PSD2 API documentation

Version	Date	Berlin group version	Author
0.9	2021-10-13	1.3.9	LKU

Content

1. Versioning

Version	Date	Comments
0.1	2019-03-13	Initial document
0.2	2019-03-18	Added Xs2A endpoints documentation Added OAuth2 server documentation Added missing code examples
0.3	2019-05-06	Added Oauth2 authorization and token endpoints description and CURL examples Added PIS endpoint's response examples Added testing data documentation
0.4	2019-06-21	Added AIS, PIS authorization endpoint's documentation Added PIS delete endpoint documentation
0.5	2019-06-27	Added AIS redirect and decoupled authorization workflow charts and documentation Added PIS redirect and decoupled authorization workflow charts and documentation
0.6	2019-08-05	Added payment state change diagram and description Added consent model table
0.7	2020-03-10	Added decoupled authorization documentation Updated JSON request/response examples Updated payment and consent decoupled methods documentation Added PSU request context data documentation
0.7.1	2021-03-16	Added missing authorization header description
0.8	2021-05-09	Added available authentication method endpoint Changed decoupled authorization logic Removed update decoupled authorization endpoint Added refresh token endpoint Added remmitance structured support Added invalidate token endpoint Fixed global consent model description and logic Updated payment state schema Added owner name in accounts responses Fixed credit and debtor mixed information Removed absolute URL's from steering links Added consent/payment authorization endpoints descriptions Added pagination and query filter to transaction list endpoint Added error codes description
0.9	2021-06-10	Updated error codes Added user info endpoint Addede extended service: recent beneficiaries

1.1 Glossary

AISP	account information service provider
API	application programming interface
ASPSP	account servicing payment service provider

PSD2	payment service provider
PIISP	card-based payment instrument issue
PISP	payment initiation service provider
PSP	payment services providers
PSU	payment service user
SCA	strong customer authentication
TPP	third party provider

1.2 Technical specification

- Character set - UTF-8
- Transport protocol
  - HTTP version 1.1
  - TLS version 1.2
- Application protocol
  - RESTful with HAL support
- Authorization protocol
  - Oauth2/redirect
  - Decoupled
- Data formats
  - JSON
  - XML
- Data model - ISO 20022

2. Developer's site

For testing and mutual partnership purposes, a developer site has been created. Every market participant now can register through the developer's site registration form and create PSD2 API clients. This later could be used for accessing Xs2a endpoints. Also, there is implemented functionality for generating TPP QWAC, QSEAL certificates for testing purposes. The developer's site can be accessed via https://developers.i-unija.lt link.

2.1 User registration

User registration form can be accessed from the main page via the *Register* tab (Figure 1). During the user registration process, please provide a valid format, existing email address and user password that meet required complexity (min. 6 characters including lowercase, uppercase, and alphanumerical symbols). The Organization/full name field is not mandatory. After successful registration you will be redirected to login page with success message (Figure 1) and an account confirmation email will be sent to your mailbox shortly. Please click confirmation link inside email message to finish user registration process. A confirmation link will be available one hour after generation.

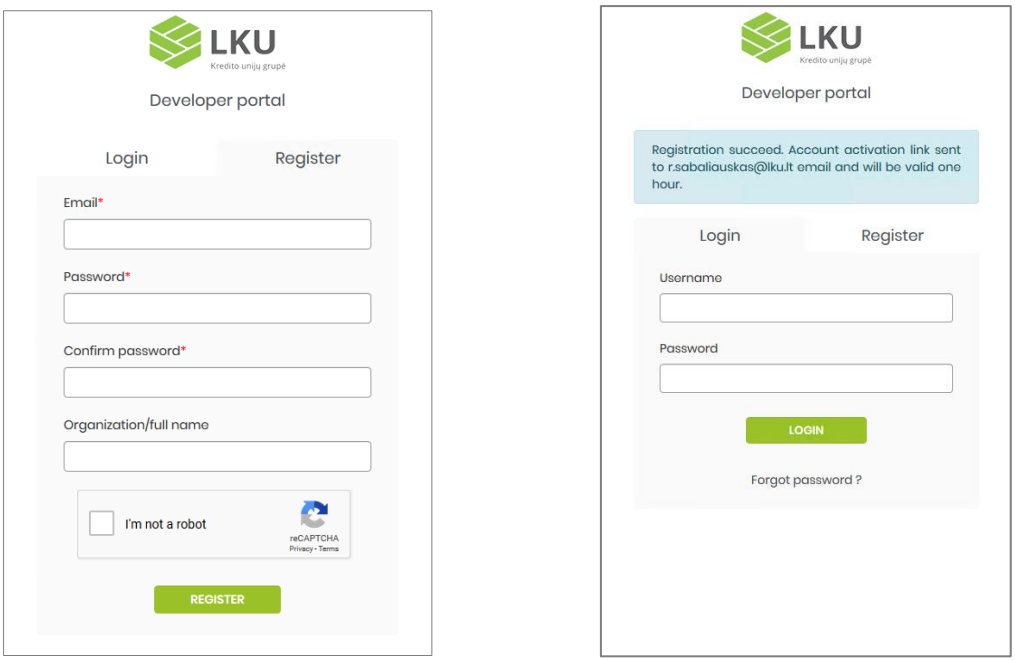

Figure 1. Registration form and success registration message

Troubleshooting guide

Not receiving confirmation link email. If the confirmation has not been received during few minutes period firstly, please check spam folder. Maybe your email server filters classified confirmation email as spam. Otherwise, if the spam folder is empty, please go to the developer 's site login screen and try to login with your credentials. If the account is not active you will get an error message with activation email resend link. Click the *resend activation link* and you will be redirected to resend form where you must enter your email address that was used during registration process (Figure 2).

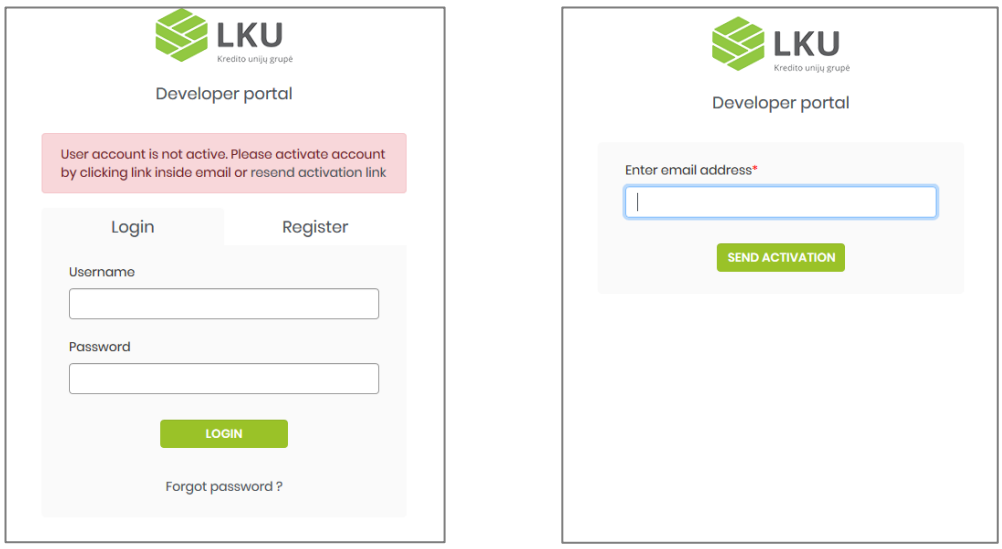

Figure 2. Inactive account error message and activation link resend form

Confirmation link expired. There could be a situation when you forgot for some reasons to click confirmation link inside email then after one-hour link will expire. Clicking an expired activation link will redirect you to the login form with corresponding error message together with activation resend link. Clicking this link will bring you to the activation link resend form from Figure 2.

