

XS2A API Developer Documentation

- [Versions](#)
- [Introduction](#)
 - [Definitions](#)
 - [API Overview](#)
- [OAuth2 API](#)
 - [Open Banking Services API](#)
 - [Authentication and Security](#)
 - [OAuth2 Access Token](#)
 - [Example](#)
 - [OAuth2 access code](#)
 - [Acquire Access token for the first time](#)
 - [Renewal of Access token](#)
 - [Accounts information services API](#)
 - [Example](#)
 - [Get a list of PSU accounts](#)
 - [Verification of account owner](#)
 - [Create a consent](#)
 - [Create an authorization resource for a consent](#)
 - [Choose SCA Redirect](#)
 - [Consent authorization status](#)
 - [Consent status](#)
 - [Accounts information](#)
 - [Account balances](#)
 - [Account transactions not older than 90 days](#)
 - [Account transactions older than 90 days](#)
 - [Create an authorization resource for transactions list](#)
 - [Choose SCA Redirect](#)
 - [Transactions list authorization status](#)
 - [Getting account transactions by statementID](#)
 - [Authorization resource status flow](#)
 - [AIS consent status flow](#)
 - [Payments initiation services API](#)
 - [Example of SEPA payment](#)
 - [Create a SEPA payment](#)
 - [SEPA payment authorization resource](#)
 - [Choose Redirect SCA method](#)
 - [SEPA payment authorization status](#)
 - [SEPA payment status](#)
 - [Example of SEPA Instant payment](#)
 - [Create a SEPA Instant payment](#)
 - [SEPA Instant payment verifications](#)
 - [SEPA Instant payment authorization resource](#)
 - [Choose Redirect SCA method](#)
 - [SEPA Instant payment authorization status](#)
 - [SEPA Instant payment status](#)
 - [Example of International payment](#)
 - [Create an International payment](#)
 - [International payment authorization resource](#)
 - [Choose Redirect SCA method](#)
 - [International payment authorization status](#)
 - [International payment status](#)
 - [Example of Periodic payment](#)
 - [Create a Periodic payment](#)
 - [Periodic payment authorization resource](#)
 - [Choose Redirect SCA method](#)
 - [Periodic payment authorization status](#)
 - [Periodic payment status](#)
 - [Example of Bulk payment](#)
 - [Create a Bulk SEPA payment](#)
 - [Bulk payment authorization resource](#)
 - [Choose Redirect SCA method](#)
 - [Bulk payment authorization status](#)
 - [Bulk payment status](#)
 - [Payment cancellation](#)
 - [Example of SEPA payment cancellation](#)
 - [Example of Periodic payment cancellation](#)
 - [Future dated payments](#)
 - [Countersigning](#)
 - [Debtor account](#)
 - [Payment status flow](#)
 - [PIISP services](#)
 - [Example](#)

- Health service
 - Example
 - Errors
- API versioning
- Errors
 - HTTP statuses
 - Common error messages
- XS2A API Changes in Version V2
 - Changes to the debtorIdentification object structure
 - New objects and parameters in international payment services
 - Changes to the address object structure
 - Changes to the parameter sendDebtorAddress
 - Changes to the error when payment draft is expired
 - Changes in payment authorization status after payment cancellation in Internet bank
 - Changes in AIS consent validity period
 - Changes in OAuth2 token validity period
 - Changes to the parameter frequencyPerDay
 - Correction of AIS services calls counting
 - Changes to the HTTP query parameter redirect_uri_fail
 - Changes to the parameter validUntil
- XS2A API Changes in Version V3
 - New service GET /health-check/ping
 - New objects ultimateDebtorAddress and ultimateCreditorAddress in international payment services
 - Optional object tppMessages introduced in payment status services answers
 - Optional HTTP parameter PSU-CORPORATE-ID introduced in /accounts-list service
 - HTTP parameter X-Request-ID obligation
 - Account transactions older than 90 days

Versions

Version	Changes
1.0	Initial version
2.0	Changes for XS2A API V2.
3.0	Changes for XS2A API V3.
3.1	Parameters Consent-ID, Statement-id, PSU-Initiated added for services: POST /{version}/accounts/{resourceId}/transactions/{statementId}/authorisations PUT /{version}/accounts/{resourceId}/transactions/{statementId}/authorisations/{authorizationId} GET /{version}/accounts/{resourceId}/transactions/{statementId}/authorisations/{authorizationId} GET /{version}/accounts/{resourceId}/transactions/{statementId} New error added: PARAMETER_NOT_CONSISTENT Invalid period dates specified!
3.2	AISP can access each consented payment account data without the customer being directly involved four times a day.
3.3	SEPA Instant payment status error messages added.

Introduction

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

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

Definitions

Notion	Description
PSD2 XS2A (or XS2A)	"Access to account" services as defined under the 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	Account Servicing Payment Service 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.
Developer portal	A site dedicated to Open banking solution description, provided by ASPSP.

API Overview

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

OAuth2 API

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

Open Banking Services API

Open Banking 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.

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

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

OAuth2 Access Token

Prior to invoking of any of the PIS, AIS, PIIS services, an OAuth2 access token must be acquired. Acquiring the access token involves PSU that will have to authorize it. Access and refresh tokens will be provided. The access token is valid for 30 minutes, after that it must be renewed with a refresh token. A refresh token is valid for 180 days. After refresh token's expiration, they must be again acquired and authorized by PSU.

The access token is authorized by redirecting PSU to the ASPSP interface. Once the access token has been authorized, 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 see the steps needed to acquire the access token.

Example

OAuth2 access code

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

GET https://api.xs2a/{version}/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=SANDBOX for Sandbox environment, please ask for PROD environment)

Then acquire an authorization URL.

HTTP Request
GET https://api.xs2a/{version}/oauth/authorization/links?branch=SANDBOX 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 that contains a URL where the PSU must be redirected. Append "redirect_uri", "redirect_uri_fail" and "state" query parameters to this URL. "redirect_uri" - location where PSU should be redirected when authorization has finished. "redirect_uri_fail" - mandatory parameter; a location where PSU should be redirected when authorization has failed. "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&redirect_uri_fail=http://tpp.com/users/1?failed=true
```

Now redirect PSU to this URL.

Acquire Access token for the first time

Once PSU has finished the access token authorization in the ASPSP interface, it is redirected to "redirect_uri". In our 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. This parameter is used to acquire the access token. The following endpoint must be invoked:

POST https://api.xs2a/{version}/oauth/token			
HTTP Headers			
Name	Type	Condition	Description

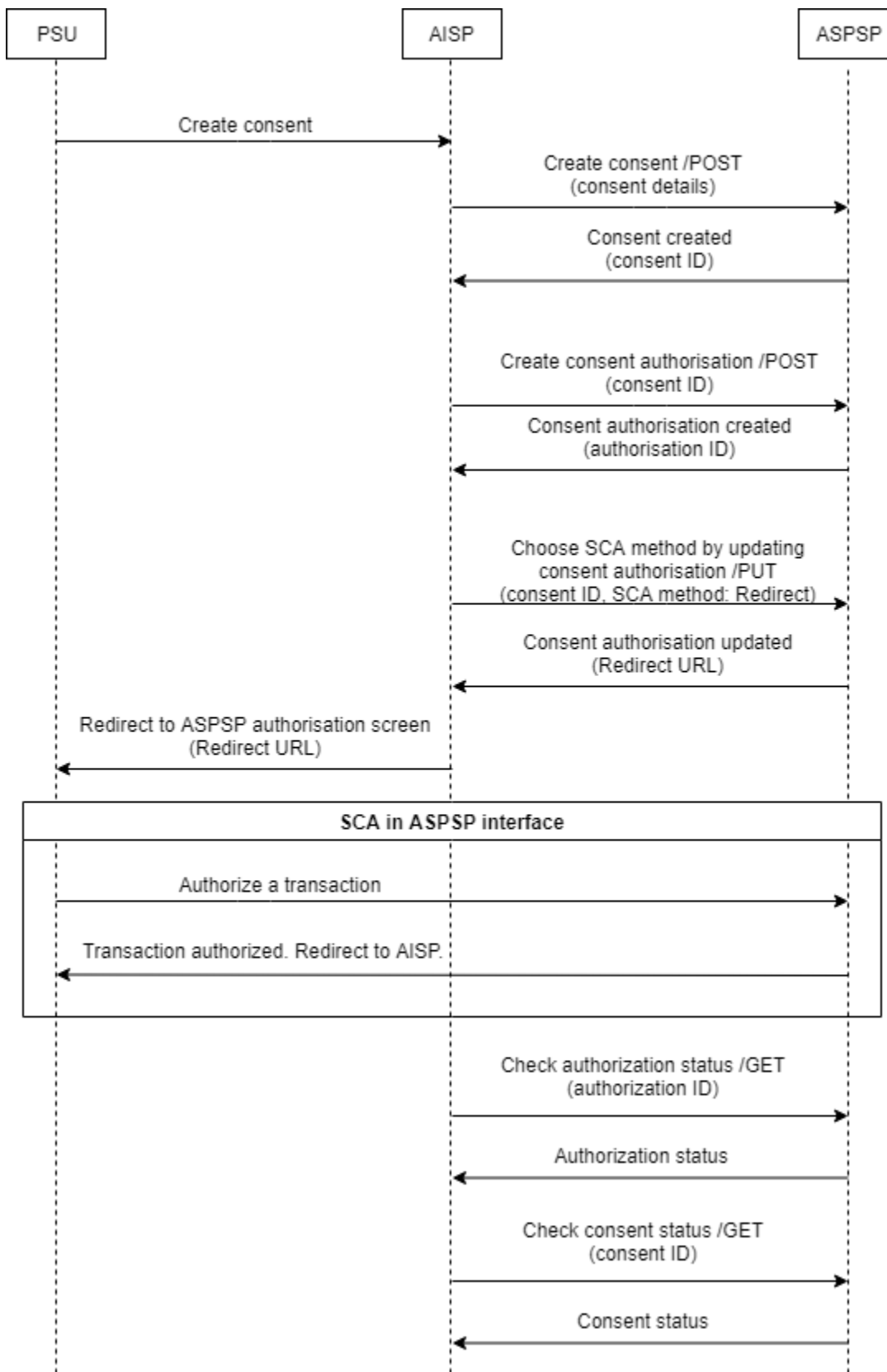
X-Request-ID	String	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
Content-Type	String	Mandatory	value: application/json
HTTP query parameters			
Name	Type	Condition	Description
branch	String	Mandatory	ASPSP branch. (branch=SANDBOX for Sandbox environment, please ask for PROD environment)
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:

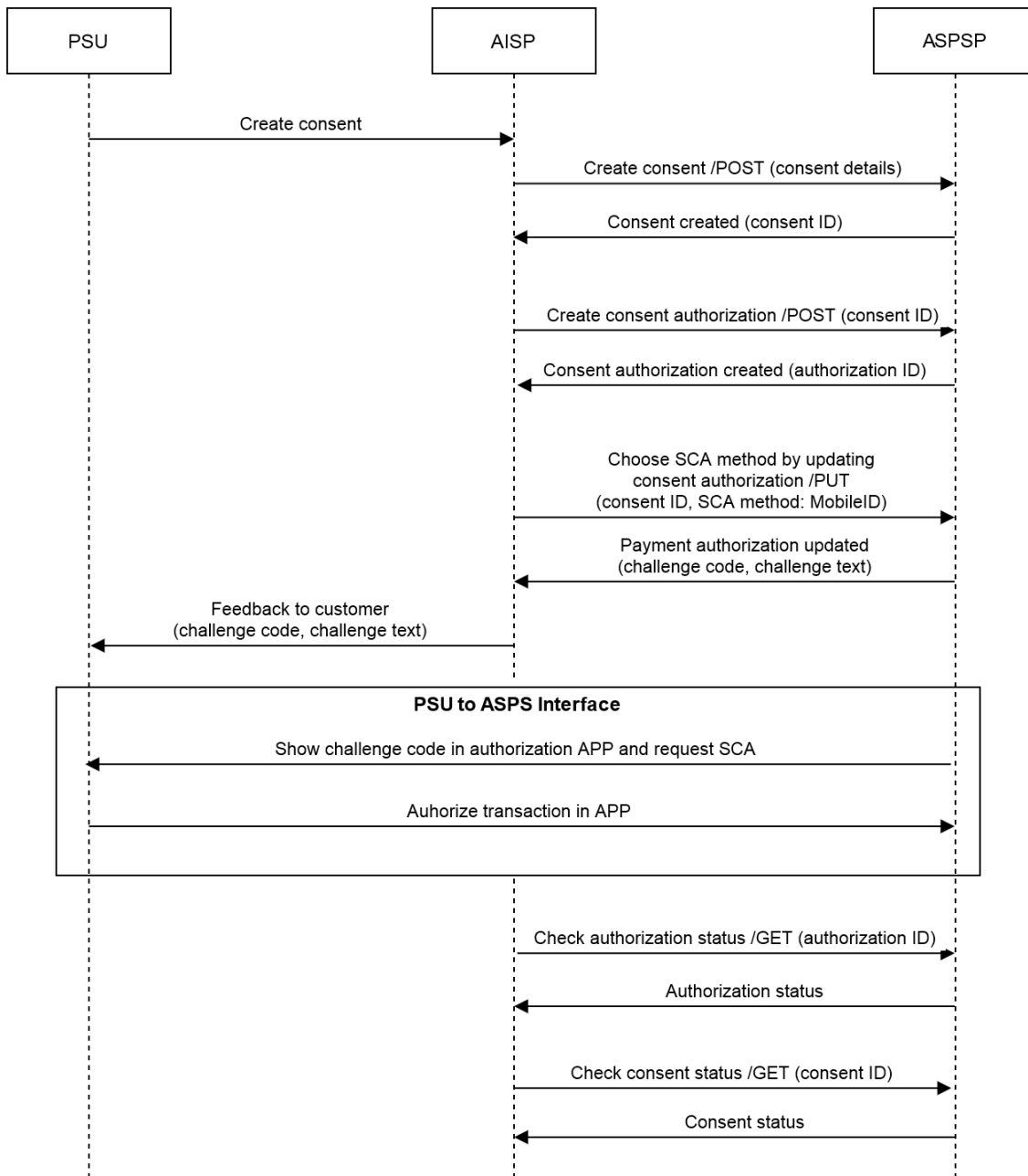
HTTP Request

