

TPP Developer Documentation

- Introduction
- Definitions
- API Overview
 - OAuth2 API
 - Services API
- Authentication and Security
- Errors
- OAuth2 Access Token
 - Example
 - OAuth2 access code
 - Access token
- AIS services
 - Example
 - Create a consent
 - Create an authorization resource for a consent
 - Choose SCA Redirect
 - Consent authorization status
 - Consent status
 - Accounts information
 - Account balances
 - Account transactions
- PIS services
 - Example
 - Create a payment
 - Payment authorization resource
 - Choose Redirect SCA method
 - Payment authorization status
 - Payment status
- PIISP services
 - Example

Introduction

The purpose of this document is to help and guide the developers around on what is possible to access in terms of the data regarding customers. It covers general principles, workflows as well as API functional and technical details. All the details on each endpoint can be found in the [Developer portal](#).

The document is written based on the [Berlin Group guidelines](#).

Definitions

	Description
PSD2 XS2A (or XS2A)	"Access to account" services as defined under Berlin Group guidelines
API	An application program interface (API) is a set of routines, protocols, and tools for building software applications.
PSU (Payment Service User)	The user here refers to a bank customer who uses the TPP application.
ASPSP	The account servicing provider.
QWAC	eIDAS Qualified Website Authentication Certificate
AISP (Account Information Service provider)	TPP providing AIS services
AIS	Account information service
PISP (Payment Initiation Service Provider)	TPP providing PIS services
PIS	Payment initiation service

PIISP (Payment Instrument Issuing Service Provider)	TPP providing PIIS services
PIIS	Payment instrument issuing service
TPP (Third Party Provider)	The Licensed Third Party Provider (TPP) is a provider of an application being used by the user and not offered by the bank. TPP is a client/consumer of the API and acts on behalf of the user under consent.
SCA	Strong customer authentication.

API Overview

The API consists of two parts: OAuth2 API and PSD2 Services API. It is implemented as REST API via HTTPS protocol with payload messages as JSON.

OAuth2 API

OAuth2 API provides means to acquire an OAuth2 access token which is necessary for invoking XS2A services.

Services API

Services API provides AIS, PIS and PIIS services.

Authentication and Security

The entire communication between TPP and API is secured by TLS version 1.2 or higher. TPP must have a valid QWAC certificate in order to pass a TLS client authentication during a TLS handshake, otherwise a connection will not be established and API services will not be invoked.

Services API requires an OAuth2 access token which can be acquired from OAuth2 API; this process demands PSU authorization.

AIS services API requires TPP to obtain PSU consent in order to access the account information; this process demands PSU authorization.

Errors

Errors are implemented as HTTP responses with a status code and sometimes a JSON body as defined by the Berlin Group guidelines.

JSON body structure is as follows:

HTTP Response

```

{
  "tppMessages" : [ {
    "category" : "ERROR",
    "code" : "TOKEN_INVALID",
    "text" : "additional text information of the ASPSP up to 512 characters"
  } ]
}

```

OAuth2 Access Token

Before any of the PIS, AIS, PIIS services can be invoked, an OAuth2 access token must be acquired. Acquiring an access token involves PSU who will have to authorize it. An access token is valid for 90 days, after that it must be renewed and it must be again authorized by PSU.

An access token is authorized by redirecting PSU to the ASPSP interface. When access token is authorized, then PSU is redirected back to TPP with a code value as a parameter. This code is then exchanged for an actual access token.

Follow the example below to understand the steps needed to acquire the access token.

Example

OAuth2 access code

A redirect URL must be acquired so that PSU could be redirected to the ASPSP interface to authorize an access token. The following endpoint must be invoked:

GET https://api.xs2a/v1/oauth/authorization/links			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
HTTP query parameters			
Name	Type	Condition	Description
branch	String	Mandatory	ASPSP branch. (branch=PARKO)

Let's acquire an authorization URL.

HTTP Request

```
GET https://api.xs2a/v1/oauth/authorization/links?branch=PARKO
X-Request-ID:e77bb703-4b3d-435f-91a9-125113ecfaf5
```

HTTP Response

```
Status: 200
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2019 07:33:55 GMT
Content-Type: application/json

[{"url" :
"http://auth.xs2a/auth/realms/demo/protocol/openid-connect/auth?response_type=code&client_id=xs2a-restapi-client&login=true&scope=openid"}]
```

An HTTP response returns a JSON body which contains a URL where PSU must be redirected. Append "redirect_uri" and "state" query parameters to this URL. redirect_uri – location where PSU should be redirected when authorization is finished. "state" is a random value that helps to protect against CSRF attacks. The final URL could look like this:

```
http://auth.xs2a/auth/realms/demo/protocol/openid-connect/auth?response_type=code&client_id=xs2a-restapi-client&login=true&scope=openid&redirect_uri=http://tpp.com/users/1&state=state123
```

Now redirect PSU to this URL.