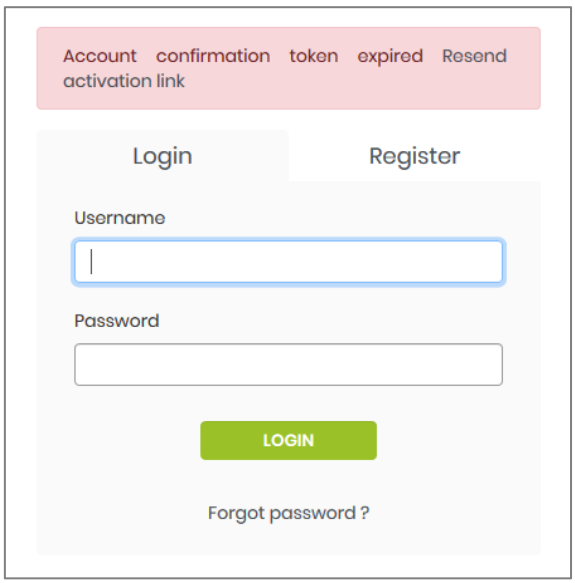

Figure 3. Expired activation token error.

2.2 Client application

After successful login user will be redirected to the main welcome page. On the left side of the page, a menu column is displayed. There are only three menu items at this version. A TPP/user which wants to start working with the PSD2 API firstly must register client application and get client credentials which later will be used with Oauth2 authorization code grant flow during the token request process. The first one is *Add application* menu item. This menu item will redirect the user to the client application add/edit form (Figure 4). In the newly opened form, a user must fill in mandatory fields: *application name, OAuth redirect URL*. *Application name* could be any text without any limitation to symbols only limited to the 128 lengths. An *OAuth redirect URL* is a URL the authorization server will redirect the user back to the application with either an authorization code or access token in the URL. Because the redirect URL will contain sensitive information, it is critical that the service does not redirect the user to arbitrary locations. A user must use the same URL with third-party OAuth client during the authorization process. Using mismatched URL will lead to Oauth error. A Scope filed is not important at this version of API. It only applies the same rule as the OAuth redirect field. If you specify this field it should match with the value passed from third-party OAuth client. A Client ID and Client secret values will be generated after form submit.

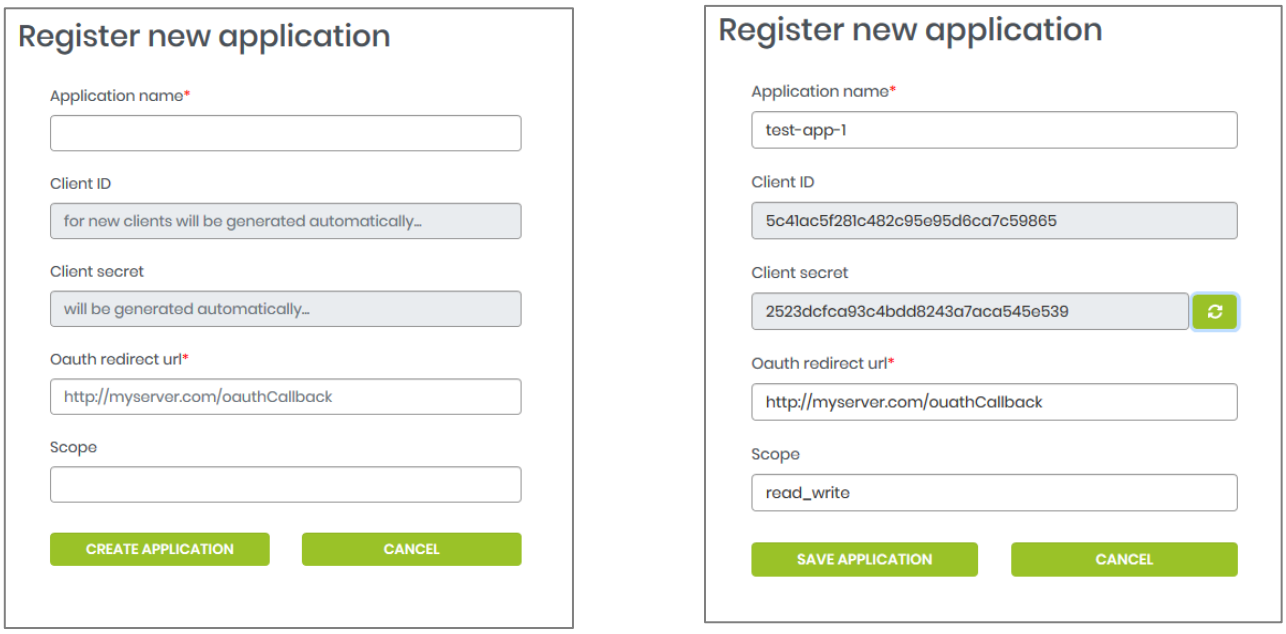

Figure 4. Client application registration and edit form

After form submit user will be redirected to the application list (Figure 5). The user can have many client applications with different OAuth client configurations for different testing purposes. A newly created application will always appear on that list too. You can remove application from the list by clicking delete icon. Clicking on the application edit button from the list will bring you to the edit form. During edit mode, the user can find out *Client ID* and *Client*  *secret* values. The *Client secret* value is only displayed once after the *regenerate* button is pressed. The next time you enter the edit form it will be hidden again. So please write down this value and keep it secretly. It is impossible to recover *Client secret* because it's stored in backend in encrypted form.

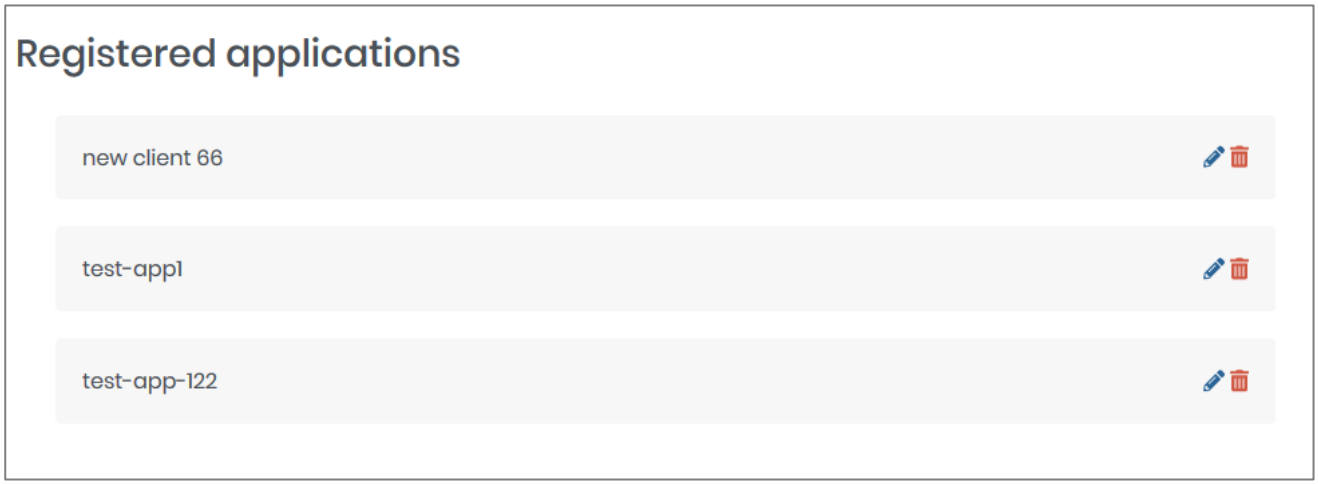

Figure 5. Client application list

2.3 Client certificate

Developer's portal provides functionality for TPP QWAC, QSEAL self-signed certificate generation for testing purpose. This could be achieved from the TPP certificate generator menu link. Once the user enters the form it has to fill correctly mandatory fields. Authorization field is the field which identifies TPP by the authorization number given by the local authority. It could consist of alphanumeric values. Other fields should be filled based on their requirement rules. A validity field is pre-filled with default 365 days value. This means that certificate after generation will be valid one year or 365 calendar days. After that period it should be regenerated again. On the last step a user has to choose certificate service roles. There are three roles (AISP, PISP, PIISP) user can choose to include into certificate. Every role gives a TPP client applications right to access to the certain group of endpoints. Deselecting service role will restrict access to related endpoints. After submitting the form, certificate data will be saved for later use and the certificate with private key will be displayed in the next window (Figure 7). You can copy or download certificate with private key values directly to the PC. The content of the certificate is not saved to backend it is only generated in memory for a current timestamp. Next time you will generate certificate it will be different and validation time will be counted from current generation timestamp. Generated certificate should be passed in mTLS transport layer.