```
POST https://api.xs2a/{version}/oauth/token?branch=SANDBOX
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"
}
```

Here is a consent authorization flow with Decoupled SCA:



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

Example

Get a list of PSU accounts

The following endpoint can be invoked to obtain a list PSU accounts. This endpoint is available only for TPP that have a AISP role.

GET https://api.xs2a/{version}/accounts-list			
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"
PSU-Corporate-ID	String	Optional	Validation of a Corporate or a Person by Corporate code or Personal code.

Then, obtain a list of PSU accounts.

HTTP Request

```
GET https://api.xs2a/{version}/accounts-list
Authorization:Bearer _ACCESS_TOKEN
Content-Type:application/json
X-Request-ID:f63daf5f-27ae-4992-9eda-940dd2a1dae0
PSU-Corporate-ID:0987654321
```

HTTP Response

```
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2019 07:35:10 GMT
Content-Type:application/json
```

```
[{
  "iban": "LT044010000100439350",
  "currency": "EUR",
  "name": "Account_name",
  "validation": "true"
}]
```

TPP can provide the obtained list of accounts to PSU to choose the account(s) which will be used for AIS or PIS services. If TPP does not have the AISP role, TPP does not provide JSON parameter `debtorAccount` in the PIS services calls. In this case, PSU can choose the debtor account in the ASPSP interface.

Verification of account owner

The endpoint `/accounts-list` provides a possibility to verify the accounts owner. The accounts are checked if the account owner is an individual (*Person*) or a company (*Corporate*) provided in the header `PSU-Corporate-ID`. The property `validation` in the answer represents the validation for every account:

- `true` – the value `PSU-Corporate-ID` was provided, the individual or the company has rights to the account;
- `false` – the value `PSU-Corporate-ID` was provided, the individual or the company has no rights to the account;
- `empty (NULL)` – the value `PSU-Corporate-ID` was not provided.

In Sandbox, the value `PSU-Corporate-ID` must be `12345678901` for the personal code or `0987654321` for the corporate code, or not provided at all – otherwise, an error will be returned.

If *Person/Corporate* verification fails, the HTTP error 401 is returned with the error code `CORPORATE_ID_INVALID` and the text reading: The PSU-Corporate-ID cannot be matched by the addressed ASPSP.

HTTP Response for error when "PSU-Corporate-ID" verification fails

```
Status: 401
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Oct 2023 17:35:56 GMT
Content-Type: application/json
```

```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "CORPORATE_ID_INVALID",
      "text": "The PSU-Corporate-ID cannot be matched by the addressed ASPSP."
    }
  ]
}
```

Create a consent

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

```
POST https://api.xs2a/{version}/consents
```

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 *Developer portal*.

Create a consent that allows TPP to access information, balances, transactions for each consented PSU account. Information for each consented PSU account can be accessed without the PSU involvement 4 times a day until a consent is valid. The maximum validity period for a consent is 180 days.

HTTP Request

```
POST https://api.xs2a/{version}/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": 4
}
```

HTTP Response

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

```
{
  "consentId": "OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612",
  "_links": {
    "self": {
      "href": "/{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612"
    },
    "startAuthorisation": {
      "href": "/{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612/authorisations"
    },
    "status": {
      "href": "/{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612/status"
    }
  }
}
```

The consent has been successfully created, but not yet 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/{version}/consents/{consentId}/authorisations
```

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".

Create an authorization.

HTTP Request
POST https://api.xs2a/{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612/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 <pre> { "authorisationId": "GFVNSKX4FZODIUDVDPRYSMUZPENUVY0000016616", "scaMethods": [{ "name": "SmartID" }, { "name": "MobileID" }, { "name": "Redirect" }], "_links": { "scaStatus": { "href": "/{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612/authorisations/GFVNSKX4FZODIUDVDPRYSMUZPENUVY0000016616" }, "selectAuthenticationMethod": { "href": "/{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612/authorisations/GFVNSKX4FZODIUDVDPRYSMUZPENUVY0000016616" } } } </pre>

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/{version}/consents/{consentId}/authorisations/{authorizationId}			
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 <i>Developer portal</i> .			

Choose an SCA Redirect method. This method will allow to redirect PSU to its ASPSP where he will be able to authorize the consent with its preferred SCA.

HTTP Request
<pre>PUT https://api.xs2a/{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612/authorisations /GFVNSKX4FZODIUDVDPYRYSMUZPENUVY0000016616 Authorization:Bearer _ACCESS_TOKEN Content-Type:application/json X-Request-ID:840ade10-c743-4d2c-80a5-d69d7d18b4f5 { "authenticationMethodId": "Redirect" }</pre>

HTTP Response
<pre>Status: 200 Transfer-Encoding:chunked Connection:keep-alive Date:Mon, 09 Sep 2019 07:35:42 GMT Content-Type:application/json { "_links": { "scaStatus": { "href": "://{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612/authorisations /GFVNSKX4FZODIUDVDPYRYSMUZPENUVY0000016616" }, "scaRedirect": { "href": "http://ib.xs2a/ib/site/psd2/login?transactionIdsString=905560" } }, "scaStatus": "scaMethodSelected" }</pre>

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 in order the ASPSP could redirect back to TPP after PSU finishes authorization. Mandatory parameter "redirect_uri_fail" must be appended in order the ASPSP could redirect to TPP after authorization fail. A complete URL with an appended parameters could look like this:

```
http://ib.xs2a/ib/site/psd2/login?transactionIdsString=905560&redirect_uri=http://tpp.com/users
/l&redirect_uri_fail=http://tpp.com/users/l?fail=true
```

Now, PSU must be redirected to that URL to start the consent authorization. If confirmation is cancelled in the Internet bank page, the authorization status becomes "failed". A new authorization resource must be created to complete the authorization.

Consent authorization status

Once PSU has finished the authorization process, TPP can check the consent authorization status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/consents/{consentId}/authorisations/{authorizationId}			
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".

Check the status.

HTTP Request
<pre>GET https://api.xs2a/{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612/authorisations /GFVNSKX4FZODIUDVDPYRSMUZPENUVY0000016616 Authorization:Bearer _ACCESS_TOKEN Content-Type:application/json X-Request-ID:33454916-0543-42f8-bb62-843be8073e6d</pre>

HTTP Response
<pre>Status: 200 Transfer-Encoding:chunked Connection:keep-alive Date:Mon, 09 Sep 2019 12:34:22 GMT Content-Type:application/json { "scaStatus": "finalised" }</pre>

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

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/{version}/consents/{consentId}/status			
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".

Check the consent status.

HTTP Request
<pre>GET https://api.xs2a/{version}/consents/OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612/status Authorization:Bearer _ACCESS_TOKEN X-Request-ID:3be572ed-450f-4c21-a990-a7a089f7f1a2</pre>

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/{version}/accounts
```

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	<ul style="list-style-type: none">• true – information access was initiated by PSU and PSU is aware of it.• false – information access was initiated by TPP.
Consent-ID	String	Mandatory	ID of PSU granted consent.

Get the account information:

HTTP Request

```
GET https://api.xs2a/{version}/accounts
Authorization: Bearer _ACCESS_TOKEN
Consent-ID: OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612
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": "/{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/balances"
        },
        "self": {
          "href": "/{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614"
        },
        "transactions": {
          "href": "/{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions"
        }
      }
    }
  ]
}
```

Each account has its own "resourceId" property. This property is required to access the specific account's balances and transactions.

Account balances

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

GET https://api.xs2a/{version}/accounts/{resourceId}/balances			
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	<ul style="list-style-type: none">true – information access was initiated by PSU and PSU is aware of it.false – information access was initiated by TPP.
Consent-ID	String	Mandatory	ID of PSU granted consent

Get the account balances.

HTTP Request

```
GET https://api.xs2a/{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/balances
Authorization: Bearer _ACCESS_TOKEN
Consent-ID: OLS4A06EQGX3P470DJG2L2DNICR8JS0000016612
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": "/{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/balances"
    },
    "transactions": {
      "href": "/{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions"
    },
    "account": {
      "href": "/{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614"
    }
  }
}
```

Account transactions not older than 90 days

To get the account's transactions not older than 90 days, invoke the following endpoint:

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

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	<ul style="list-style-type: none">true – information access was initiated by PSU and PSU is aware of it.false – information access was initiated by TPP.
Consent-ID	String	Mandatory	ID of PSU granted consent.

HTTP query parameters

dateFrom	String	Mandatory	Starting date (inclusive the date dateFrom) of the transaction list. Not earlier than 90 days.
dateTo	Date	Optional	End date (inclusive the data dateTo) of the transaction list, default is "now" if not given.
bookingStatus	String	Optional	Booking statuses. Available values: booked.

Get the account's transactions.

HTTP Request