Access token

When PSU finishes the access token authorization in the ASPSP interface, it is redirected to "redirect_uri", in our example case it would be:

`http://tpp.com/users/1?state=state123&code=988affc6-f96a-4463-ab08-ea074e9fcb7c.2cbbaa24-32ed-4c84-89c5-077c02c203e0.3b7be450-bb75-4115-8d6a-d13d61785b06`

The "code" query parameter is appended to the final URL. The "code" parameter is used to acquire the access token. The following endpoint must be invoked:

POST <code>https://api.xs2a/v1/oauth/token</code>			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
HTTP query parameters			
Name	Type	Condition	Description
branch	String	Mandatory	ASPSP branch. (branch=PARKO)
HTTP body: JSON			
\$.grant_type	String	Mandatory	value: authorization_code
\$.code	String	Mandatory	The "code" parameter received when PSU was redirected back to TPP
\$.redirect_uri	String	Mandatory	The same "redirect_uri" parameter that was used when redirecting user to the ASPSP interface

Exchange code for an access token:

```

HTTP Request
POST https://api.xs2a/v1/oauth/token?branch=PARKO
Content-Type:application/json
X-Request-ID:859beb82-e608-43fb-aa35-8e348d3b829e

{
  "grant_type": "authorization_code",
  "code":
  "988affc6-f96a-4463-ab08-ea074e9fcb7c.2cbbaa24-32ed-4c84-89c5-077c02c203e0
  .3b7be450-bb75-4115-8d6a-d13d61785b06",
  "redirect_uri": "http://tpp.com/users/1"
}

```

```

HTTP Response
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 07:34:10 GMT
Content-Type:application/json

```

```
{
  "access_token":
  "eyJhbGciOiJSUzI1NiIsInR5cCI6IgorIiwia2lkIiA6ICJOb3BYTy1zdZzZTek9ObEg0Y1VXR3liODFEXy16U2huSUViVUNBT0FydHhFIn0.eyJqdGkiOiI3YmNhZjAxMy0yMzU4LTQlZDc0tYWZkYS0xNTg0MzE5YWQ1YWIiLCJleHAiOiE1NjgwMTQ3NDksIm5iZiI6MCwiaWF0IjoxNTY4MDE0NDQ5LkZpc3MiOiJodHRwOi8vMTcyLjE2LjEuMTYxOjQ0NDQvYXV0aC9yZWZsbnMvZGVtbyIsInN1YiI6ImU3MTYzZjdmLTYYnZctNGYzOC05YTBhLWY3MjRjYjRmNWEyZSIsInR5cCI6IkkjYXJlciIsImF6cCI6InhzMmEtcMvZdGFwaS1jbGllbnQiLCJub25jZSI6Ijg5YWY1YTRmLTcyODAtNDZjZi04MWFmLWZmMzMWIxYzNiMjliZiIsImFldGhfdGltZSI6MTU2ODAxNDQ0O0Swic2Vzc2l9ZdGF0ZSI6IjJjYmJhYTI0LTMyZWQtNGM4NC04OwM1LTA3N2MwMmMyMDN1MCI6ImFjciI6IjEiLCJzY29wZSI6Im9wZW5pZCIsInByZWZlcnJlZDk5Y2VybWVtZSI6ImR6aXVnYWlsYSJ9.ZyORhmQCRp9J2LtJMzvIqQdff-q4scCfG8_BGZBkaBUaOs3izQlWWjrtJt9SiW-glJQBrjvqx2edeLIL2tsZ_Npfsf8Eg-HQ7GY9aB4yu8x0PdU0WLKEEqbwY39KUaS_2llwaDT0FlFLEFru-ECJR-aoHMP3ZOocX-M5ZwxkxwEh_OmDX1Y7krZlg5Hr-2j6YGGokOvTKL6mcw8eLZ1ne16zlxI9HapPoXAlH1Xvq7zG2bgUfTxNrYnebXaeItPzEbQXd4WHEiZtq6FhWxE5qkWmsYCjziQUU7NJWpvQwnDQfdWbhc7rAx7Sm16jgIZXnlm5SXX_410TVYwWt_5iA",
  "expires_in": 300,
  "refresh_expires_in": 1799,
  "refresh_token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IgorIiwia2lkIiA6ICJlZGVizjk3YS02NzNiLTQyNGU0ThmYi0zOWIxZjcwYzhlOTUifQ.eyJqdGkiOiI1N2QzYWZlZS0zNzc2LTRhYXQ0OGVhNy1iOWQ5NzIxMjhhYmMiLCJleHAiOiE1NjgwMTYyNDksIm5iZiI6MCwiaWF0IjoxNTY4MDE0NDQ5LkZpc3MiOiJodHRwOi8vMTcyLjE2LjEuMTYxOjQ0NDQvYXV0aC9yZWZsbnMvZGVtbyIsImF1ZCI6Imh0dHA6Ly8xNzIuMTYyMS4xNjE6NDQ0NC9hdXRoL3JlYWxtcy9kZWlvIiwic3ViIjoizTcxNjNmN2YtNjI3Ny00ZjM4LTIhMGEtZjcyNGNiNGY1YjY1IiwidHlwIjoiaU0vMmVzaCI6ImF6cCI6InhzMmEtcMvZdGFwaS1jbGllbnQiLCJub25jZSI6Ijg5YWY1YTRmLTcyODAtNDZjZi04MWFmLWZmMzMWIxYzNiMjliZiIsImFldGhfdGltZSI6MCwic2Vzc2l9ZdGF0ZSI6IjJjYmJhYTI0LTMyZWQtNGM4NC04OwM1LTA3N2MwMmMyMDN1MCI6InNjb3BlIjoib3BlbmkiIn0.v-8w7GthNojs68H6mT7a50xzbEze9LEnJyMW4p65qJk",
  "token_type": "bearer",
  "id_token":
  "eyJhbGciOiJSUzI1NiIsInR5cCI6IgorIiwia2lkIiA6ICJOb3BYTy1zdZzZTek9ObEg0Y1VXR3liODFEXy16U2huSUViVUNBT0FydHhFIn0.eyJqdGkiOiIwNzk5MzZkZi0yMGRiLTRhYXQ0tYTY3MCIjNTEwMzYwZjc4MTQlLCJleHAiOiE1NjgwMTQ3NDksIm5iZiI6MCwiaWF0IjoxNTY4MDE0NDQ5LkZpc3MiOiJodHRwOi8vMTcyLjE2LjEuMTYxOjQ0NDQvYXV0aC9yZWZsbnMvZGVtbyIsImF1ZCI6InhzMmEtcMvZdGFwaS1jbGllbnQiLCJub25jZSI6Ijg5YWY1YTRmLTcyODAtNDZjZi04MWFmLWZmMzMWIxYzNiMjliZiIsImFldGhfdGltZSI6MTU2ODAxNDQ0O0Swic2Vzc2l9ZdGF0ZSI6IjJjYmJhYTI0LTMyZWQtNGM4NC04OwM1LTA3N2MwMmMyMDN1MCI6ImFjciI6IjEiLCJwcmVmcjYzZWZlZDk5YzVhbnU0IjkiLCJzY29wZSI6Im9wZW5pZCIsInByZWZlcnJlZDk5Y2VybWVtZSI6ImR6aXVnYWlsYSJ9.dPBm0OabgjqfZAVJ-bcwcxRsr6mCuFAyKn5HMYtt2_pBhPQJNU0oUIeuxJBm1ZIDAYEtFB0heBoYfimuI4Ck806d-lyV5Yy8IzDglDX7nn9HKJJ-U8DhkGkP0IRh5g88eT-J9gZnjxuezp hJ8ES418JEp0L-Td9Ww_G78Mqll938NaARRA69qcSvz6komb-EbVp3zNrEVnNhkLyRBAOgSEwX2F5ahsgYiNXH6hfmoxq9hrckW08VQh2j02YWjCKE3JpaMcm0fmTr8-LquqHRtx-hjXTOTIhWzJAxqvAYiOgP6he01-q71s06fg0-tTuFYUvatLSwks4paDTUJ3zkHSw",
  "not-before-policy": 0,
```

```
    "session_state": "2cbbaa24-32ed-4c84-89c5-077c02c203e0",
    "scope": "openid"
}
```