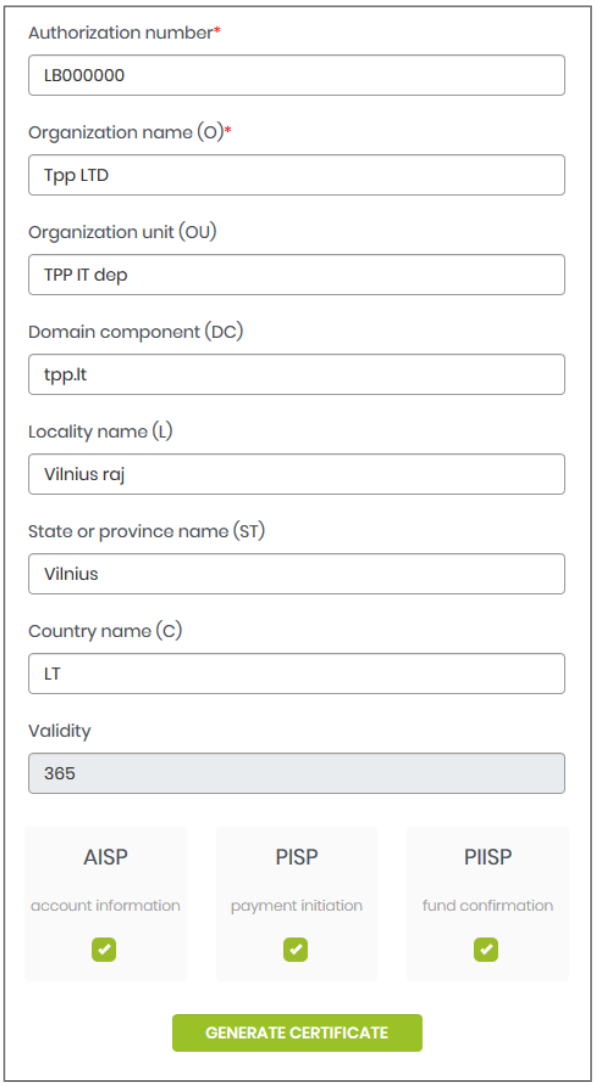

Figure 6. cerificata data form

Table 1. Roles related to core services

AISP	
	ESTABLISH ACCOUNT INFORMATION CONSENT
	GET ACCOUNT DETAILS OF THE LIST OF ACCESSIBLE ACCOUNTS
	GET BALANCES FOR A GIVEN ACCOUNT
	GET TRANSACTION INFORMATION FOR A GIVEN ACCOUNT
PISP	
	INITIATION OF A SINGLE PAYMENT
PIISP	
	GET CONFIRMATION ON THE AVAILABILITY OF FUNDS

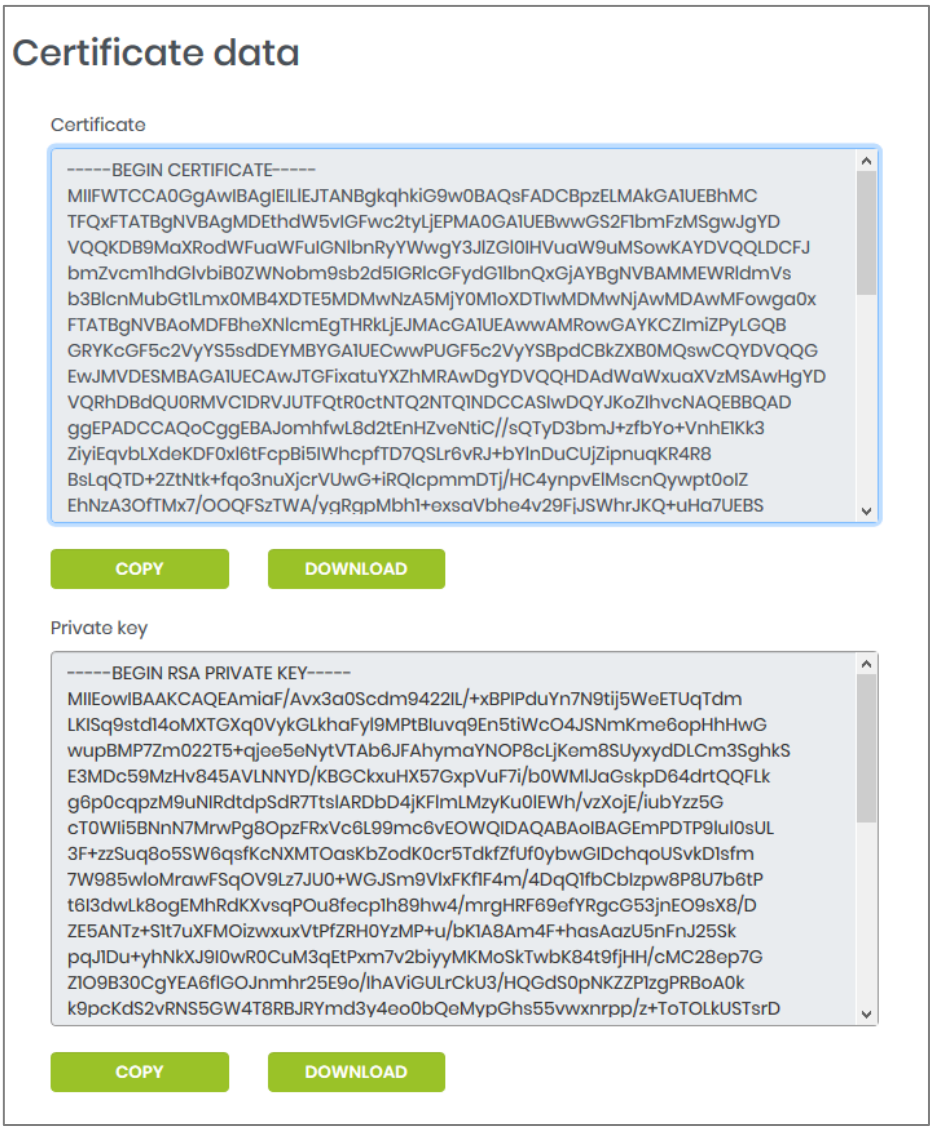

Figure 7. Certificate with public key and private key data

2.4 Testing data

Developer 's portal provides access to testing data set. It could be found via *Testing data* link from the left side menu. There are five PSU testing accounts created for test purposes. Each account has individual personal data sets, like accounts, person codes, addresses etc. attached to PSU login. The passwords for all PSU logins are 000000.

PSU ID	PSU title/name	PSU union	PSU accounts
004868	Daiva Daivaitienė	Jurbarko kredito unija	LT405013300010031000 LT585013300031011000 LT575013300032001000
422159	Jonas Jonaitis	Šilutės kredito unija	LT705010200010002000 LT925010200032001010
942050	Tomas Tomaitis	Kredito unija "Neris"	LT595016600010003333 LT265016600032003333
074060	UAB "Kubina"	Šeimos kredito unija	LT425016500014001111
495761	UAB "Ageras"	Šilutės kredito unija	LT185016500014002222

Figure 8. Dataset table inside developer 's portal

During the user's 2FA authentication process you will need to provide a TAN card number. The TAN number value is the same as the number requested on the screen. For example, if 20 TAN card number is requested, the user should enter value 20 to the input field. Basically, there are three steps for PSU authentication. After successful PSU authentication server will redirect to client's OAuth2 callback URL that was defined in client configuration form with code and state parameters. Using code, client_id and client_secret parameters TPP can request OAuth token from the server.

