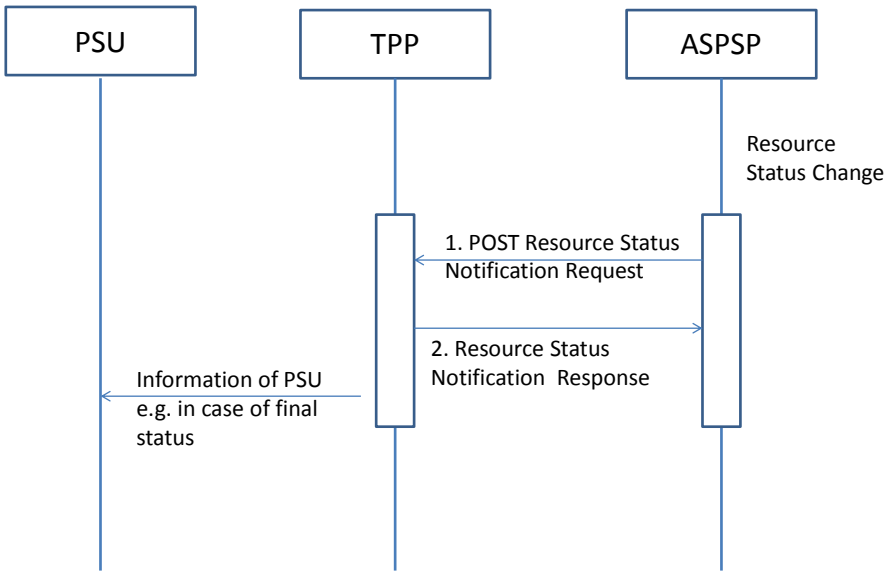
Wildcard definitions shall be taken into account for compliance checks by the ASPSP.

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

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 TPP, the TPP needs to check the resource status after step 2.) before informing e.g. the PSU.

6.1.1 Push Payment Status with JSON encoding

Call

```
POST <TPP-Notification-URL>
```

Creates a Resource Notification on the TPP 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	Conditional	This shall be contained, if the push notification is about a payment initiation.
consentId	String	Conditional	This shall be contained if the push notification is about establishing a consent.
basketId	String	Conditional	This shall be contained if the push notification is about signing a basket.
authorisationId	String	Optional	This attribute should be contained if the push notification is about a specific SCA status.
cancellationId	String	Optional	This attribute should be contained if the push notification is about a specific SCA status of a cancellation authorisation sub-resource.
transactionStatus	Transaction Status	Optional	This attribute might be contained if the related resource contains a transaction status which has changed.
consentStatus	Consent Status	Optional	This attribute might be contained if the consent 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.
_links	Links	Optional	The following link types are supported.
			scaStatus This shall be contained if the related SCA status is not reported at the same time by the scaStatus attribute. The TPP then needs to get

Attribute	Type	Condition	Description
			the scaStatus by a GET command using this hyperlink.
			status This shall be contained if the related consent or transactio status is not reported at the same time. The TPP 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 1.0, published 08 February 2018

[XS2A-IG] NextGenPSD2 XS2A Interoperability Framework, Implementation Guidelines, The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface, version 1.3, published 15 October 2018

[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