An HTTP response returns a JSON body. Extract the access_token from the JSON and use it when calling any of the PIS, AIS, PIIS services. The access token will have to be provided as an HTTP header, the access token value should be preceded by "Bearer ":

Authorization:Bearer

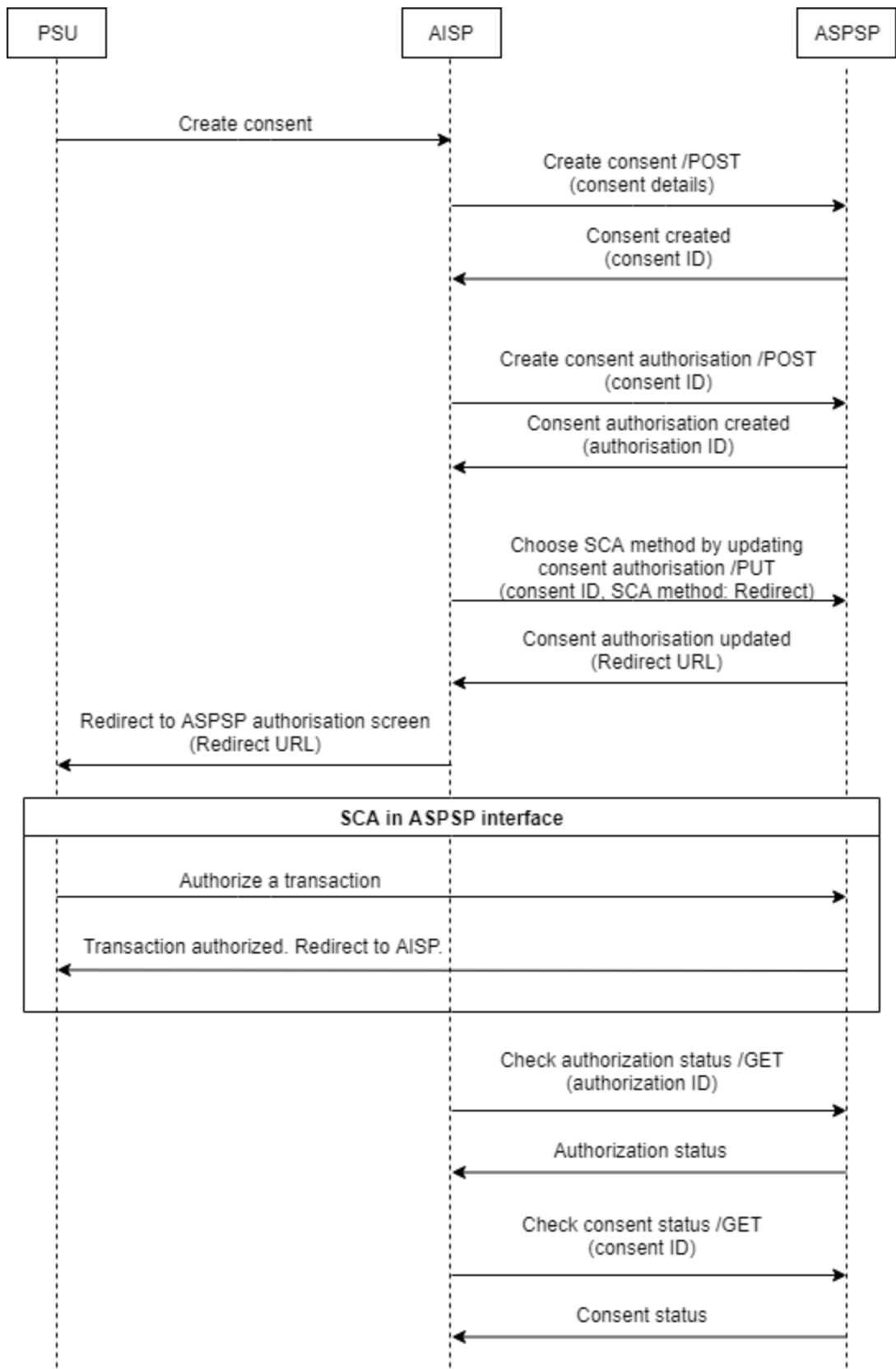
```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXLTZkdzZTek9ObEg0Y1VXR3liODFEXy16U2huSUViVUNBT0FydHh  
FIn0.eyJqdGkiOiI3YmNhZjAxMy0yMzU4LTQ1ZDctYWZkYS0xNTg0MzE5YWQ1YWYiLCJleHAiOiJlNjgwMTQ3NDksIm5iZiI6MCwiaWF0Ij  
oxNTY4MDE0NDQ5LlCjpc3MiOiJodHRwOi8vMTcyLjE2LjEuMTYxOjQ0NDQvYXV0aC9yZWZsbXVzZGVtbyIsInN1YiI6ImU3MTYzZjdmLTYyN  
zctNGYzOC05YTBhLWY3MjRjYjRmNWYyZSIsInR5cCI6IkpXLTZkdzZTek9ObEg0Y1VXR3liODFEXy16U2huSUViVUNBT0FydHhFIn0.  
YWY1YTRmLTcyODAtNDZjZi04MWFmLWwzMWIxYzNiMjliZiIsImF1dG8iOiJlNjgwMTQ3NDksIm5iZiI6MCwiaWF0IjoxNTY4MDE0NDQ5Ll  
hYTI0LTMyZWQtNGM4NC04OWM1LTA3N2MwMmMyMDNlMCI6ImF1dG8iOiJlNjgwMTQ3NDksIm5iZiI6MCwiaWF0IjoxNTY4MDE0NDQ5LlCjpc3  
I6ImR6aXVnYWlsYSJ9.Zy0RhmQCRp9J2LtJMzvIgQdff-q4scCf8_BGZBkaBUaOs3izQlWWjrtJt9SiW-glJQBrjvxq2edeLIL2tsZ_Npf  
sf8Eg-HQ7GY9aB4yu8x0PgU0WLKEEqbwy39KUaS_21lwaDT0f1FLFru-ECjR-aoHMP3ZOocX-M5Zwxkwxeh_OmDX1Y7krZlg5Hr-2j6YG  
GokOvTKL6mcw8eLz1ne16z1Xi9HapPoXAlH1Xvq7zG2bgUfTxNrYnebXaeItPzEbQXd4WHEiZtq6FhWxE5qkWmsYCjziQUU7NJWPvQwnDQf  
dWbhc7rAx7Sm16jgIZXnlm5SXX_410TVYWwT_5iA
```