3. Xs2A INTERFACE

3.1 Accessing Xs2A interface

If client wants to get access to PSD2 API it should pass *Authorization* HTTP header parameter in every request. *Authorization* header contains bearer token issued by the oauth server. Before accessing Oauth server client has to register client application following steps in section 2.2. After registering application client will get c*lientId* and *clientSecret* parameters. These parameters should be passed to the Oauth server 's /authorization and /token endpoints. For example, this could be done using Postman (Figure 9).

EDIT COLLECTION

Name

Collection Name

Description | Authorization | Pre-request Scripts | Tests | Variables

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

TYPE

OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about authorization

Add auth data to

Request Headers

Current Token

This access token is only available to you. Sync the token to let collaborators on this request use it.

Access Token: Available Tokens | eyJhbGciOiJ

Header Prefix: Bearer

Configure New Token

Token Name: Token

Grant Type: Authorization Code

Callback URL: https://oauth.pstmn.io/v1/callback

Authorize using browser

Auth URL: {{oauthServerUrl}}/auth/oauth/authorize

Access Token URL: {{oauthServerUrl}}/auth/oauth/token

Client ID: {{clientId}}

Client Secret: {{clientSecret}}

Scope: PIS AIS account_list

State: 65465132465

Client Authentication: Send as Basic Auth header

Get New Access Token

Cancel | Update

Figure 9. Postman OAuth server authorization request form

Authorization header should be filled with Bearer token data received from OAuth server during PSU authentication and token request process. If this header is missing or OAuth token is not valid then a client will receive an error. Also, TPP must pass the generated QWAC certificate in the mTLS transport layer. All information about TPP roles and authorization number are parsed from this certificate. If the information in the certificate is missing or is not valid then a TPP client will get an API error. If TPP's has already issued valid EIDAS certificate they could skip certificate generation step and use their own certificates. But before that TPP must contact with Lithuania central credit union and discuss certificate exchange steps.

3.2 Getting OAuth token and user info

Redirect method

OAuth2 JWT tokens are issued requesting IdP server 's special endpoints. Server could be accessed via https://auth-dev.i-unija.lt/auth/ URL. Basically, there are two endpoints which participate in the OAuth2 flow process. The first one is responsible for the client authorization and the second one is responsible for the JWT token issuing.

Authorization endpoint GET /auth/oauth/authorize

response_type	mandatory	„code" is only supported as response type
client_id	mandatory	Generated application clientId from developer 's portal
scope	mandatory	Scope should be the same as in developer 's portal
state	mandatory	A dynamical value set by the TPP and used to prevent XSRF attacks.
redirect_uri	mandatory	the URI of the TPP where the OAuth2server is redirecting the PSU's user agent after the authorization.

CURL authorization call:

```
curl -L -X GET --url 'https://auth-dev.i-
unija.lt/auth/oauth/authorize?state=<random_state_string>
&client_id=<client_id>&scope=<scope>&response_type=code&redirect_uri=<callback_uri>'
```

Executing this call will redirect the client app to the login form where the user has to authorize himself by entering one of the credentials from the developer 's portal testing data section. After successful user authorization client app will be redirected to the *redirect_uri* parameter URL with *code* and *state* parameters (<redirect_uri>?code=<access_code>&state=<your_state_string>). After this step TPP can request token by calling token issuing endpoint.

Authorization endpoint POST /auth/oauth/token

grant_type	mandatory	„authorization_code" is only supported as grant type
client_id	mandatory	Generated application clientId from developer 's postal
client_secret	mandatory	Generated client secret from developer 's portal

code	mandatory	A code that has been received by redirect_uri parameter.
redirect_uri	mandatory	The URI of the TPP where the OAuth2server is redirecting the PSU's user agent after the authorization.

CURL authorization call:

```
curl -L -X POST --url 'https://auth-dev.i-
unija.lt/auth/oauth/token?grant_type=authorization_code
&client_id=<client_id>&client_secret=<client_secret>&code=<received_code>&redirect_uri=<c
allback_uri>' --header 'Content-Type: application/x-www-form-urlencoded' --header
'Authorization: Basic Base64_encoded_string(clientId:clientSecret)'
```

Token issuing response example.


```
{
    "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImxrdS1hdXRoLWtleS.....",
    "token_type": "bearer",
    "refresh_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImxrdS1hdXRoLWtle.....",
    "expires_in": 7195,
    "scope": "AIS PIS account_list",
    "aud": "bec9ae75f4944988bb3942ff278c11c5",
    "sub": 741806,
    "auth_method": "MSignature",
    "iss": "LKU.LT",
    "iat": 1609745619,
    "jti": "43047475-7b34-4456-b28b-05c41e582498"
}
```


Also, this process could be done via postman authorize panel (Figure 9). After authorizing request call client will be redirected to ASPSP login page where PSU 's has to enter their credentials. During this stage, PSU must authorize themselves with 2FA method.

Decoupled method

Decoupled authorization method is used where redirect method is not capable to create smooth authorization UI transition for user experience. In such environments decoupled method comes to help. Decoupled method does not do redirects, instead a client application communicates directly to authorization server via REST endpoints. One of the examples of such workflow could be a mobile application. Server could be accessed via https://auth-dev.i-unija.lt/auth/ URL. There are four endpoints involved in decoupled authorization process. The authorization endpoint is used to create authorization object in OAuth 2.0 server. The second endpoint is used to update authorization object with selected sca method. After solving SCA challenge successfully TPP can access third endpoint dedicated for authorization object status check. Status checking process should be repeated until one of the following (*finalized*, *failed*) statuses are returned. If status is *finalized,* then TPP could obtain JWT access token using token issuing endpoint. If the status is *failed,* then whole authorization process should be repeated from the beginning. Other technical information is provided within the token response.