```
GET https://api.xs2a/{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions?dateFrom=2023-06-11
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 2023 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": "2023-09-09"
    },
    {
      "balanceAmount": {
        "currency": "EUR",
        "amount": "52.05"
      },
      "balanceType": "interimAvailable",
      "creditLimitIncluded": true,
      "referenceDate": "2023-09-09"
    }
  ],
  "_links": {
    "balances": {
      "href": "{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/balances"
    },
    "self": {
      "href": "{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions"
    },
    "account": {
      "href": "{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614"
    }
  }
}
```

Account transactions older than 90 days

To get the account's transactions older than 90 days, a statement should be authorized by PSU. The following endpoint must be invoked to get `statementID`:

POST `https://api.xs2a/{version}/accounts/{resourceId}/transactions`

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	<ul style="list-style-type: none"> • true – information access was initiated by PSU and PSU is aware of it. • false – information access was initiated by TPP.
Consent-ID	String	Mandatory	ID of PSU granted consent.
HTTP query parameters			
dateFrom	String	Mandatory	Starting date (inclusive the date dateFrom) of the transaction list. Earlier than 90 days.
dateTo	Date	Optional	End date (inclusive the data dateTo) of the transaction list, default is "now" if not given. Period from "dateFrom" till "dateTo" cannot exceed 1 year.
bookingStatus	String	Optional	Booking statuses. Available values: booked.

Order account's transactions list:

HTTP Request
<pre>POST https://api.xs2a/{version}/accounts/9HXBMEARZZYDBABB3GFVVMFX56YJCU0000016614/transactions?dateFrom=2023-06-11 Authorization:Bearer _ACCESS_TOKEN Consent-ID:OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612 PSU-Initiated:true X-Request-ID:36b63034-d730-4d6e-8ba5-b714acc198f9</pre>

HTTP Response
<pre>Status: 201 Transfer-Encoding:chunked Connection:keep-alive Date:Mon, 09 Sep 2023 12:56:26 GMT Content-Type:application/json { "statementId": 456, "_links": { "startAuthorisation": { "href": "/{version}/accounts/123/transactions/456/authorisations" } } }</pre>

Create an authorization resource for transactions list

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

POST https://api.xs2a/{version}/accounts/{resourceId}/transactions/{statementId}/authorisations			
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	<ul style="list-style-type: none"> • true – information access was initiated by PSU and PSU is aware of it. • false – information access was initiated by TPP.

Consent-ID	String	Mandatory	ID of PSU granted consent.
------------	--------	-----------	----------------------------

Create an authorization:

HTTP Request

```
POST https://api.xs2a/{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions/456/authorisations
Authorization:Bearer _ACCESS_TOKEN
Consent-ID:OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612
PSU-Initiated:true
X-Request-ID:70cc22b0-004c-4375-8215-aec00933b238
```

HTTP Response

```
Status: 201
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2023 07:35:32 GMT
Content-Type:application/json

{
  "authorisationId": "GFVNSKX4FZODIUDVDPRYSMUZPENUVUY0000016619",
  "scaMethods": [
    {
      "name": "SmartID"
    },
    {
      "name": "MobileID"
    },
    {
      "name": "Redirect"
    }
  ],
  "_links": {
    "scaStatus": {
      "href": "://{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions/456/authorisations/GFVNSKX4FZODIUDVDPRYSMUZPENUVUY0000016619"
    },
    "selectAuthenticationMethod": {
      "href": "://{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions/456/authorisations/GFVNSKX4FZODIUDVDPRYSMUZPENUVUY0000016619"
    }
  }
}
```

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, update the created authorization resource. The following endpoint must be invoked:

PUT https://api.xs2a/{version}/accounts/{resourceId}/transactions/{statementId}/authorisations/{authorizationId}			
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".

Consent-ID	String	Mandatory	ID of PSU granted consent.
PSU-Initiated	Boolean	Mandatory	<ul style="list-style-type: none"> true – information access was initiated by PSU and PSU is aware of it. false – information access was initiated by TPP.
HTTP body defined in <i>Developer portal</i> .			

Choose an SCA Redirect method. This method will allow to redirect PSU to its ASPSP where he will be able to authorize the consent with its preferred SCA.

HTTP Request

```
PUT https://api.xs2a/{version}/accounts/9HXBMUEARZZYDBABB3GFVMFX56YJCU0000016614/transactions/456/authorisations
/GFVNSKX4FZODIUDVDPRYSMUZPENVUY0000016619
Authorization:Bearer _ACCESS_TOKEN
Content-Type:application/json
Consent-ID:OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612
PSU-Initiated:true
X-Request-ID:840ade10-c743-4d2c-80a5-d69d7d18b4f5

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

HTTP Response

```
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2023 07:35:42 GMT
Content-Type:application/json

{
  "_links": {
    "scaStatus": {
      "href": "://{version}/accounts/9HXBMUEARZZYDBABB3GFVMFX56YJCU0000016614/transactions/456/authorisations
/GFVNSKX4FZODIUDVDPRYSMUZPENVUY0000016619"
    },
    "scaRedirect": {
      "href": "http://ib.xs2a/ib/site/psd2/login?statementIdsString=456"
    }
  },
  "scaStatus": "scaMethodSelected"
}
```

The 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 in order the ASPSP could redirect back to TPP after PSU finishes authorization. Mandatory parameter "redirect_uri_fail" must be appended in order the ASPSP could redirect to TPP after authorization fail. A complete URL with an appended parameters could look like this:

```
http://ib.xs2a/ib/site/psd2/login?statementIdsString=456&redirect_uri=http://tpp.com/users/1&redirect_uri_fail=http://tpp.com/users/1?fail=true
```

Now, PSU must be redirected to that URL to start the consent authorization. If confirmation is cancelled in the Internet bank page, the authorization status becomes "failed". A new authorization resource must be created to complete the authorization.

Transactions list authorization status

Once PSU has finished the authorization process, TPP can check the consent authorization status. The following endpoint must be invoked:

```
GET https://api.xs2a/{version}/accounts/{resourceId}/transactions/{statementId}/authorisations/
{authorizationId}
```

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	<ul style="list-style-type: none"> true – information access was initiated by PSU and PSU is aware of it. false – information access was initiated by TPP.
Consent-ID	String	Mandatory	ID of PSU granted consent.

HTTP Request

```
GET https://api.xs2a/{version}/accounts/9HXBMEARZZYDBABB3GFVVMFX56YJCU0000016614/transactions/456/authorisations/GFVNSKX4FZODIUDVDPYRSMUZPENUVY0000016619
Authorization:Bearer _ACCESS_TOKEN
Consent-ID:OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612
PSU-Initiated:true
X-Request-ID:33454916-0543-42f8-bb62-843be8073e6d
```

HTTP Response

```
Status: 200
Transfer-Encoding:chunked
Connection:keep-alive
Date:Mon, 09 Sep 2023 12:34:22 GMT
Content-Type:application/json
```

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

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

Getting account transactions by `statementID`

In order to get account's transactions older than 90 days by `statementID`, following endpoint must be invoked:

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	<ul style="list-style-type: none"> true – information access was initiated by PSU and PSU is aware of it. false – information access was initiated by TPP.
Consent-ID	String	Mandatory	ID of PSU granted consent.

HTTP Request

```
GET https://api.xs2a/{version}/accounts/9HXBMEARZZYDBABB3GFVVMFX56YJCU0000016614/transactions/456
Authorization:Bearer _ACCESS_TOKEN
Consent-ID:OLS4AO6EQGX3P47ODJG2L2DNICR8JS0000016612
PSU-Initiated:true
X-Request-ID:33454916-0543-42f8-bb62-843be8073e6d
```