For brevity, examples using the access token will not display its actual value, but will use `_ACCESS_TOKEN` in its place.

AIS services

AIS services allow TPP to access PSU account information, account balances and account transactions (statements). Before any of the services can be used, TPP must obtain a consent from PSU. The consent contains information to what accounts PSU has granted access to TPP and how often account information can be accessed. The consent must be authorized by PSU with SCA.

There may only be one authorized consent at any time for a particular PSU. If additional access rights are to be granted by PSU, then a new consent must be created and authorized – the previous consent will be invalidated.



Follow the example below to understand the steps needed to acquire the consent.

Example

Create a consent

A consent must be created with the account access details. The following endpoint must be invoked:

POST https://api.xs2a/v1/consents			
https://psd2-sandbox.fininbox.com/#/Consents/createConsent			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"
HTTP body defined in https://psd2-sandbox.fininbox.com/#/Consents/createConsent			

Let's create a consent that allows TPP to access PSU account LT044010000100439350 information, balances, transactions. Information can be accessed 100 times per day until 2019-10-04.

HTTP Request

```
POST https://api.xs2a/v1/consents
Authorization:Bearer _ACCESS_TOKEN
Content-Type:application/json
X-Request-ID:f63daf5f-27ae-4992-9eda-940dd2a1dae0

{
  "access": {
    "accounts": [
      {
        "iban": "LT044010000100439350",
        "currency": "EUR"
      }
    ],
    "balances": [
      {
        "iban": "LT044010000100439350",
        "currency": "EUR"
      }
    ],
    "transactions": [
      {
        "iban": "LT044010000100439350",
        "currency": "EUR"
      }
    ]
  },
  "recurringIndicator": true,
  "validUntil": "2019-10-04",
  "frequencyPerDay": 100
}
```


HTTP Response

```
Status: 201
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2019 07:35:10 GMT
Content-Type: application/json

{
  "consentId": "OLS4A06EQGX3P47ODJG2L2DNICR8JS0000016612",
  "_links": {
    "self": {
      "href": "/v1/consents/OLS4A06EQGX3P47ODJG2L2DNICR8JS0000016612"
    },
    "startAuthorisation": {
      "href":
"/v1/consents/OLS4A06EQGX3P47ODJG2L2DNICR8JS0000016612/authorisations"
    },
    "status": {
      "href":
"/v1/consents/OLS4A06EQGX3P47ODJG2L2DNICR8JS0000016612/status"
    }
  }
}
```

The consent was successfully created, but it has not yet been authorized. Extract the "consentId" value for later use.

Create an authorization resource for a consent

Before a consent can be authorized by PSU with SCA, an authorization resource must be created for the consent. The following endpoint must be invoked:

POST https://api.xs2a/v1/consents/\${consentId}/authorisations			
https://psd2-sandbox.fininbox.com/#/Consents/startConsentAuthorisation			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"

Let's create a consent.