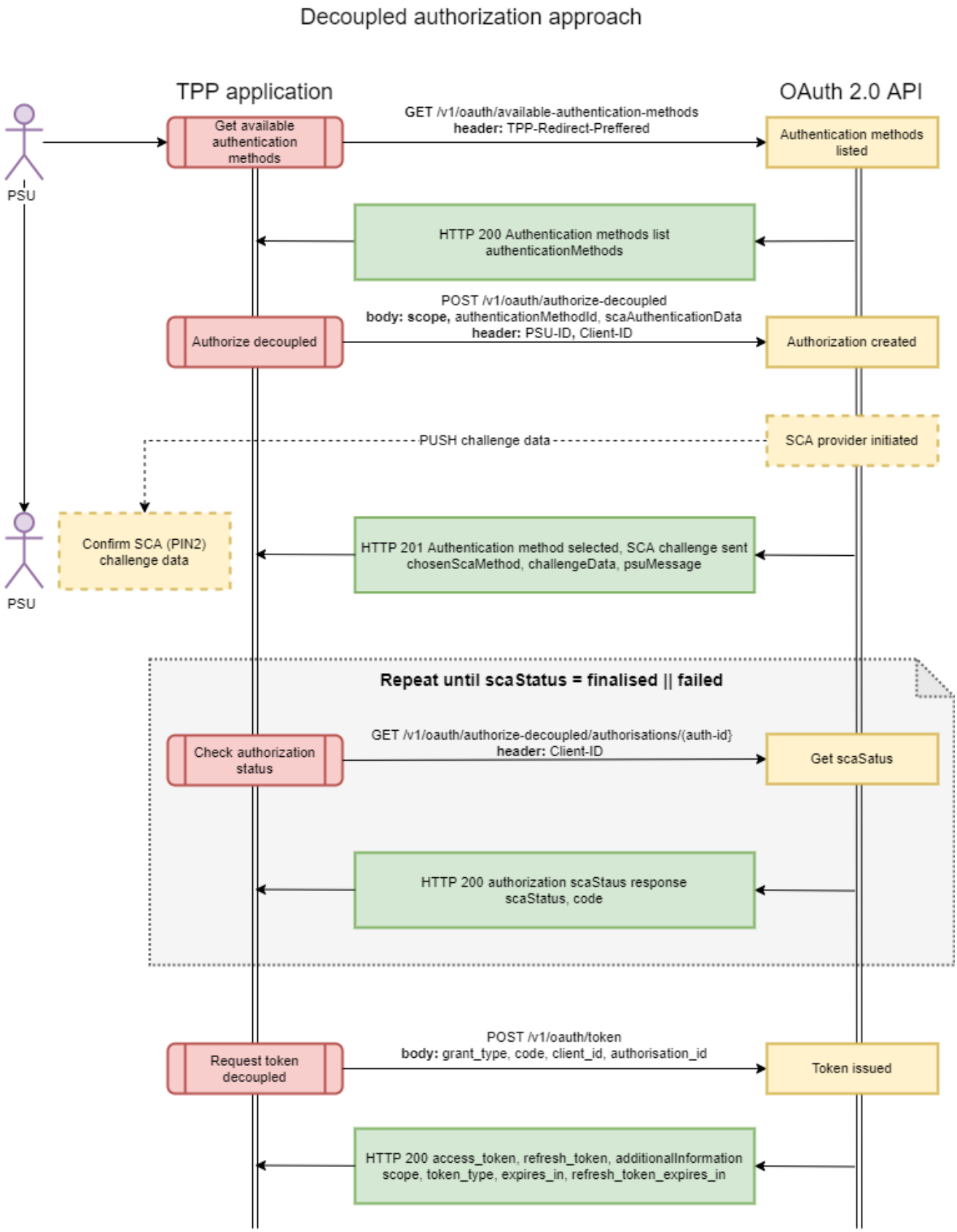

Figure 10. Decoupled authorization approach

Get available authentication methods endpoint GET /v1/available-authentication-methods

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
TPP-Redirect-Preferred	optional	If not specified or set to "false" then it will return authentication methods supported with decoupled approach. If "true" then it will return authentication methods supported with redirect approach.

Response code

201 Created	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party


```
{
    "authenticationMethods": [
        {
            "authenticationType": "Mobilus parašas",
            "authenticationMethodId": "MSignature"
        },
        {
            "authenticationType": "Smart-ID",
            "authenticationMethodId": "SmartId"
        },
        {
            "authenticationType": "Simulated auth method",
            "authenticationMethodId": "SimAuth"
        }
    ]
}
```


Initiates decouple authorization endpoint POST /v1/oauth/authorize-decoupled

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
PSU-ID	mandatory	Payment service user ID
Client-ID	mandatory	Generated application clientId from developer 's portal or issued by ASPSP

Request body

scope	mandatory	Service scopes (AIS, PIS, PIIS)
authenticationMethodId	mandatory	Select authentication method from list provided by start authorization process response
scaAuthenticationData	mandatory	Authentication data depending on sca method (phone number, person code, empty for simAuth method)

Request example

```
{
    "scope":"PIS AIS PIIS",
    "authenticationMethodId":"SmartId",
    "scaAuthenticationData":"xxxxxxxxxxxx"
}
```

Response code

201 Created	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response body

```
{
    "authorisationId": "4ff1e5e2-2195-495a-b94f-317e7920db22",
    "chosenScaMethod": {
        "authenticationMethodId": "SmartId"
    },
    "challengeData": {
        "data": [
            "xxxx"
        ],
        "otpFormat": "integer",
        "additionalInformation": "Smart-ID"
    },
    "psuMessage": "Norėdami prisijungti su Smart-ID turite atsisiųsti nemokamą programėlę į savo išmanųjį
        telefoną ar planšetinį kompiuterį.",
    "links": {
        "scaStatus": {
            "href": "/auth/v1/oauth/authorize-decoupled/authorisations/4ff1e5e2-2195-495a-b94f-317e7920db22"
        }
    }
}
```

Currently one authentication method is supported in sandbox environment *SimAuth* (with scaAuthenticationData: 0) and three methods in production: *SmartId/MSignature/Sms* in redirect method and *SmartId/MSignature* in decoupled method.

Get status endpoint GET /v1/oauth/authorize-decoupled/authorisations/{authorisation-id}

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Client-ID	mandatory	Generated application clientId from developer 's portal or issued by ASPSP

Response code

200 OK	The request has succeeded

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response body (scaStatus = finsalised)

```
{
    "scaStatus": "finalised",
    "code": "xxxxxxxxxxxxxxxx"
}
```

Response body (scaStatus = received)

```
{
    "scaStatus": "received",
    "links": {
        "self": {
            "href": "http://192.168.12.50:8089/auth/v1/oauth/authorize-decoupled/....."
        }
    }
}
```

Response body (scaStatus = started || failed)

```
{
    "scaStatus": "started"
}

{
    "scaStatus": "failed"
}
```

Access token endpoint POST /v1/oauth/token

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Content-type	mandatory	Default: application/x-www-form-urlencoded

Request body (x-www-form-urlencoded)

client_id	mandatory	Generated application clientId from developer 's portal or issued by ASPSP
client_secret	mandatory	Generated application clientSecret from developer 's portal or issued by ASPSP
code	mandatory	Scope should be the same as in developer 's portal
grant_type	mandatory	authorization_code only supported

Response code

200 OK	The request has been fulfilled

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response body


```
{
    "access_token": "xxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxx",
    "token_type": "bearer",
    "expires_in": 7120,
    "refresh_token": "xxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxx",
    "refresh_token_expires_in": "7776000",
    "scope": "PIIS AIS PIS",
    "additionalInformation": {
        "aud": "xxxxxxxxxxxxxxxxxxx",
        "sub": 100000000,
        "iss": "LKU.LT",
        "iat": 1620380090,
        "jti": "xxxxxxxxxxxxxxxxxxx"
    }
}
```


Refresh token endpoint POST /v1/oauth/token

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Content-type	mandatory	Default: application/x-www-form-urlencoded

Request body (x-www-form-urlencoded)

client_id	mandatory	Generated application clientId from developer 's portal or issued by ASPSP
client_secret	mandatory	Generated application clientSecret from developer 's portal or issued by ASPSP
code	mandatory	Scope should be the same as in developer 's portal
grant_type	mandatory	refresh_token only supported

Response code

200 OK	The request has been fulfilled

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response body


```
{
    "access_token": "xxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxx",
    "token_type": "bearer",
    "expires_in": 7120,
    "refresh_token": "xxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxx.xxxxxxxxxxxxxxxxxxx",
    "refresh_token_expires_in": "7776000",
    "scope": "PIIS AIS PIS",
    "additionalInformation": {
        "aud": "xxxxxxxxxxxxxxxxxxx",
        "sub": 100000000,
        "iss": "LKU.LT",
        "iat": 1620380090,
        "jti": "xxxxxxxxxxxxxxxxxxx"
    }
}
```


Invalidate token endpoint POST /v1/oauth/invalidate-token

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
PSU-ID	mandatory	Payment service user ID
Client-ID	mandatory	Generated application clientId from developer 's portal or issued by ASPSP

Response code

204 No content	Indicates that the server has successfully fulfilled the request and that there is no content to send in the response payload body

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

User info

User info endpoint provides API client with the information related to the currently authenticated user. *Sub* field provides autnenticated user PSU-ID, *name* field provides PSU user full name or company name if it is corporate user and *kycVerified* field shows if the user has been verified in banking system. In the future this endpoint will be supplemented with additional information related to PSU credit union name, address etc.