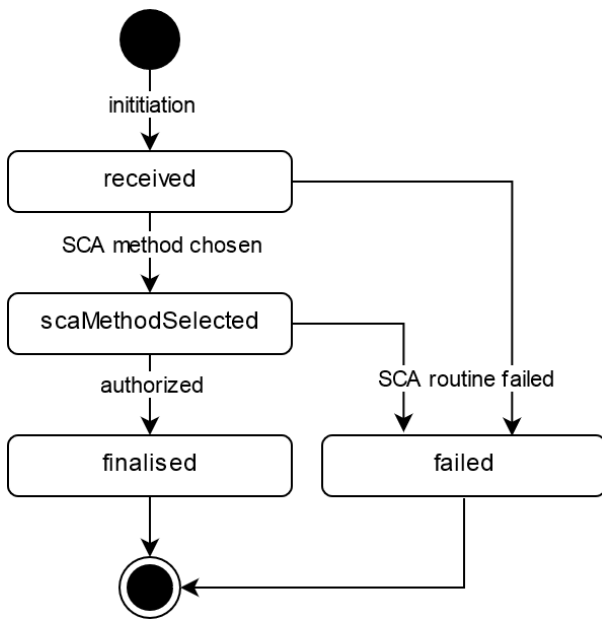
HTTP Response

Status: 200
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2023 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": "2023-09-09"
    },
    {
      "balanceAmount": {
        "currency": "EUR",
        "amount": "52.05"
      },
      "balanceType": "interimAvailable",
      "creditLimitIncluded": true,
      "referenceDate": "2023-09-09"
    }
  ],
  "_links": {
    "balances": {
      "href": "{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/balances"
    },
    "self": {
      "href": "{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614/transactions"
    },
    "account": {
      "href": "{version}/accounts/9HXBMEARZZYDBABB3GFVMFX56YJCU0000016614"
    }
  }
}
```

If SCA is not completed for account's transactions list, **HTTP Error 428** is returned.

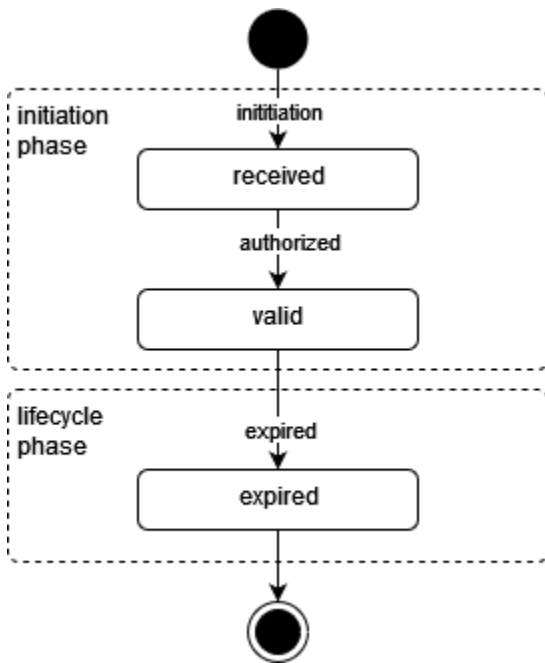
Authorization resource status flow



Authorization resource states description:

1. **received** – a newly created authorization resource;
2. **scaMethodSelected** – state after the update of authorization resource, when SCA method is selected;
3. **finalised** – PSU has confirmed an authorization resource;
4. **failed** – SCA routine has failed.

AIS consent status flow



AIS consent states description:

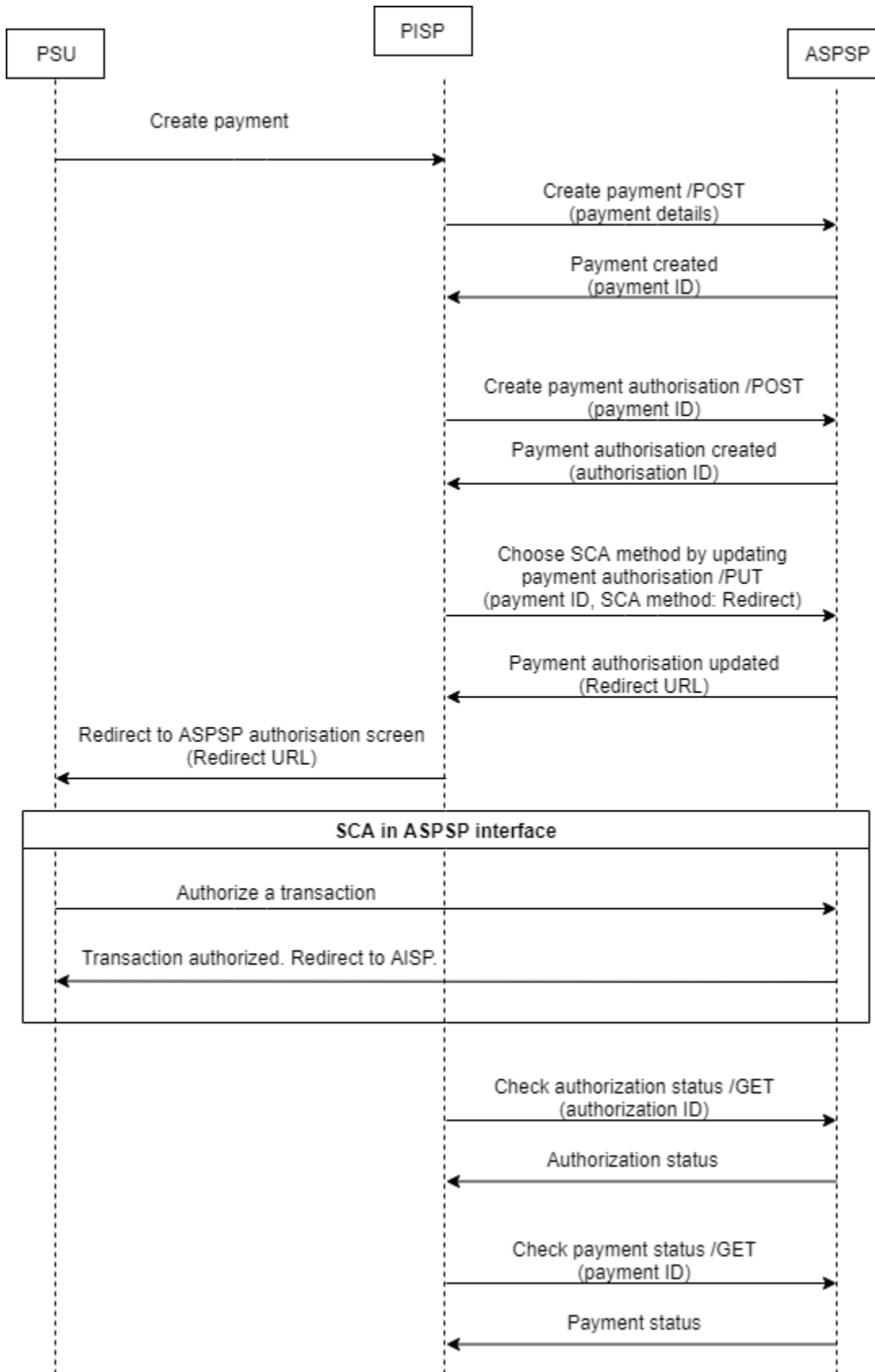
1. **received** – a newly registered consent;
2. **valid** – a consent is confirmed by PSU with SCA;
3. **expired** – a consent has expired.

If consent is revoked by PSU or terminated by TPP, consent is deleted and consent information service returns error that such consent does not exist.

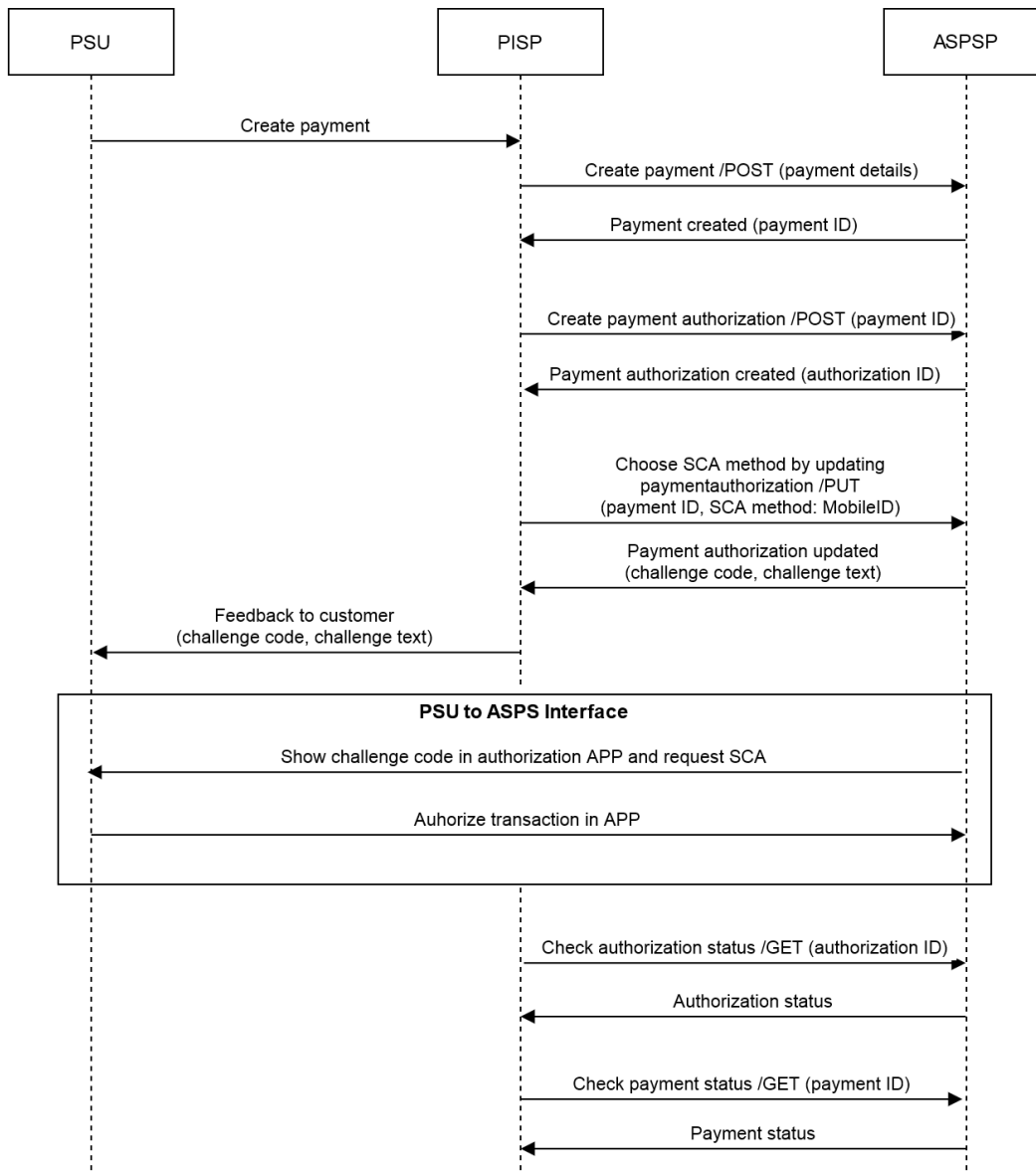
Payments initiation services API

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

Here is a payment flow with Redirect SCA approach:



Here is a payment flow with Decoupled SCA approach:



Follow examples below to understand the steps needed to make certain payments.

Example of SEPA payment

Create a SEPA payment

A payment must be created with its details. The following endpoint must be invoked to create a SEPA payment:

POST <code>https://api.xs2a/{version}/payments/sepa-credit-transfers</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
Authorization	String	Mandatory	The value is "Bearer " followed by an access token, i.e. "Bearer encodedAccessToken".
HTTP body defined in <i>Developer portal</i> .			

Create a SEPA payment.

HTTP Request

```
POST https://api.xs2a/{version}/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": "ACTC",
  "paymentId": "905562",
  "transactionFees": {
    "currency": "EUR",
    "amount": "0"
  },
  "transactionFeeIndicator": false
}
```

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

SEPA 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/{version}/payments/sepa-credit-transfers/{paymentId}/authorisations
```

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/{version}/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": "UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016620",
  "scaMethods": [
    {
      "name": "SmartID"
    },
    {
      "name": "MobileID"
    },
    {
      "name": "Redirect"
    }
  ],
  "_links": {
    "scaStatus": {
      "href": "/{version}/payments/sepa-credit-transfers/905562/authorisations
/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016620"
    },
    "selectAuthenticationMethod": {
      "href": "/{version}/payments/sepa-credit-transfers/905562/authorisations
/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016620"
    }
  }
}
```

The authorization resource has been 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, it is required to update the created authorization resource. The following endpoint must be invoked:

```
PUT https://api.xs2a/{version}/payments/sepa-credit-transfers/{paymentId}/authorisations/{authorisationId}
```

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 *Developer portal*.

Choose an SCA Redirect method. This method will allow redirecting PSU to its ASPSP where he will be able to authorize the payment with its preferred SCA.

HTTP Request

```
PUT https://api.xs2a/{version}/payments/sepa-credit-transfers/905562/authorisations
/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016620
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": "{version}/payments/sepa-credit-transfers/905562/authorisations
/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016620"
    },
    "scaRedirect": {
      "href": "http://ib.xs2a/ib/site/psd2/login?transactionIdsString\u003d905562"
    }
  },
  "scaStatus": "scaMethodSelected"
}
```

An update has been successful, it returns JSON property "scaStatus" with value "scaMethodSelected". The "scaRedirect" property provides an URL to an ASPSP's authorization interface. The "redirect_uri" parameter must be appended to this URL in order the ASPSP could redirect PSU back to TPP interface after PSU finishes authorization. Mandatory parameter "redirect_uri_fail" must be appended in order the ASPSP could redirect PSU to TPP interface after authorization fail. A complete URL with the appended parameters could look like this:

```
http://ib.xs2a/ib/site/psd2/login?transactionIdsString=u003d905562&redirect_uri=http://tpp.com/users
/l&redirect_uri_fail=http://tpp.com/users/l?fail=true
```

Now, PSU must be redirected to that URL to start a payment authorization. If confirmation is cancelled in the Internet bank page, the authorization status becomes "failed". A new authorization resource must be created to complete the authorization.

SEPA payment authorization status

Once PSU has finished the authorization process, TPP can check the payment authorization status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/payments/sepa-credit-transfers/{paymentId}/authorisations/{authorizationId}			
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/{version}/payments/sepa-credit-transfers/905562/authorisations
/UQMNRIIBFPAZHY98ITNW4Q57T5UJMI0000016620
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 finalised.

SEPA payment status

Now, the payment status can be verified to check if the payment got a final status. The following endpoint must be invoked:

```
GET https://api.xs2a/{version}/payments/sepa-credit-transfers/{paymentId}/status
```

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/{version}/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": "ACSC",
  "fundsAvailable": false
}
```

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

Example of SEPA Instant payment

Create a SEPA Instant payment

A payment must be created with its details. The following endpoint must be invoked to create a SEPA Instant payment:

POST https://api.xs2a/{version}/payments/instant-sepa-credit-transfers			
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 <i>Developer portal</i> .			

Create a SEPA Instant payment.

HTTP Request
<pre>POST https://api.xs2a/{version}/payments/instant-sepa-credit-transfers Authorization:Bearer _ACCESS_TOKEN Content-Type:application/json X-Request-ID:ed70e51a-d977-4f06-9de8-331a0eac12d3 { "requestedExecutionDate": "2019-10-20", "endToEndIdentification": "12346", "instructedAmount": { "currency": "EUR", "amount": "123.50" }, "debtorAccount": { "iban": "LT044010000100439350" }, "creditorName": "PSD2 Demo Creditor", "creditorAccount": { "iban": "LT377300012345678901" }, "remittanceInformationUnstructured": "PSD2 Reason of payment" }</pre>

HTTP Response
<pre>Status: 201 Transfer-Encoding:chunked Connection:keep-alive Date:Mon, 09 Sep 2019 07:38:03 GMT Content-Type:application/json { "transactionStatus": "ACTC", "paymentId": "905563", "transactionFees": { "currency": "EUR", "amount": "0" }, "transactionFeeIndicator": false }</pre>

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

SEPA Instant payment verifications

The following verifications will be performed on SEPA Instant payment creation:

- parameter `requestedExecutionDate` must be equal to the payment service call date;
- the value of parameter `amount` is checked against the maximum allowed SEPA Instant payment amount;
- the creditor bank must be an active participant of SEPA INST.

SEPA Instant 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 <code>https://api.xs2a/{version}/payments/instant-sepa-credit-transfers/{paymentId}/authorisations</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.
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 <code>https://api.xs2a/{version}/payments/instant-sepa-credit-transfers/905563/authorisations</code> Authorization: Bearer <code>_ACCESS_TOKEN</code> X-Request-ID: <code>776f9342-32ab-406c-b966-0c0edbfd40f1</code>

HTTP Response
Status: 201 Transfer-Encoding: chunked Connection: keep-alive Date: Mon, 09 Sep 2019 07:38:09 GMT Content-Type: application/json <pre>{ "authorisationId": "UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016622", "scaMethods": [{ "name": "SmartID" }, { "name": "MobileID" }, { "name": "Redirect" }], "_links": { "scaStatus": { "href": "/{version}/payments/instant-sepa-credit-transfers/905563/authorisations/UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016622" }, "selectAuthenticationMethod": { "href": "/{version}/payments/instant-sepa-credit-transfers/905563/authorisations/UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016622" } } }</pre>

The authorization resource has been 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/{version}/payments/instant-sepa-credit-transfers/{paymentId}/authorisations/{authorisationId}			
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 <i>Developer portal</i> .			