HTTP Request

```
POST
https://api.xs2a/v1/consents/OLS4A06EQGX3P470DJG2L2DNICR8JS0000016612/authorisations
Authorization:Bearer _ACCESS_TOKEN
X-Request-ID:70cc22b0-004c-4375-8215-aec00933b238
```

HTTP Response

```
Status: 201
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2019 07:35:32 GMT
Content-Type: application/json

{
  "authorisationId": "GFVNSKX4FZODIUDVDPRYSMUZPENUVUY0000016616",
  "scaMethods": [
    {
      "name": "MobileID"
    },
    {
      "name": "Redirect"
    }
  ],
  "_links": {
    "scaStatus": {
      "href":
"/v1/consents/OLS4A06EQGX3P470DJG2L2DNICR8JS0000016612/authorisations/GFVN
SKX4FZODIUDVDPRYSMUZPENUVUY0000016616"
    },
    "selectAuthenticationMethod": {
      "href":
"/v1/consents/OLS4A06EQGX3P470DJG2L2DNICR8JS0000016612/authorisations/GFVN
SKX4FZODIUDVDPRYSMUZPENUVUY0000016616"
    }
  }
}
```

The authorization resource was successfully created. The "scaMethods" property provides a list of available SCA methods. Extract the "authorisationId" value for later use.

Choose SCA Redirect

To choose an SCA method, we must update the created authorization resource. The following endpoint must be invoked:

```
PUT https://api.xs2a/v1/consents/{consentId}/authorisations/{authorizationId}
```

<https://psd2-sandbox.fininbox.com/#/Consents/updateConsentsPsuData>

HTTP Headers

Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"

HTTP body defined in <https://psd2-sandbox.fininbox.com/#/Consents/updateConsentsPsuData>

Let's choose an SCA Redirect method. This method will allow to redirect PSU to its ASPSP where it will be able to authorize the consent with its preferred SCA.

```

HTTP Request

PUT
https://api.xs2a/v1/consents/OLS4A06EQGX3P470DJG2L2DNICR8JS0000016612/auth
orisations/GFVNSKX4FZODIUDVDPRYSMUZPENUVY0000016616
Authorization:Bearer _ACCESS_TOKEN
Content-Type:application/json
X-Request-ID:840ade10-c743-4d2c-80a5-d69d7d18b4f5

{
  "authenticationMethodId": "Redirect"
}

```

```

HTTP Response

Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 07:35:42 GMT
Content-Type:application/json

{
  "_links": {
    "scaStatus": {
      "href":
"/v1/consents/OLS4A06EQGX3P470DJG2L2DNICR8JS0000016612/authorisations/GFVN
SKX4FZODIUDVDPRYSMUZPENUVY0000016616"
    },
    "scaRedirect": {
      "href":
"http://ib.xs2a/ib/site/psd2/login?transactionIdsString=905560"
    }
  },
  "scaStatus": "scaMethodSelected"
}

```

An update was successful, it returns JSON property "scaStatus" with value "scaMethodSelected". The "scaRedirect" property provides a URL to an ASPSP authorization interface. The "redirect_uri" parameter must be appended to this URL, so that ASPSP could redirect back to TPP after PSU finishes authorization. A full URL with an appended "redirect_uri" could look like this:

http://ib.xs2a/ib/site/psd2/login?transactionIdsString=905560&redirect_uri=http://tpp.com/users/1

Now PSU must be redirected to that URL to start the consent authorization.

Consent authorization status

When PSU finishes the authorization process, TPP can check the consent authorization status. The following endpoint must be invoked:

GET https://api.xs2a/v1/consents/{consentId}/authorisations/{authorizationId}			
https://psd2-sandbox.fininbox.com/#/Consents/getConsentScaStatus			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"

Let's check the status.

HTTP Request

```
GET
https://api.xs2a/v1/consents/OLS4A06EQGX3P470DJG2L2DNICR8JS0000016612/authorisations/GFVNSKX4FZODIUDVDPYRSMUZPENUVUY0000016616
Authorization:Bearer _ACCESS_TOKEN
Content-Type:application/json
X-Request-ID:33454916-0543-42f8-bb62-843be8073e6d
```

HTTP Response

```
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 12:34:22 GMT
Content-Type:application/json

{
  "scaStatus": "finalised"
}
```

If the "scaStatus" property value is "finalised", then this authorization is finalized.

Consent status

Now the consent status can be verified to make sure that the consent is ready to be used. The following endpoint must be invoked:

GET https://api.xs2a/v1/consents/{consentId}/status			
https://psd2-sandbox.fininbox.com/#/Consents/getConsentStatus			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"

Let's check the consent status.

HTTP Request

```

GET
https://api.xs2a/v1/consents/OLS4A06EQGX3P470DJG2L2DNICR8JS0000016612/status
Authorization:Bearer _ACCESS_TOKEN
X-Request-ID:3be572ed-450f-4c21-a990-a7a089f7f1a2

```

HTTP Response

```

Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 12:34:34 GMT
Content-Type:application/json

{
  "consentStatus": "valid"
}

```

If the "consentStatus" property value is "valid", then the consent is ready to be used.

Accounts information

In order to get a list of accounts, invoke the following endpoint:

GET https://api.xs2a/v1/accounts			
https://psd2-sandbox.fininbox.com/#/Accounts/getAccountList			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"

PSU-Initiated	Boolean	Mandatory	true – information access was initiated by PSU and PSU is aware of it. false – information access was initiated by TPP.
Consent-ID	String	Mandatory	PSU granted consent ID.