Get user info endpoint GET /v1/userinfo

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
PSU-ID	mandatory	Payment service user ID
Client-ID	mandatory	Generated application clientId from developer 's portal or issued by ASPSP

Response code

200 OK	The request has succeeded

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response body

```
{
    "sub": "455693",
    "name": "Jonas Jonaitis",
    "kycVerified": true
}
```

3.3 PSU request context data

TPP-PSU request data must be passed in HTTP header and strongly recommended to be used in every request (AIS, PIS, PIIS services). These parameters hold various information related to PSU user and mainly are used for risk management and fraud detection. In current implementation no business logic depends on the field values.

PSU-IP-Address	optional	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP.It shall be contained if and only if this request was actively initiated by the PSU.
PSU-IP-Port	optional	The forwarded IP Port header field consists of the corresponding HTTP request IP Port field between PSU and TPP, if available.
PSU-Accept	optional	The forwarded IP Accept header fields consist of the corresponding HTTP request Accept header fields between PSU and TPP, if available.

PSU-Accept-Charset	optional	Not used
PSU-Accept-Encoding	optional	Not used
PSU-Accept-Language	optional	Not used
PSU-User-Agent	optional	The forwarded Agent header field of the HTTP request between PSU and TPP, if available.
PSU-Http-Method	optional	HTTP method used at the PSU –TPP interface, if available.
PSU-Device-ID	optional	UUID (Universally Unique Identifier) for a device, which is used by the PSU, if available.
PSU-Geo-Location	optional	The forwarded Geo Location of the corresponding HTTP request between PSU and TPP if available.

3.4 AIS endpoints

Establish account information consent

Establish account information consent is the first step of PSD2 API account data exchange process. An ais role is needed for accessing this endpoint. Four types of consent model could be applied when creating consent.

Consent model	Description	Example

Bank offered consent	ASPSP returns a list of accounts (only accounts that are accessible through xs2a according to internal bank rules) with all rights and accounts selected by default. This type of consent model gives the possibility for the PSU to select accounts and rights during consent SCA process (Figure 10).	{
		"access": {
		"accounts": [],
		"balances": [],
		"transactions": []
		},
		"frequencyPerDay": 10,
		"recurringIndicator": true,
		"validUntil": "2020-10-10"
		}

Detailed consent	The Consent Management is handled between TPP and PSU. TPP sends a request with detailed accounts list and rights. PSU cannot select or alter consent data during the SCA process (Figure 10). The ASPSP is displaying the consent details to the PSU when performing the SCA.	{
		"access": {
		"accounts": [
		{
		"iban": "LT405013300010031000"
		},
		{
		"iban": "LT575013300032001000"
		}
		],
		"balances": [
		{
		"iban": "LT405013300010031000"
		}
		],
		"transactions": [
		{
		"iban": "LT575013300032001000"
		}
		]

		}, "frequencyPerDay": 10, "recurringIndicator": true, "validUntil": "2020-10-10" }
Available accounts consent	With this consent, TPP gets access to all accounts with all rights. Only accounts that are accessible through xs2a according to internal bank rules are used in consent. The ASPSP is displaying only the general access to the PSU's account to the PSU when performing the SCA. PSU cannot select or alter consent data during the SCA process (Figure 10).	{ "access": { "availableAccounts": "allAccounts" }, "frequencyPerDay": 10, "recurringIndicator": true, "validUntil": "2020-10-10" }
Global consent	The Consent Management is handled between TPP and PSU. The TPP is submitting then a global consent information, which is only the PSU identification, to the ASPSP for authorization by the PSU. If this function is supported, it will imply a consent on all available accounts of the PSU on all PSD2 related account information services. For this specific Consent Request, no assumptions are made for the SCA Approach by this specification.	{ "access": { "allPsd2": "allAccounts" }, "frequencyPerDay": 10, "recurringIndicator": true, "validUntil": "2020-10-10" }

Request POST /v1/consents/

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
TPP-Redirect-Preferred	optional	If it equals "true", the TPP prefers a redirect over an embedded SCA Approach.
TPP-Redirect-URI	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
TPP-Nok-Redirect-URI	conditional	If this URI is contained, the TPP is asking to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the redirect SCA method. This might be ignored by the ASPSP.
TPP-Explicit-Authorisation-Preferred	optional	If it equals "true", the TPP prefers to start the authorisation process separately. If it equals "false" or if the parameter is not used, there is no preference of the TPP.
Content-Type	optional	Content type application/json

Request body

Access	mandatory	Requested access services

recurringIndicator	mandatory	True, if the consent is for recurring access to the account data. false, if the consent is for one access to the account data
validUntil	mandatory	This parameter is requesting a valid until date for the requested consent.
frequencyPerDay	mandatory	This field indicates the requested maximum frequency for an access per day. For a one-off this attribute is set to "1".
combinedServiceIndicator	mandatory	The request is a part of requests session

Request example


```
{
    "access": {
        "accounts": [
            {
                "iban": "LT705010200010002000",
                "currency": "EUR"
            },
            {
                "iban": "LT925010200032001010",
                "currency": "EUR"
            }
        ],
        "balances": [
            {
                "iban": "LT705010200010002000",
                "currency": "EUR"
            }
        ],
        "transactions": [
            {
                "iban": "LT705010200010002000",
                "currency": "EUR"
            },
            {
                "iban": "LT925010200032001010",
                "currency": "EUR"
            }
        ]
    },
    "frequencyPerDay": 10,
    "recurringIndicator": true,
    "validUntil": "2020-10-10"
}
```


Response code

201 Created	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party
Aspsp-Sca-Approach	Possible values are REDIRECT or DECOUPLED

Response example (TPP-Redirect-Preferred = true/false/null, TPP-Explicit-Authorisation-Preferred = true)

```
{
    "consentStatus": "received",
    "consentId": "9c143e1b-a898-4df8-99c2-cf8856306326",
    "_links": {
        "self": {
            "href": "/v1/consents/9c143e1b-a898-4df8-99c2-cf8856306326"
        },
        "startAuthorisation": {
            "href": "/v1/consents/9c143e1b-a898-4df8-99c2-cf8856306326/authorisations"
        },
        "status": {
            "href": "/v1/consents/9c143e1b-a898-4df8-99c2-cf8856306326/status"
        }
    }
}
```


Response example (TPP-Redirect-Preferred = true, TPP-Explicit-Authorisation-Preferred = false)


```
{
    "consentStatus": "received",
    "consentId": "97f9b9e1-bce3-4690-83ef-d0cad5e34955",
    "_links": {
        "self": {
            "href": "/v1/consents/97f9b9e1-bce3-4690-83ef-d0cad5e34955"
        },
        "scaStatus": {
            "href": "/v1/consents/97f9b9e1-bce3-4690-83ef-d0cad5e34955/authorisations/
                b7f64a26-62d0-4790-8aaa-e6cde7ff0798"
        },
        "scaRedirect": {
            "href": "https://psd2.i-unija.lt/account/b7f64a26-62d0-4790-8aaa-e6cde7ff0798/"
        },
        "status": {
            "href": "/v1/consents/97f9b9e1-bce3-4690-83ef-d0cad5e34955/status"
        }
    }
}
```


Get consent status

Request GET /v1/consents/{consentId}/status

Path parameters

consentId	The consent identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by

		the initiating party
Authorization	mandatory	Oauth2 authorization bearer token

Response code

200 OK	The request has succeeded

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example

```
{
    "consentStatus": "received"
}
```

Get consent

Request GET /v1/consents/{consentId}

Path parameters

consentId	The consent identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Response code

200 OK	**The request has succeeded**

Response header

X-Request-ID	**ID of the request, unique to the call, as determined by the initiating party**