Choose an SCA Redirect method. This method will allow redirecting PSU to its ASPSP where he will be able to authorize the payment with its preferred SCA.

HTTP Request
<pre>PUT https://api.xs2a/{version}/payments/instant-sepa-credit-transfers/905563/authorisations /UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016622 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" }</pre>
HTTP Response
<pre>Status: 200 Transfer-Encoding:chunked Connection:keep-alive Date:Mon, 09 Sep 2019 07:38:15 GMT Content-Type:application/json { "_links": { "scaStatus": { "href": "{version}/payments/instant-sepa-credit-transfers/905563/authorisations /UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016622" }, "scaRedirect": { "href": "http://ib.xs2a/ib/site/psd2/login?transactionIdsString\u003d905563" } }, "scaStatus": "scaMethodSelected" }</pre>

An update has been successful, it returns JSON property "scaStatus" with value "scaMethodSelected". The "scaRedirect" property provides an URL to an ASPSP's authorization interface. The "redirect_uri" parameter must be appended to this URL in order the ASPSP could redirect PSU back to TPP interface after PSU finishes authorization. Mandatory parameter "redirect_uri_fail" must be appended in order the ASPSP could redirect PSU to TPP interface after authorization fail. A complete URL with the appended parameters could look like this:

```
http://ib.xs2a/ib/site/psd2/login?transactionIdsString=u003d905563&redirect_uri=http://tpp.com/users
/l&redirect_uri_fail=http://tpp.com/users/l?fail=true
```

Now, PSU must be redirected to that URL to start a payment authorization. If confirmation is cancelled in the Internet bank page, the authorization status becomes "failed". A new authorization resource must be created to complete the authorization.

SEPA Instant payment authorization status

Once PSU has finished an authorization process, TPP can check the payment authorization status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/payments/instant-sepa-credit-transfers/{paymentId}/authorisations/{authorizationId}			
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/{version}/payments/instant-sepa-credit-transfers/905563/authorisations/UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016622 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 finalised.

SEPA Instant payment status

Now, the payment status can be verified to check if the payment got a final status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/payments/instant-sepa-credit-transfers/{paymentId}/status			
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/{version}/payments/instant-sepa-credit-transfers/905563/status Authorization:Bearer _ACCESS_TOKEN X-Request-ID:67cebfa0-c775-4079-b38d-8e574555f6d5

If the "transactionStatus" property value is "ACCC", then the SEPA Instant payment is completed.

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 "ACCC", then the SEPA Instant payment is completed.

If a SEPA Instant payment could not be completed because of insufficient funds, the "transactionStatus" property value is "RJCT" and the error message is returned.

HTTP Response

```
Status: 200
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 29 Sep 2024 11:50:30 GMT
Content-Type: application/json
```

```
{
  "transactionStatus": "RJCT",
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "FUNDS_NOT_AVAILABLE",
      "text": "The payment could not be completed because of insufficient funds."
    }
  ]
}
```

If a SEPA Instant payment is not possible, the "transactionStatus" property value is "RJCT" and the error message is returned.

HTTP Response

```
Status: 200
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 29 Sep 2024 11:55:40 GMT
Content-Type: application/json
```

```
{
  "transactionStatus": "RJCT",
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "SEPA_INSTANT_NOT_POSSIBLE",
      "text": "We recommend initiating a SEPA credit transfer."
    }
  ]
}
```

Example of International payment

Create an International payment

A payment must be created with its details. The following endpoint must be invoked to create an International payment:

POST
https://api.xs2a/{version}/payments/cross-border-credit-transfers

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 *Developer portal*.

Create an International payment.

HTTP Request

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

```
{
  "requestedExecutionDate": "2019-10-20",
  "endToEndIdentification": "12345",
  "debtorAccount": {
    "iban": "LT044010000100439350"
  },
  "creditorName": "PSD2 Demo Creditor",
  "creditorAccount": {
    "iban": "LV377300012345678901"
  },
  "instructedAmount": {
    "amount": 123.50,
    "currency": "EUR"
  },
  "remittanceInformationUnstructured": "PSD2 Reason of payment",
  "priority": "HIGH",
  "chargeBearer": "SHAR",
  "benBankName": "benBankName",
  "benBankAddress": {
    "street": "street",
    "buildingNumber": "buildingNumber",
    "townName": "townName",
    "postCode": "postCode",
    "country": "LV"
  },
  "benBankCodeType": "SWIFT",
  "benBankCodeSwift": "benBankCodeSwift",
  "documentNumber": "documentNumber",
  "sendDebtorAddress": "TRUE",
  "sendDebtorCode": "TRUE",
  "creditorAddress": {
    "street": "street",
    "buildingNumber": "buildingNumber",
    "townName": "townName",
    "postCode": "postCode",
    "country": "LV"
  }
}
```

HTTP Response

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

```
{
  "transactionStatus": "ACTC",
  "paymentId": "905564",
  "transactionFees": {
    "currency": "EUR",
    "amount": "0"
  },
  "transactionFeeIndicator": false
}
```

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

International 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/{version}/payments/cross-border-credit-transfers/{paymentId}/authorisations
```

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/{version}/payments/cross-border-credit-transfers/905564/authorisations
Authorization: Bearer _ACCESS_TOKEN
X-Request-ID: 776f9342-32ab-406c-b966-0c0edbdf40f1
```


HTTP Response

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

{
  "authorisationId": "UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016621",
  "scaMethods": [
    {
      "name": "SmartID"
    },
    {
      "name": "MobileID"
    },
    {
      "name": "Redirect"
    }
  ],
  "_links": {
    "scaStatus": {
      "href": "/{version}/payments/cross-border-credit-transfers/905564/authorisations/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016621"
    },
    "selectAuthenticationMethod": {
      "href": "/{version}/payments/cross-border-credit-transfers/905564/authorisations/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016621"
    }
  }
}
```

The authorization resource has been 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, it is required to update the created authorization resource. The following endpoint must be invoked:

PUT https://api.xs2a/{version}/payments/cross-border-credit-transfers/{paymentId}/authorisations/{authorisationId}			
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 <i>Developer portal</i> .			

Choose an SCA Redirect method. This method will allow redirecting PSU to its ASPSP where he will be able to authorize the payment with its preferred SCA.

HTTP Request

```
PUT https://api.xs2a/{version}/payments/cross-border-credit-transfers/905564/authorisations
/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016621
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": "{version}/payments/cross-border-credit-transfers/905564/authorisations
/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016621"
    },
    "scaRedirect": {
      "href": "http://ib.xs2a/ib/site/psd2/login?transactionIdsString\u003d905564"
    }
  },
  "scaStatus": "scaMethodSelected"
}
```

An update has been successful, it returns JSON property "scaStatus" with value "scaMethodSelected". The "scaRedirect" property provides an URL to an ASPSP's authorization interface. The "redirect_uri" parameter must be appended to this URL in order the ASPSP could redirect PSU back to TPP interface after PSU finishes authorization. Mandatory parameter "redirect_uri_fail" must be appended in order the ASPSP could redirect PSU to TPP interface after authorization fail. A complete URL with the appended parameters could look like this:

```
http://ib.xs2a/ib/site/psd2/login?transactionIdsString=u003d905564&redirect_uri=http://tpp.com/users/1&redirect_uri_fail=http://tpp.com/users/1?fail=true
```

Now, PSU must be redirected to that URL to start a payment authorization. If confirmation is cancelled in the Internet bank page, the authorization status becomes "failed". A new authorization resource must be created to complete the authorization.

International payment authorization status

Once PSU has finished the authorization process, TPP can check the payment authorization status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/payments/cross-border-credit-transfers/{paymentId}/authorisations/{authorizationId}			
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/{version}/payments/cross-border-credit-transfers/905564/authorisations
/UQMNRIIBFPAZHY98ITNW4Q57T5UJMI0000016621
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 finalised.

International payment status

Now, the payment status can be verified to check if the payment got a final status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/payments/cross-border-credit-transfers/{paymentId}/status			
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/{version}/payments/cross-border-credit-transfers/905564/status
Authorization:Bearer _ACCESS_TOKEN
X-Request-ID:67cebfaf0-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": "ACSC",
  "fundsAvailable": false
}
```

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

Example of Periodic payment

Create a Periodic payment

A payment must be created with its details. Periodic payment with daily periodicity can be created after one day from current date. Periodic payment with Weekly, Monthly or Annually periodicity can be created after three days from current date. The following endpoint must be invoked to create a Periodic payment:

POST https://api.xs2a/{version}/periodic-payments/sepa-credit-transfers			
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 <i>Developer portal</i> .			

Create a Periodic payment.

HTTP Request
<pre>POST https://api.xs2a/{version}/periodic-payments/sepa-credit-transfers Authorization:Bearer _ACCESS_TOKEN Content-Type:application/json X-Request-ID:ed70e51a-d977-4f06-9de8-331a0eac12d3 { "endToEndIdentification": "12345", "debtorAccount": { "iban": "LT044010000100439350" }, "instructedAmount": { "amount": 123.50, "currency": "EUR" }, "creditorAccount": { "iban": "LT377300012345678901" }, "creditorAgent": "creditorAgent", "creditorName": "creditorName", "creditorAddress": { "street": "street", "buildingNumber": "buildingNumber", "city": "city", "postalCode": "postalCode", "country": "LT" }, "remittanceInformationUnstructured": "PSD2 Reason of payment", "startDate": "2019-10-21", "endDate": "2019-12-30", "executionRule": "following", "frequency": "Daily" }</pre>

HTTP Response

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

```
{
  "transactionStatus": "ACTC",
  "paymentId": "905565",
  "transactionFees": {
    "currency": "EUR",
    "amount": "0"
  },
  "transactionFeeIndicator": false
}
```

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

Periodic 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/{version}/periodic-payments/sepa-credit-transfers/{paymentId}/authorisations
```

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/{version}/periodic-payments/sepa-credit-transfers/905565/authorisations
Authorization: Bearer _ACCESS_TOKEN
X-Request-ID: 776f9342-32ab-406c-b966-0c0edbdf40f1
```

HTTP Response

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

{
  "authorisationId": "UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016623",
  "scaMethods": [
    {
      "name": "SmartID"
    },
    {
      "name": "MobileID"
    },
    {
      "name": "Redirect"
    }
  ],
  "_links": {
    "scaStatus": {
      "href": "/{version}/periodic-payments/sepa-credit-transfers/905565/authorisations/UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016623"
    },
    "selectAuthenticationMethod": {
      "href": "/{version}/periodic-payments/sepa-credit-transfers/905565/authorisations/UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016623"
    }
  }
}
```

The authorization resource has been 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, it is required to update the created authorization resource. The following endpoint must be invoked:

PUT https://api.xs2a/{version}/periodic-payments/sepa-credit-transfers/{paymentId}/authorisations/{authorisationId}			
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 <i>Developer portal</i> .			

Choose an SCA Redirect method. This method will allow redirecting PSU to its ASPSP, where he will be able to authorize the payment with its preferred SCA.

HTTP Request

```
PUT https://api.xs2a/{version}/periodic-payments/sepa-credit-transfers/905565/authorisations
/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016623
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": "{version}/periodic-payments/sepa-credit-transfers/905565/authorisations
/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016623"
    },
    "scaRedirect": {
      "href": "http://ib.xs2a/ib/site/psd2/login?transactionIdsString\u003d905565"
    }
  },
  "scaStatus": "scaMethodSelected"
}
```

An update has been successful, it returns JSON property "scaStatus" with value "scaMethodSelected". The "scaRedirect" property provides an URL to an ASPSP's authorization interface. The "redirect_uri" parameter must be appended to this URL in order the ASPSP could redirect PSU back to TPP interface after PSU finishes authorization. Mandatory parameter "redirect_uri_fail" must be appended in order the ASPSP could redirect PSU to TPP interface after authorization fail. A complete URL with the appended parameters could look like this:

```
http://ib.xs2a/ib/site/psd2/login?transactionIdsString=u003d905565&redirect_uri=http://tpp.
com/users/1&redirect_uri_fail=http://tpp.com/users/1?fail=true
```

Now, PSU must be redirected to that URL to start a payment authorization. If confirmation is cancelled in the Internet bank page, the authorization status becomes "failed". A new authorization resource must be created to complete the authorization.

Periodic payment authorization status

Once PSU has finished the authorization process, TPP can check the payment authorization status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/periodic-payments/sepa-credit-transfers/{paymentId}/authorisations/{authorizationId}			
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/{version}/periodic-payments/sepa-credit-transfers/905565/authorisations
/UQMNRIIBFPAZHY98ITNW4Q57T5UJMI0000016623
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 finalised.

Periodic payment status

Now, the payment status can be verified to check if the payment got a final status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/periodic-payments/sepa-credit-transfers/{paymentId}/status			
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/{version}/periodic-payments/sepa-credit-transfers/905565/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": "ACSC",
  "fundsAvailable": false
}
```

Periodic payment status "transactionStatus" property value is "ACCP" until first transaction is made. Earliest date for first transaction is second day after periodic payment creation. If the "transactionStatus" property value is "ACSC", then the Periodic payment contract is successfully created.