Let's get the account information:

HTTP Request

```
GET https://api.xs2a/v1/accounts
Authorization:Bearer _ACCESS_TOKEN
Consent-ID:OLS4A06EQGX3P47ODJG2L2DNICR8JS0000016612
PSU-Initiated:true
X-Request-ID:e9dd4b5a-4103-48ee-94c0-ce7dd4d31911
```

HTTP Response

```
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 07:41:29 GMT
Content-Type:application/json

{
  "accounts": [
    {
      "resourceId": "9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614",
      "iban": "LT044010000100439350",
      "currency": "EUR",
      "_links": {
        "balances": {
          "href":
            "/v1/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/balances"
        },
        "self": {
          "href": "/v1/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614"
        },
        "transactions": {
          "href":
            "/v1/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions"
        }
      }
    }
  ]
}
```

Each account has its own "resourceId" property, this property is required to access a specific account's balances and transactions.

Account balances

In order to get the account's balances, invoke the following endpoint:

GET https://api.xs2a/v1/accounts/{resourceId}/balances			
https://psd2-sandbox.fininbox.com/#/Accounts/getBalances			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"
PSU-Initiated	Boolean	Mandatory	true – information access was initiated by PSU and PSU is aware of it. false – information access was initiated by TPP.
Consent-ID	String	Mandatory	PSU granted consent ID.

Let's get the account balances.

HTTP Request

```
GET
https://api.xs2a/v1/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/balances
Authorization:Bearer _ACCESS_TOKEN
Consent-ID:OLS4A06EQGX3P47ODJG2L2DNICR8JS0000016612
PSU-Initiated:false
X-Request-ID:09976dbd-d435-4173-a267-c8f72345f672
```

HTTP Response

Status: 200
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2019 12:49:30 GMT
Content-Type: application/json

```
{
  "account": {
    "iban": "LT044010000100439350",
    "currency": "EUR"
  },
  "balances": [
    {
      "balanceAmount": {
        "currency": "EUR",
        "amount": "54.05"
      },
      "balanceType": "closingBooked",
      "creditLimitIncluded": false,
      "referenceDate": "2019-09-09"
    },
    {
      "balanceAmount": {
        "currency": "EUR",
        "amount": "52.05"
      },
      "balanceType": "interimAvailable",
      "creditLimitIncluded": true,
      "referenceDate": "2019-09-09"
    }
  ],
  "_links": {
    "self": {
      "href":
"/v1/accounts/9HXBMUEARZZYDBABB3GFVMFX56YJCU0000016614/balances"
    },
    "transactions": {
      "href":
"/v1/accounts/9HXBMUEARZZYDBABB3GFVMFX56YJCU0000016614/transactions"
    },
    "account": {
      "href": "/v1/accounts/9HXBMUEARZZYDBABB3GFVMFX56YJCU0000016614"
    }
  }
}
```

Account transactions

In order to get the account's transactions, invoke the following endpoint:

GET https://api.xs2a/v1/accounts/{resourceId}/transactions

https://psd2-sandbox.fininbox.com/#/Accounts/getTransactionList

HTTP Headers

Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"
PSU-Initiated	Boolean	Mandatory	true – information access was initiated by PSU and PSU is aware of it. false – information access was initiated by TPP.
Consent-ID	String	Mandatory	PSU granted consent ID.

HTTP query parameters

dateFrom	String	Mandatory	Starting date (inclusive the date dateFrom) of the transaction list.
dateTo	Date	Optional	End date (inclusive the data dateTo) of the transaction list, default is "now" if not given.
bookingStatus	Date	Optional	Booking statuses. Available values: booked

Let's get account transactions.

HTTP Request

```
GET
https://api.xs2a/v1/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions?dateFrom=2019-06-11
Authorization:Bearer _ACCESS_TOKEN
Consent-ID:OLS4A06EQGX3P47ODJG2L2DNICR8JS0000016612
PSU-Initiated:false
X-Request-ID:36b63034-d730-4d6e-8ba5-b714acc198f9
```