Response example


```
{
    "access": {
        "accounts": [
            {
                "iban": "LTXXXXXXXXXXXXXXXXXXXX",
                "currency": "EUR"
            },
            {
                "iban": "LTXXXXXXXXXXXXXXXXXXXX",
                "currency": "EUR"
            }
        ],
        "transactions": [
            {
                "iban": "LTXXXXXXXXXXXXXXXXXXXX",
                "currency": "EUR"
            }
        ]
    },
    "recurringIndicator": true,
    "validUntil": "2021-08-05",
    "frequencyPerDay": 4,
    "lastActionDate": "2021-05-07",
    "consentStatus": "valid"
}
```


Response example (in case of global consent)


```
{
    "access": {
        "allPsd2": "allAccounts"
    },
    "recurringIndicator": true,
    "validUntil": "9999-12-31",
    "frequencyPerDay": 2147483647,
    "lastActionDate": "2021-05-07",
    "consentStatus": "valid"
}
```


Consent authorizations: redirect SCA approach

During this approach TPP must send *Tpp-Redirect-Preffered* header set to true. This means that consent will be authorized in redirect approach. Also, there are two ways how consent authorization object will be created in redirect manner: implicit and explicit. Implicit method will create authorization object during *create consent* call. No sequential calls are needed. A *scaRedirect* steering link will be added to the *create consent* JSON response. Following this redirect link a PSU will be redirect to the LCKU consent summary and

SCA selection and approval form where PSU must enter their PIN2 credentials. Also, *Aspsp-Sca-Approach: REDIRECT* header will be added to the response.

Figure 11. Account selection

Using explicit method TPP will have to make additional call for consent authorization object creation. A separate call *starts the authorisation process for consent* will create consent authorization object and return *scaRedirect* steering link inside JSON response. Same as in implicit method following this redirect link will redirect PSU to the LCKU consent summary and SCA selection, approval form. It's highly recommended to use implicit method with SCA redirect approach.