Example of Bulk payment

Only SEPA payments or only International payments can be created inside one Bulk payment. They cannot be mixed.

Create a Bulk SEPA payment

A payment must be created with its details. The following endpoint must be invoked to create a Bulk payment:

POST https://api.xs2a/{version}/bulk-payments/sepa-credit-transfers			
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 <i>Developer portal</i> .			

Create a Bulk payment.

HTTP Request
<pre>POST https://api.xs2a/{version}/bulk-payments/sepa-credit-transfers Authorization:Bearer _ACCESS_TOKEN Content-Type:application/json X-Request-ID:ed70e51a-d977-4f06-9de8-331a0eac12d3 { "debtorAccount": { "iban": "LT044010000100439350" }, "requestedExecutionDate": "2019-10-20", "payments": [{ "creditorAccount": { "iban": "LT377300012345678901" }, "creditorName": "PSD2 Demo Creditor", "instructedAmount": { "amount": "123.50", "currency": "EUR" }, "remittanceInformationUnstructured": "PSD2 Reason of payment" }] }</pre>

HTTP Response

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

```
{
  "transactionStatus": "PDNG",
  "paymentId": "905566",
  "transactionFees": {
    "currency": "EUR",
    "amount": "0"
  },
  "transactionFeeIndicator": false
}
```

Bulk 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/{version}/bulk-payments/sepa-credit-transfers/{paymentId}/authorisations
```

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/{version}/bulk-payments/sepa-credit-transfers/905566/authorisations
Authorization: Bearer _ACCESS_TOKEN
X-Request-ID: 776f9342-32ab-406c-b966-0c0edbdf40f1
```

HTTP Response

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

```
{
  "authorisationId": "UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016624",
  "scaMethods": [
    {
      "name": "SmartID"
    },
    {
      "name": "MobileID"
    },
    {
      "name": "Redirect"
    }
  ],
  "_links": {
    "scaStatus": {
      "href": "/{version}/bulk-payments/sepa-credit-transfers/905566/authorisations/UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016624"
    },
    "selectAuthenticationMethod": {
      "href": "/{version}/bulk-payments/sepa-credit-transfers/905566/authorisations/UQMNRI BFP AZHY98ITNW4Q57T5UJJMI0000016624"
    }
  }
}
```

The authorization resource has been 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, it is required to update the created authorization resource. The following endpoint must be invoked:

PUT https://api.xs2a/{version}/bulk-payments/sepa-credit-transfers/{paymentId}/authorisations/{authorisationId}			
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 <i>Developer portal</i> .			

Choose an SCA Redirect method. This method will allow redirecting PSU to its ASPSP where he will be able to authorize the payment with its preferred SCA.

HTTP Request

```
PUT https://api.xs2a/{version}/bulk-payments/sepa-credit-transfers/905565/authorisations
/UQMNRI BFPZHY98ITNW4Q57T5UJ JMI0000016624
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"
}
```

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

```
{
  "_links": {
    "scaStatus": {
      "href": "{version}/bulk-payments/sepa-credit-transfers/905566/authorisations
/UQMNRI BFPZHY98ITNW4Q57T5UJ JMI0000016624"
    },
    "scaRedirect": {
      "href": "http://ib.xs2a/ib/site/psd2/login?transactionIdsString\u003d905566"
    }
  },
  "scaStatus": "scaMethodSelected"
}
```

An update has been successful, it returns JSON property "scaStatus" with value "scaMethodSelected". The "scaRedirect" property provides an URL to an ASPSP's authorization interface. The "redirect_uri" parameter must be appended to this URL in order the ASPSP could redirect PSU back to TPP interface after PSU finishes authorization. Mandatory parameter "redirect_uri_fail" must be appended in order the ASPSP could redirect PSU to TPP interface after authorization fail. A complete URL with the appended parameters could look like this:

```
http://ib.xs2a/ib/site/psd2/login?transactionIdsString=u003d905566&redirect_uri=http://tpp.
com/users/1&redirect_uri_fail=http://tpp.com/users/1?fail=true
```

Now, PSU must be redirected to that URL to start a payment authorization. If confirmation is cancelled in the Internet bank page, the authorization status becomes "failed". A new authorization resource must be created to complete the authorization.

Bulk payment authorization status

Once PSU has finished the authorization process, TPP can check the payment authorization status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/bulk-payments/sepa-credit-transfers/{paymentId}/authorisations/{authorizationId}			
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/{version}/bulk-payments/sepa-credit-transfers/905565/authorisations
/UQMNRI BFP AZHY98ITNW4Q57T5UJ JMI0000016624
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 finalised.

Bulk payment status

Now, the payment status can be verified to check if the payment got a final status. The following endpoint must be invoked:

GET https://api.xs2a/{version}/bulk-payments/sepa-credit-transfers/{paymentId}/status			
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/{version}/bulk-payments/sepa-credit-transfers/905565/status
Authorization:Bearer _ACCESS_TOKEN
X-Request-ID:67cebf a0-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": "ACSC",
  "fundsAvailable": false
}
```

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

Bulk International payment flow is the same as Bulk SEPA payment flow, only {payment-product} part in endpoints "sepa-credit-transfers" should be replaced with "cross-border-credit-transfers".

Payment cancelation

To cancel created payment, the following endpoint must be invoked:

DELETE https://api.xs2a/{version}/{payment-service}/{payment-product}/{paymentId}			
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".

SEPA Instant payments cannot be cancelled.

If payment can be cancelled, "CANC" status will be returned.

If payment cannot be cancelled, an error will be returned.

In order to cancel Periodic payment, authorization by PSU with SCA is needed.

Example of SEPA payment cancellation

Cancel a SEPA payment:

HTTP Request
DELETE https://api.xs2a/{version}/payments/sepa-credit-transfers/905562 Authorization:Bearer _ACCESS_TOKEN Content-Type:application/json X-Request-ID:ed70e51a-d977-4f06-9de8-331a0eac12d3
HTTP Response
Status: 200 Transfer-Encoding:chunked Connection:keep-alive Date:Mon, 09 Sep 2019 07:38:03 GMT Content-Type:application/json { "transactionStatus": "CANC" }

The payment has been successfully cancelled.

Example of Periodic payment cancellation

Cancel a Periodic payment:

HTTP Request
DELETE https://api.xs2a/{version}/periodic-payments/sepa-credit-transfers/905565 Authorization:Bearer _ACCESS_TOKEN Content-Type:application/json X-Request-ID:ed70e51a-d977-4f06-9de8-331a0eac12d3

HTTP Response

```
Status: 202
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2019 07:38:03 GMT
Content-Type: application/json