HTTP Response

Status: 200

Transfer-Encoding: chunked

Connection: keep-alive

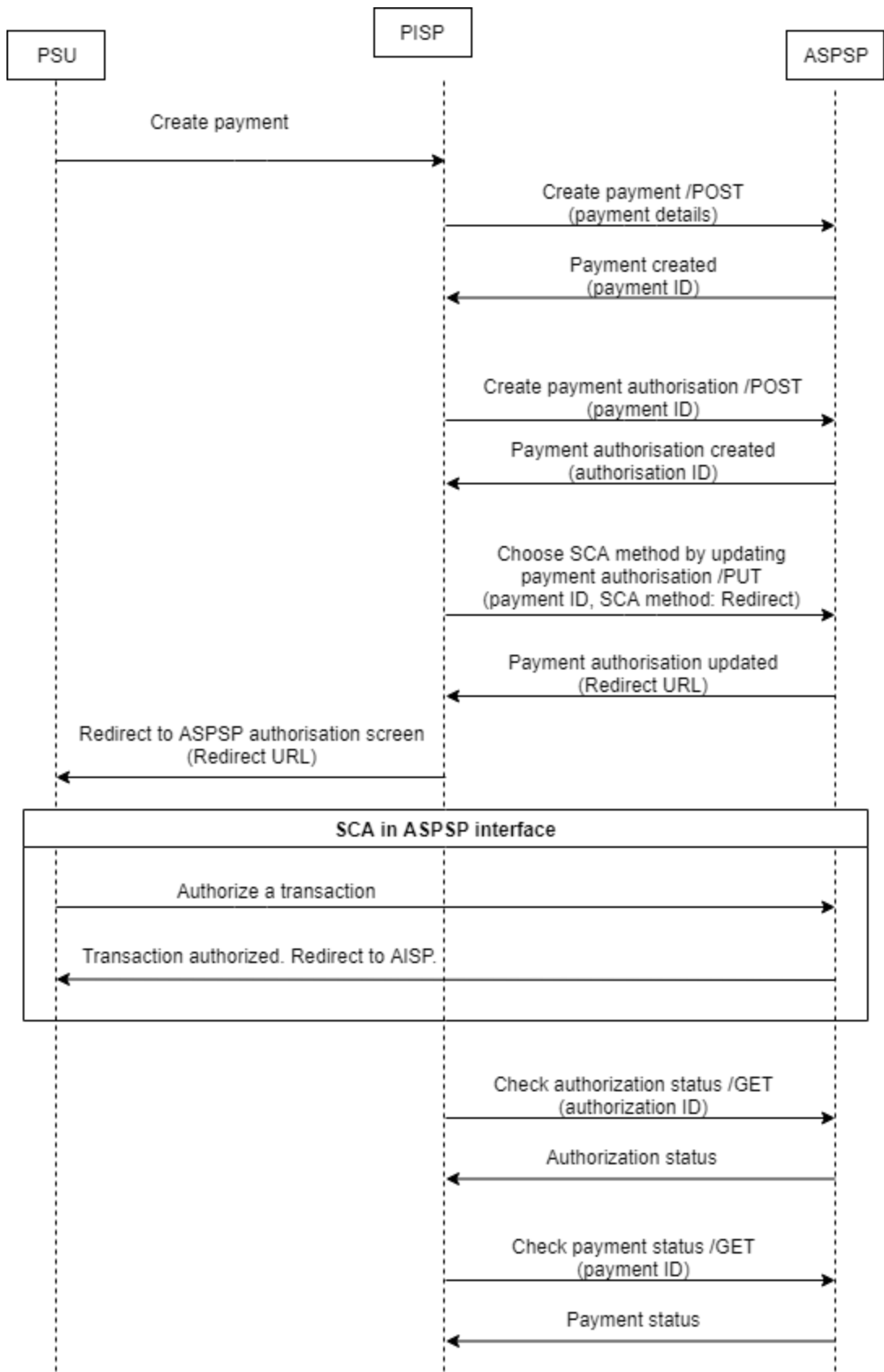
Date: Mon, 09 Sep 2019 12:56:26 GMT

Content-Type: application/json

```
{
  "account": {
    "iban": "LT044010000100439350",
    "currency": "EUR"
  },
  "transactions": {
    "booked": [],
    "pending": []
  },
  "balances": [
    {
      "balanceAmount": {
        "currency": "EUR",
        "amount": "54.05"
      },
      "balanceType": "closingBooked",
      "creditLimitIncluded": false,
      "referenceDate": "2019-09-09"
    },
    {
      "balanceAmount": {
        "currency": "EUR",
        "amount": "52.05"
      },
      "balanceType": "interimAvailable",
      "creditLimitIncluded": true,
      "referenceDate": "2019-09-09"
    }
  ],
  "_links": {
    "balances": {
      "href":
"/v1/accounts/9HXBMUEARZZYDBABB3GFVMFX56YJCU0000016614/balances"
    },
    "self": {
      "href":
"/v1/accounts/9HXBMUEARZZYDBABB3GFVMFX56YJCU0000016614/transactions"
    },
    "account": {
      "href": "/v1/accounts/9HXBMUEARZZYDBABB3GFVMFX56YJCU0000016614"
    }
  }
}
```

PIS services

PIS services allow TPP to make payments from PSU accounts. The payments must be authorized by PSU with SCA.



Follow the example below to understand the steps needed to make a payment.

Example

Create a payment

A payment must be created with its details. The following endpoint must be invoked:

POST https://api.xs2a/v1/payments/sepa-credit-transfers			
https://psd2-sandbox.fininbox.com/#/Payments/initiatePayment			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"
HTTP body defined in https://psd2-sandbox.fininbox.com/#/Payments/initiatePayment			

Let's create a SEPA payment.

HTTP Request

```
POST https://api.xs2a/v1/payments/sepa-credit-transfers
Authorization:Bearer _ACCESS_TOKEN
Content-Type:application/json
X-Request-ID:ed70e51a-d977-4f06-9de8-331a0eac12d3

{
  "requestedExecutionDate": "2019-10-20",
  "endToEndIdentification": "12345",
  "instructedAmount": {
    "currency": "EUR",
    "amount": "123.50"
  },
  "debtorAccount": {
    "iban": "LT044010000100439350"
  },
  "creditorName": "PSD2 Demo Creditor",
  "creditorAccount": {
    "iban": "LT377300012345678901"
  },
  "remittanceInformationUnstructured": "PSD2 Reason of payment"
}
```

HTTP Response

```
Status: 201
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2019 07:38:03 GMT
Content-Type: application/json
```

```
{
  "transactionStatus": "ACCP",
  "paymentId": "905562",
  "transactionFees": {
    "currency": "EUR",
    "amount": "0"
  },
  "transactionFeeIndicator": false
}
```

The payment was successfully created. Now it should be authorized by PSU with SCA. Extract the "paymentId" value for later use.

Payment authorization resource

Before a payment can be authorized by PSU with SCA, an authorization resource must be created for the payment. The following endpoint must be invoked:

POST https://api.xs2a/v1/payments/sepa-credit-transfers/\${paymentId}/authorisations			
https://psd2-sandbox.fininbox.com/#/Payments/startPaymentAuthorisation			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"

Create an authorization resource for the payment.