{
  "transactionStatus": "ACCP",
  "_links": {
    "startAuthorisation": {
      "href": "/{version}/periodic-payments/sepa-credit-transfers/905565/cancellation-authorisations"
    }
  }
}
```

If cancellation is possible, a link to create authorization resource is returned. The cancellation authorization flow with authorization resource is the same as in payments flow.

Future dated payments

Future-dated payments can be created.

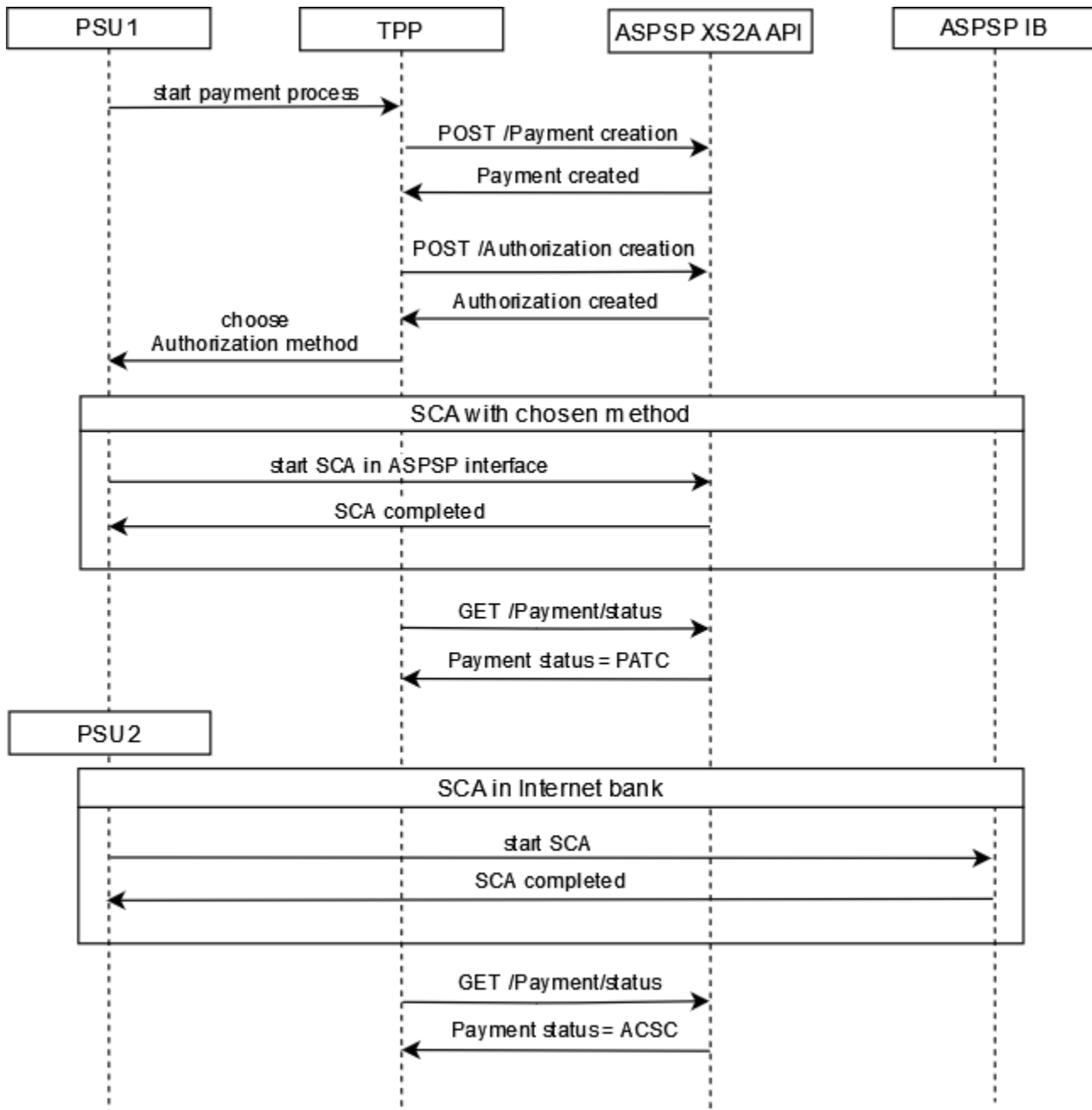
To create a payment in future, in a SEPA payment service, JSON parameter "requestedExecutionDate" should be provided with some date in future.

Parameter "requestedExecutionDate" is checked against the maximum allowed days in future for future-dated payments.

Countersigning

If countersigning is needed to completely authorize a payment, after first SCA, payment status becomes "PATIC". The following authorizations are performed in Internet bank. TPP should check payment status until status becomes final: "ACCC" for SEPA Instant payments, "ACSC" for other payments.

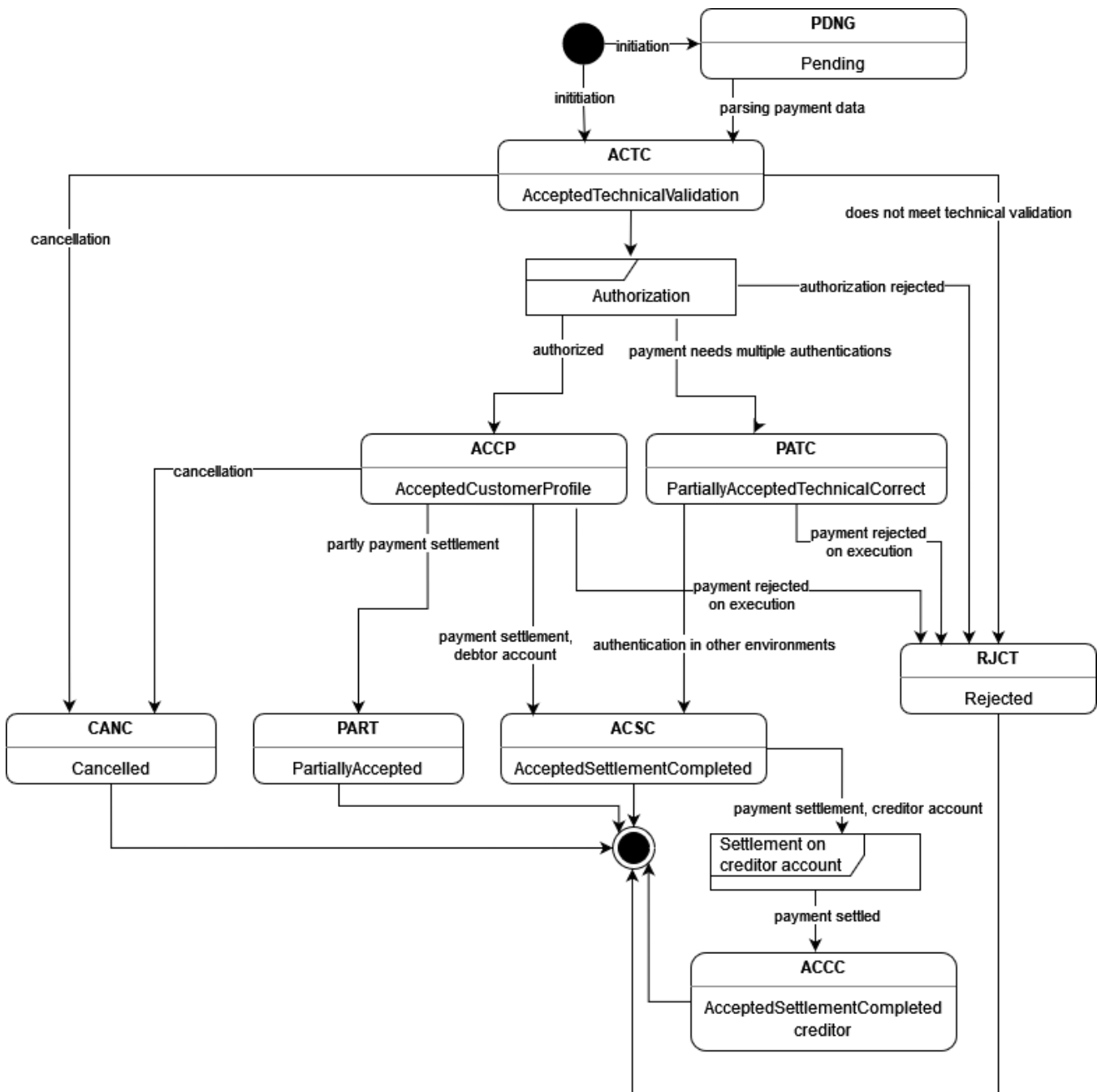
Here is simplified countersigning flow:



Debtor account

In all payments, except BULK payments, a Debtor account could not be provided in a payment creation service call. In this case only Redirect SCA approach is available. Debtor account is selected by customer in the Internet bank.

Payment status flow



Payment states description:

- PDNG – the payment has been received successfully (status available only for Bulk payments);
- ACTC – the payment has been registered successfully;
- ACCP – the payment has been confirmed by PSU with SCA, the money has not been booked from the debtor account yet;
- PATC – the payment has been authorized partially, payments needs multiple authentications;
- PART – a number of transactions have been accepted, whereas another number of transactions have not yet achieved 'accepted' status (status available only for Bulk payments);
- ACSC – the money has been booked from the debtor account;
- ACCC – the payment has been completed on the creditor's account (status available only for SEPA Instant payments);
- RJCT – the payment routine has failed (on any state).

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/{version}/funds-confirmations			
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 <i>Developer portal</i> .			

Check if the funds are available.

HTTP Request
<pre>POST https://api.xs2a/{version}/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" } }</pre>

HTTP Response
<pre>Status: 200 Transfer-Encoding:chunked Connection:keep-alive Date:Mon, 09 Sep 2019 07:40:52 GMT Content-Type:application/json { "fundsAvailable": false }</pre>

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

Health service

The following endpoint can be invoked to check XS2A service health status.

GET https://api.xs2a/{version}/health-check/ping			
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=SANDBOX for Sandbox environment, please ask for PROD environment)
--------	--------	-----------	---

Example

HTTP Request

```
GET https://api.xs2a/{version}/health-check/ping?branch=SANDBOX
X-Request-ID:f63daf5f-27ae-4992-9eda-940dd2a1dae0
```

HTTP Response

```
Status: 200
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2023 07:34:56 GMT
Content-Type: application/json
```

```
{
  "status": "ok"
}
```

Errors

HTTP Response for error when QWAC certificate is missing

```
Status: 403 Forbidden
Connection: keep-alive
Date: Mon, 09 Sep 2023 08:34:56 GMT
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /v2/oauth/authorization/links
on this server.</p>
</body></html>
```

HTTP Response for error when Healthcheck frequency exceeded

```
Status: 429
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2023 07:35:56 GMT
Content-Type: application/json
```

```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "ACCESS_EXCEEDED",
      "text": "Healthcheck frequency exceeded!"
    }
  ]
}
```

HTTP Response for error when XS2A services are not available

```
Status: 503
Transfer-Encoding: chunked
Connection: keep-alive
Date: Mon, 09 Sep 2023 07:36:56 GMT
Content-Type: application/json

{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "TEMPORARILY_UNAVAILABLE",
      "text": "XS2A services are temporarily not available"
    }
  ]
}
```

API versioning

Only a major version is used in API URI endpoints.

A new API major version is created when breaking changes are introduced. Breaking changes are those that are not backward compatible. For example:

- Removing endpoints, parameters
- Adding new mandatory parameters
- Changing datatypes or names of parameters

A new major version of the API is not created for non-breaking changes. For example:

- Adding new endpoints
- Adding new optional input parameters
- Adding new parameters to existing endpoints
- Changing the order of the parameters in the response payloads
- Adding new error messages

A non-breaking change can be introduced without prior noticing, because it does not affect current operations. Information about non-breaking changes is provided in Sandbox and Swagger.

Information about breaking changes is provided in Sandbox and Swagger 3 months before Production release date. A previous version is removed from Sandbox and Production after 1 month of Production release. A new and previous versions of API are available in Sandbox and Production in this period.

Errors

The errors are implemented as HTTP responses with a status code, and in certain cases – with 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"
  } ]
}
```

HTTP statuses

HTTP status	Description
-------------	-------------

200 OK	Request has succeeded.
201 Created	New resource has been created.
204 No Content	DELETE request has succeeded where a resource was successfully deleted.
400 Bad Request	The request cannot be fulfilled due to bad syntax in payload.
401 Unauthorized	Authentication is possible but has failed or not yet been provided.
403 Forbidden	The request was valid, but the server is refusing action. The user might not have the necessary permissions for a resource.
404 Not Found	The requested resource could not be found but may be available again in the future.
405 Method Not Allowed	The requested resource is valid and exists, but an unacceptable HTTP method has been used.
409 Conflict	Indicates that the request could not be processed because of conflict in the request, such as an edit conflict.
429 Too Many Requests	The TPP has sent too many requests in a given amount of time.
503 Service Unavailable	Request failed due to unexpected error or maintenance on ASPSP side.

Common error messages

Code	Service	Error text	Description	HTTP status
ACCESS_EXCEEDED	AIS	Consent frequency exceeded!	The access on the account has been exceeding the consented multiplicity per day.	429
CERTIFICATE_INVALID	Oauth2 / AIS / PIS	Certificate is invalid.	The contents of the signature/corporate seal certificate are not matching PSD2 general PSD2 or attribute requirements.	401
CERTIFICATE_INVALID	Oauth2 / AIS / PIS	TPP does not have a PISP role.	TPP does not have an authorized role to perform the action.	401
CONSENT_EXPIRED	AIS	Consent expired!	The consent was created by this TPP but has expired and needs to be renewed.	401
CONSENT_INVALID	AIS	Consent not valid!	The consent definition is not complete or invalid.	401
CONSENT_INVALID	AIS	Invalid consent Id!	The consent was created by this TPP but is not valid for the addressed service/resource.	401
PSU_CREDENTIALS_INVALID	AIS / PIS/ PIIS	The PSU cannot be matched by the addressed ASPSP or is blocked.	The user cannot be matched by the addressed ASPSP or is blocked by ASPSP.	401
FORMAT_ERROR	AIS / PIS	{property} must match "[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}"	The specified property must match the pattern "[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}"	400
FORMAT_ERROR	PIS	{property} must not be null.	Error is produced when property format is not properly given.	400
FORMAT_ERROR	AIS	frequencyPerDay must be less than or equal to 4.	The value of frequencyPerDay must be less than or equal to 4 for the AIS operation.	400
FORMAT_ERROR	Oauth2 / AIS / PIS	Invalid request body format.	Format of certain request fields are not matching the XS2A requirements. An explicit path to the corresponding field might be added in the return message.	400
INVALID_GRANT	Oauth2	Code not valid.	The provided code is not valid for authorization.	400
INVALID_GRANT	Oauth2	Maximum allowed refresh token reuse exceeded.	The maximum allowed reuse of the refresh token has been exceeded.	400
INVALID_GRANT	Oauth2	Token is not active.	The provided token is not active.	400
PAYMENT_FAILED	PIS	Date in Future.	The specified date is in the future and is not allowed for the payment operation.	400
PAYMENT_FAILED	PIS	Date in the past.	The specified date is in the past and is not allowed for the payment operation.	400