HTTP Request

```
POST
https://api.xs2a/v1/payments/sepa-credit-transfers/905562/authorisations
Authorization: Bearer _ACCESS_TOKEN
X-Request-ID: 776f9342-32ab-406c-b966-0c0edbfd40f1
```

HTTP Response

```
Status: 201
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2019 07:38:09 GMT
Content-Type: application/json

{
  "authorisationId": "UQMNRIBFPZHY98ITNW4Q57T5UJJMI0000016620",
  "scaMethods": [
    {
      "name": "MobileID"
    },
    {
      "name": "Redirect"
    }
  ],
  "_links": {
    "scaStatus": {
      "href":
"/v1/payments/sepa-credit-transfers/905562/authorisations/UQMNRIBFPZHY98I
TNW4Q57T5UJJMI0000016620"
    },
    "selectAuthenticationMethod": {
      "href":
"/v1/payments/sepa-credit-transfers/905562/authorisations/UQMNRIBFPZHY98I
TNW4Q57T5UJJMI0000016620"
    }
  }
}
```

The authorization resource was successfully created. The "scaMethods" property provides a list of available SCA methods. Extract the "authorisationId" value for later use.

Choose Redirect SCA method

In order to choose an SCA method we must update the created authorization resource. The following endpoint must be invoked:

PUT https://api.xs2a/v1/payments/sepa-credit-transfers/{paymentId}/authorisations/{authorisationId}			
https://psd2-sandbox.fininbox.com/#/Payments/updatePaymentPsuData			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"

HTTP body defined in <https://psd2-sandbox.fininbox.com/#/Payments/updatePaymentPsuData>

Let's choose an SCA Redirect method. This method will allow to redirect PSU to its ASPSP where it will be able to authorize the payment with its preferred SCA.

HTTP Request

```
PUT
https://api.xs2a/v1/payments/sepa-credit-transfers/905562/authorisations/U
QMNRIBFPAZHY98ITNW4Q57T5UJJMI0000016620
Authorization:Bearer _ACCESS_TOKEN
Content-Type:application/json
Date:Mon, 09 Sep 2019 05:38:15 GMT
X-Request-ID:b211b4bc-4829-40b4-9a30-cea6742db479

{
  "authenticationMethodId": "Redirect"
}
```

HTTP Response

```
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 07:38:15 GMT
Content-Type:application/json

{
  "_links": {
    "scaStatus": {
      "href":
"/v1/payments/sepa-credit-transfers/905562/authorisations/UQMNRIBFPAZHY98I
TNW4Q57T5UJJMI0000016620"
    },
    "scaRedirect": {
      "href":
"http://ib.xs2a/ib/site/psd2/login?transactionIdsString\u003d905562"
    }
  },
  "scaStatus": "scaMethodSelected"
}
```

An update was successful, it returns JSON property "scaStatus" with value "scaMethodSelected". The "scaRedirect" property provides a URL to an ASPSP authorization interface. A "redirect_uri" parameter must be appended to this URL, so that ASPSP could be redirected back to TPP after PSU finishes authorization. A full URL with the appended "redirect_uri" could look like this:

```
http://ib.xs2a/ib/site/psd2/login?transactionIdsString=905562&redirect_uri=http://tpp.com/users/1
```

Now PSU must be redirected to that URL to start a payment authorization.

Payment authorization status

When PSU finishes an authorization process, TPP can check the payment authorization status. The following endpoint must be

invoked:

GET https://api.xs2a/v1/payments/sepa-credit-transfers/\${paymentId}/authorisations/\${authorizationId}			
https://psd2-sandbox.fininbox.com/#/Payments/getPaymentInitiationScaStatus			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"

Check the payment authorization status.

```
HTTP Request
GET
https://api.xs2a/v1/payments/sepa-credit-transfers/905562/authorisations/U
QMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016620
Authorization:Bearer _ACCESS_TOKEN
X-Request-ID:dca3db1c-d3eb-42db-91cf-0383f4c0ab63
```

```
HTTP Response
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 11:59:21 GMT
Content-Type:application/json

{
  "scaStatus": "finalised"
}
```

If "scaStatus" property value is "finalised", then this authorization is finalized.

Payment status

Now the consent status can be verified to make sure that the consent is ready to be used. The following endpoint must be invoked:

GET https://api.xs2a/v1/payments/sepa-credit-transfers/\${paymentId}/status			
https://psd2-sandbox.fininbox.com/#/Payments/getPaymentInitiationStatus			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"

Check the payment status.

HTTP Request

```
GET https://api.xs2a/v1/payments/sepa-credit-transfers/905562/status
Authorization:Bearer _ACCESS_TOKEN
X-Request-ID:67cebfa0-c775-4079-b38d-8e574555f6d5
```

HTTP Response

```
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 11:59:36 GMT
Content-Type:application/json

{
  "transactionStatus": "ACCC",
  "fundsAvailable": false
}
```

If the "transactionStatus" property value is "valid", then the payment is completed.

PIISP services

It checks whether a specific amount is available at a point of time of the request on the account linked to a given IBAN.

Example

POST https://api.xs2a/v1/funds-confirmations			
https://psd2-sandbox.fininbox.com/#/Funds%20confirmations/checkAvailabilityOfFunds			
HTTP Headers			
Name	Type	Condition	Description
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken"
HTTP body defined in:			
https://psd2-sandbox.fininbox.com/#/Funds%20confirmations/checkAvailabilityOfFunds			

Let's check if the funds are available.

HTTP Request

```
POST https://api.xs2a/v1/funds-confirmations
Authorization:Bearer _ACCESS_TOKEN
Content-Type:application/json
X-Request-ID:79f76293-fa88-4370-b40e-6d4618eeb73b
```

```
{
  "account": {
    "iban": "LT044010000100439350"
  },
  "instructedAmount": {
    "currency": "EUR",
    "amount": "1000"
  }
}
```

HTTP Response

```
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 07:40:52 GMT
Content-Type:application/json
```

```
{
  "fundsAvailable": false
}
```

A response is returned stating that the funds are not available.