PAYMENT_FAILED	PIS	Date must be at least 3 days from now.	The specified date must be at least 3 days in the future for the payment operation.	400
PAYMENT_FAILED	PIS	Date must be in future.	The specified date must be in the future for the payment operation.	400
PAYMENT_FAILED	PIS	Entered category purpose code not supported.	The specified category purpose code is not supported for the payment operation.	400
PAYMENT_FAILED	PIS	Entered purpose code not supported.	The specified purpose code is not supported for the payment operation.	400
PAYMENT_FAILED	PIS	Gavėjo bankas nerastas.	The creditor bank could not be found.	400
PAYMENT_FAILED	PIS	Incorrect schedule of regular payments.	The schedule of regular payments is incorrect.	400
PAYMENT_FAILED	PIS	Internal error. Error number {id}.	The error comes from payment processing.	400
PAYMENT_FAILED	PIS	You do not have rights to create operations.	The user does not have the rights to create operations.	400
PAYMENT_FAILED	PIS	Įvestas nepalaikomas paskirties kodas.	The entered purpose code is not supported.	400
PAYMENT_FAILED	PIS	Nenurodyta mokėjimo paskirtis.	The payment purpose is not specified.	400
PAYMENT_FAILED	PIS	Nenurodytas gavėjo vardas.	The creditor name is not specified.	400
PAYMENT_FAILED	PIS	Nėra darbo grupės narių, kurie galėtų pasirašyti operaciją su nurodyta suma, arba jie neturi teisės į su operacija susijusią sąskaitą ar klientą.	There are no users who can confirm the transaction with the specified amount, or they do not have rights to the account or client associated with the transaction.	400
PAYMENT_FAILED	PIS	Neteisingas automatinių mokėjimų grafikas.	The schedule of periodic payment is invalid.	400
PAYMENT_FAILED	PIS	Bank code does not correspond to creditor account.	The provided bank code does not correspond to creditor account.	400
PAYMENT_FAILED	PIS	Payee bank not defined as SEPA member.	The payee bank is not defined as a SEPA member.	400
PAYMENT_FAILED	PIS	SEPA payment data does not meet requirements.	The SEPA payment data does not meet the requirements.	400
PAYMENT_FAILED	PIS	SEPA Instant payment data does not meet requirements.	The SEPA Instant payment data does not meet the requirements.	400
PAYMENT_FAILED	PIS	Ultimate beneficiary: Invalid identification type.	The identification type of the ultimate beneficiary is invalid.	400
PAYMENT_FAILED	PIS	Vartotojas neturi teisių kurti operacijų.	The user does not have rights to create transactions.	400
PAYMENT_FAILED	PIS	XML failas neatitinka XSD schemas.	The XML file does not conform to the XSD schema.	400
PAYMENT_FAILED	PIS	XML file is invalid against XSD schema.	The XML file is invalid against the XSD schema.	400
PAYMENT_FAILED	PIS	Neteisingas LEI kodas	The LEI code is invalid.	400
PARAMETER_NOT_CONSISTENT	PIS	Currency is missing!	The currency parameter is missing in the request.	400
PARAMETER_NOT_CONSISTENT	AIS	Invalid consent date until!	The specified consent date until is invalid.	400
PARAMETER_NOT_CONSISTENT	AIS	Invalid period dates specified!	Invalid period dates specified for account's transactions.	400
PARAMETER_NOT_SUPPORTED	Oauth2 / AIS / PIS	Unknown authentication method: {property}.	The parameter is not supported by the API provider.	400
RESOURCE_EXPIRED	PIS	Draft data is already expired.	The addressed resource is not addressable by this request.	403
RESOURCE_UNKNOWN	AIS / PIS	Invalid account Id!	The addressed resource is not associated with the PSU or TPP.	404

RESOURCE_UNKNOWN	PIS	Invalid payment Id!	The addressed resource is not associated with the PSU or TPP.	404
SERVICE_INVALID	Oauth2 / AIS / PIS	The addressed service is not valid for the addressed resources or the submitted data.	This service is not reachable for the addressed PSU due to a channel independent blocking by the ASPSP.	405
STATUS_INVALID	AIS / PIS	Authorization for the consent/payment has already been finalised!	The addressed resource does not allow additional authorization.	409
STATUS_INVALID	Oauth2 / AIS / PIS	Invalid authorization parameters!	The provided authorization parameters are invalid.	409
STATUS_INVALID	AIS / PIS	Your phone number is not specified.	The phone number is not specified for the user.	409
STATUS_INVALID	Oauth2 / AIS / PIS	Jūs neturite teisių patvirtinti šios operacijos (negalima keisti šio pranešimo į kitokį, susijusį su darbo grupėmis!).	You do not have the rights to confirm this operation.	409
STATUS_INVALID	PIS	User already confirmed transaction.	The addressed resource does not allow additional authorization.	409
STATUS_INVALID	Oauth2 / AIS / PIS	User does not have account in Smart-ID system.	The user does not have an account in the Smart-ID system.	409
STATUS_INVALID	Oauth2 / AIS / PIS	Vartotojas jau pasirašė operacija.	The addressed resource does not allow additional authorization.	409
TOKEN_INVALID	Oauth2	Could to acquire subject (sub) from access token.	The OAuth2 token is associated to the TPP but is not valid for the addressed service/resource.	401
TOKEN_INVALID	Oauth2	Token invalid.	The OAuth2 token is associated to the TPP but is not valid for the addressed service/resource.	401

XS2A API Changes in Version V2

Changes to the debtorIdentification object structure

The value of the `debtorId` parameter is now accepted in the `id` parameter, which is inside the `debtorIdentification` object. The `debtorId` parameter is eliminated.

The following services are affected:

- POST `/v2/payments/sepa-credit-transfers`
- GET `/v2/payments/sepa-credit-transfers/{paymentId}`
- POST `/v2/payments/instant-sepa-credit-transfers`
- GET `/v2/payments/instant-sepa-credit-transfers/{paymentId}`

New objects and parameters in international payment services

Parameter	Data type	Description
<code>ultimateDebtor</code>	String	Initial payer
<code>ultimateDebtorIdentification</code>	Object	Initial payer data array
<code>ultimateCreditor</code>	String	Ultimate payee
<code>ultimateCreditorIdentification</code>	Object	Ultimate payee data array
<code>purposeCode</code>	String	Purpose code
<code>ctgyPurp</code>	String	Purpose category code

Structure of objects `ultimateDebtorIdentification` and `ultimateCreditorIdentification`:

Parameter	Data type	Length
<code>id</code>	String	35
<code>partyType</code>	String	Enum
<code>type</code>	String	Enum
<code>schemeNameCode</code>	String	4

birthDate	Date	
birthCountry	String	2
birthTown	String	35

The following services are affected:

- POST `/{{version}}/payments/cross-border-credit-transfers`
- GET `/{{version}}/payments/cross-border-credit-transfers/{{paymentId}}`
- POST `/{{version}}/bulk-payments/cross-border-credit-transfers`
- GET `/{{version}}/bulk-payments/cross-border-credit-transfers/{{paymentId}}`

Changes to the address object structure

The structure of the address object changes to:

Parameter	Mandatory	Data type	Length
Street	No	String	70
buildingNumber	No	String	16
postBox	No	String	16
Room	No	String	70
postCode	No	String	16
townName	Yes	String	35
districtName	No	String	35
Country	Yes	String	2

The new address structure is used in object `creditorAddress`, and parameters `benBankAddress` and `corrBankAddress` become the objects of a new structure.

The following services are affected:

- POST `/{{version}}/payments/cross-border-credit-transfers`
- GET `/{{version}}/payments/cross-border-credit-transfers/{{paymentId}}`
- POST `/{{version}}/bulk-payments/cross-border-credit-transfers`
- GET `/{{version}}/bulk-payments/cross-border-credit-transfers/{{paymentId}}`

Changes to the parameter `sendDebtorAddress`

Parameter `sendDebtorAddress` is now mandatory in cross-border payments.

The following services are affected:

- POST `/{{version}}/payments/cross-border-credit-transfers`
- POST `/{{version}}/bulk-payments/cross-border-credit-transfers`

Changes to the error when payment draft is expired

Error is changed for SEPA, SEPA INSTANT, cross-border, and periodic payments in cases when a debtor account is chosen in the Internet bank and a payment draft has expired.

Old error:

HTTP status: **503**

New error:

HTTP status: **403**

```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "RESOURCE_EXPIRED",
      "text": "Draft data is already expired."
    }
  ]
}
```


The following services are affected:

- GET `/ {version} /payments/sepa-credit-transfers/{paymentId}`
- GET `/ {version} /payments/instant-sepa-credit-transfers/{paymentId}`
- GET `/ {version} /payments/cross-border-credit-transfers/{paymentId}`
- GET `/ {version} /periodic-payments/sepa-credit-transfers/{paymentId}`
- GET `/ {version} /payments/sepa-credit-transfers/{paymentId}/status`
- GET `/ {version} /payments/instant-sepa-credit-transfers/{paymentId}/status`
- GET `/ {version} /payments/cross-border-credit-transfers/{paymentId}/status`
- GET `/ {version} /periodic-payments/sepa-credit-transfers/{paymentId}/status`

Changes in payment authorization status after payment cancellation in Internet bank

Payment authorization status after cancellation in Internet bank is now `failed` instead of `scaMethodSelected`.

The following services are affected, according to template:

- GET `/ {version} /{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}`

Changes in AIS consent validity period

The maximum available AIS consent validity period is changed from 90 days to 180 days.

Changes in OAuth2 token validity period

OAuth2 token validity period is changed from 90 days to 180 days.

Changes to the parameter `frequencyPerDay`

Maximum value of parameter `frequencyPerDay` is now 4 instead of unlimited value.

The following service is affected:

- POST `/ {version} /consents`

Correction of AIS services calls counting

AIS services calls are now calculated all together for available count per day according to `frequencyPerDay` value instead of separate count for each service.

Changes to the HTTP query parameter `redirect_uri_fail`

Parameter `redirect_uri_fail` is now mandatory for OAuth2 token obtaining URL, AIS consent authorization URL, payments authorization URL.

Changes to the parameter `validUntil`

If a date in parameter value is later than the maximal available date, date is adjusted to the a maximal available date. Consent object retrieved by the GET Consent Request contains the adjusted date.

The following services are affected:

- POST `/ {version} /consents`
- GET `/ {version} /consents/{consentId}`

XS2A API Changes in Version V3

New service GET `/health-check/ping`

GET `/ {version} /health-check/ping` endpoint can be invoked to check XS2A service health status.

New objects ultimateDebtorAddress and ultimateCreditorAddress in international payment services

Parameter	Data type	Description
ultimateDebtorAddress	Object	Initial payer address.
ultimateCreditorAddress	Object	Ultimate payee address.

The structure of the address object:

Parameter	Mandatory	Data type	Length
Street	No	String	70
buildingNumber	No	String	16
postBox	No	String	16
Room	No	String	70
postCode	No	String	16
townName	Yes	String	35
districtName	No	String	35
Country	Yes	String	2

The following services are affected:

- POST `/v1/payments/cross-border-credit-transfers`
- GET `/v1/payments/cross-border-credit-transfers/{paymentId}`
- POST `/v1/bulk-payments/cross-border-credit-transfers`
- GET `/v1/bulk-payments/cross-border-credit-transfers/{paymentId}`

Optional object tppMessages introduced in payment status services answers

The structure of the address object:

Parameter	Mandatory	Data type
category	Yes	String
code	yes	String
text	No	String

The following services are affected:

- GET `/v1/{payment-service}/{payment-product}/{paymentId}/status`

Optional HTTP parameter PSU-CORPORATE-ID introduced in /accounts-list service

Identification of a Corporate or a Person by Corporate code or Personal code.

Valid representations for `PSU-CORPORATE-ID` in service answer parameter `validation` for particular account:

- "true" - `PSU-CORPORATE-ID` value was provided, the individual or company has rights to the account.
- "false" - `PSU-CORPORATE-ID` value was provided, the individual or company has no rights to the account.
- Empty(NULL) - `PSU-CORPORATE-ID` value was not provided.

The following services are affected:

- GET `/v1/accounts-list`

HTTP parameter X-Request-ID obligation

Error corrected for HTTP parameter `X-Request-ID` requirement. In some cases parameter `X-Request-ID` was not mandatory. Now `X-Request-ID` is obligatory.

Account transactions older than 90 days

Account transactions service for transactions older than 90 days requires authorization by PSU. Please follow steps describes in chapter "Account transactions older than 90 days".

The following services are affected:

- GET `/{version}/accounts/{resourceId}/transactions`
- POST `/{version}/accounts/{resourceId}/transactions`
- GET `/{version}/accounts/{resourceId}/transactions/{statementId}`
- POST `/{version}/accounts/{resourceId}/transactions/{statementId}/authorisations`
- GET `/{version}/accounts/{resourceId}/transactions/{statementId}/authorisations`
- GET `/{version}/accounts/{resourceId}/transactions/{statementId}/authorisations/{authorisationId}`
- PUT `/{version}/accounts/{resourceId}/transactions/{statementId}/authorisations/{authorisationId}`



Important

The detailed descriptions of the objects and parameters are given in the Swagger file.