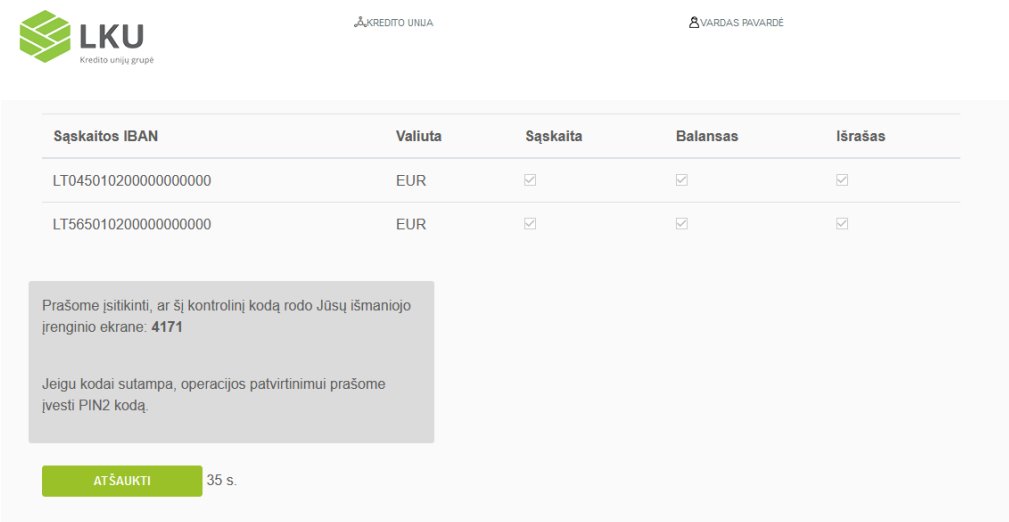

Figure 12. Confirmation of consent using SCA method

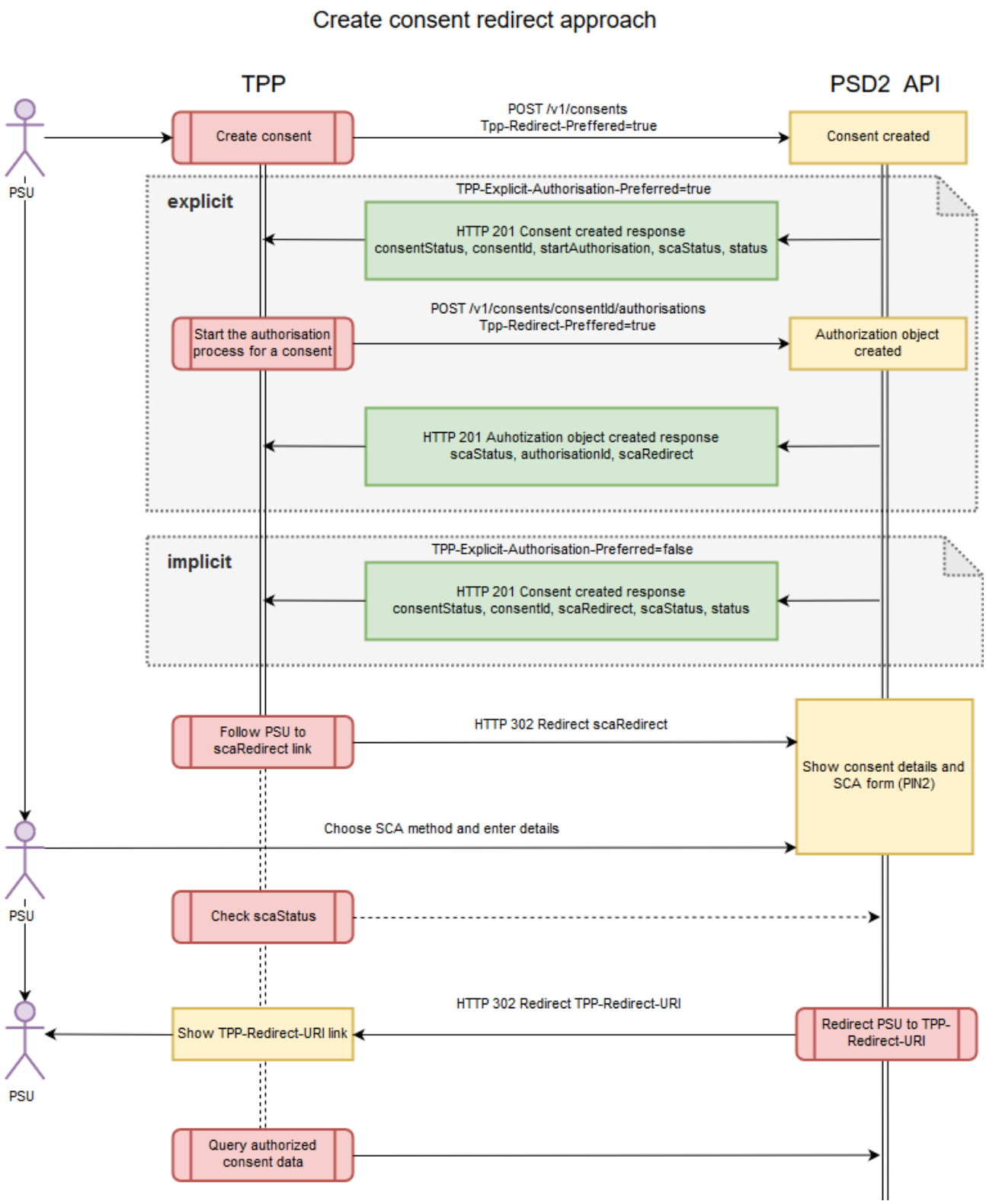

Figure 13. Create consent redirect approach

Consent authorisations: decoupled SCA approach

The main difference of *redirect* approach from *decoupled* is that PSU has to enter their credential details in ASPSP environment. In *decoupled* approach an explicit authorisation method only exists this means that TPP has always to make additional calls to the API after *create consent* call execution. In the first step TPP has to call *create consent* endpoint without *Tpp-Redirect-Preffered* header or setting this header value to false. In response TPP will get *startAuthorisation* steering link. In the second step TPP has to *start authorization process for a consent* using HTTP POST method. After executing this call TPP will receive a list of available SCA methods inside *scaMethods* array and *selectAuthenticationMethod* hyperlink in the JSON response. SCA methods list should be depicted in TPP environment so that PSU could select preferred SCA method (mobile signature, smart ID and etc.).


```
{
    "scaStatus": "received",
    "authorisationId": "3ad01c54-984e-4c19-ada6-eef77719b5c8",
    "scaMethods": [
        {
            "authenticationType": "PUSH_OTP",
            "authenticationMethodId": "SmartId",
            "name": "Smart-ID",
            "explanation": "Norėdami prisijungti su Smart-ID turite atsisiųsti nemokamą programėlę į savo išmanųjį
                telefoną ar planšetinį kompiuterį."
        }
    ],
    "_links": {
        "selectAuthenticationMethod": {
            "href": "/v1/consents/a5839962-f91b-438c-90c3-4b623ba37721/authorisations/
                3ad01c54-984e-4c19-ada6-eef77719b5c8"
        },
        "scaStatus": {
            "href": "/v1/consents/a5839962-f91b-438c-90c3-4b623ba37721/authorisations/
                3ad01c54-984e-4c19-ada6-eef77719b5c8"
        }
    }
}
```


After PSU selects method TPP should initiate *update PSU data for consent* call executing *selectAuthenticationMethod* link with PUT HTTP method and JSON request body with *authenticationMethodId* element which contains method ID from the SCA methods.

```
{
    "authenticationMethodId": "{{authentication-method-id}}"
}
```

During this call ASPSP must initialize internal SCA provider's process which will push OTP challenge data to the PSU device and adds same challenge code data to the JSON response of the *update PSU data for consent* request. PSU must confirm this challenge using PIN2 code. If the confirmation was successful consent status will be changed to *valid* and authorization object will be *finalized*. If the authorization is unsuccessful consent status will not change but authorization object status will be changed to failed. In this case TPP should start authorization process from the second step: *start the authorization process for consent*. More information about request and response structure could be found in the 34 page.

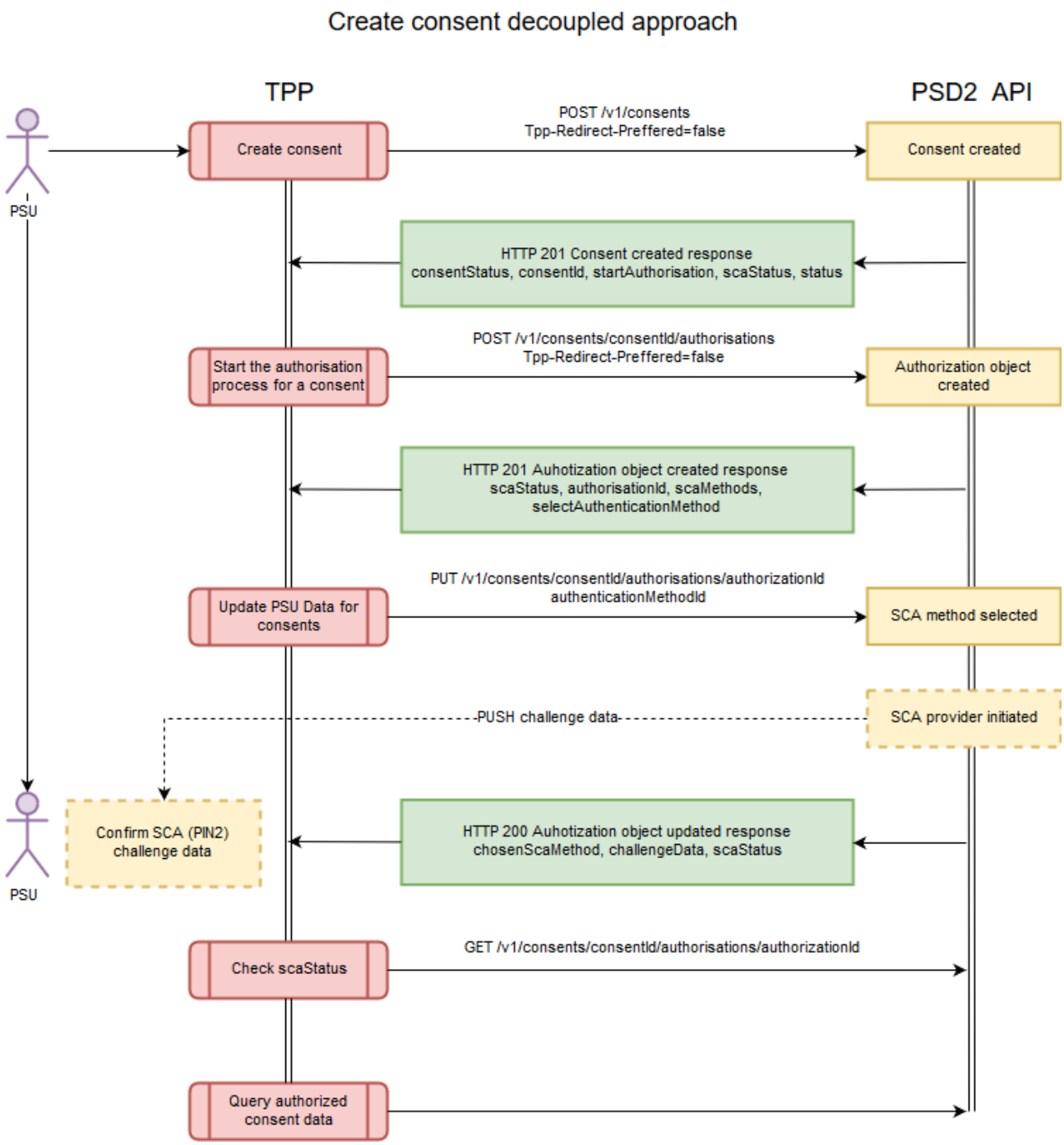

Figure 14. Create consent decoupled approach

Start the authorisation process for a consent

Request POST /v1/consents/{consent-id}/authorisations

Path parameters

consent-id	The consent identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
TPP-Redirect-Preferred	optional	If it equals "true", the TPP prefers a redirect over an embedded SCA Approach.
TPP-Redirect-URI	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
TPP-Nok-Redirect-URI	conditional	If this URI is contained, the TPP is asking to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the redirect SCA method. This might be ignored by the ASPSP.
Content-Type	optional	Content type application/json

Response code

201 Created	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party
Aspsp-Sca-Approach	Possible values are REDIRECT or DECOUPLED

Response example (TPP-Redirect-Preferred = false)


```
{
    "scaStatus": "received",
    "authorisationId": "600bb724-f4f8-4342-b59b-b94b3da5a9c4",
    "scaMethods": [
        {
            "authenticationType": "PUSH_OTP",
            "authenticationMethodId": "SmartId",
            "name": "Smart-ID",
            "explanation": "Norėdami prisijungti su Smart-ID turite atsisiųsti nemokamą programėlę į savo išmanųjį
                telefoną ar planšetinį kompiuterį."
        }
    ],
    "_links": {
        "selectAuthenticationMethod": {
            "href": "/v1/consents/7a82a31b-80e0-4139-a6ee-381a768ec866/authorisations/
                600bb724-f4f8-4342-b59b-b94b3da5a9c4"
        },
        "scaStatus": {
            "href": "/v1/consents/7a82a31b-80e0-4139-a6ee-381a768ec866/authorisations/
                600bb724-f4f8-4342-b59b-b94b3da5a9c4"
        }
    }
}
```


Response example (TPP-Redirect-Preferred = true, TPP-Redirect-URI=http://....)


```
{
    "scaStatus": "received",
    "authorisationId": "ea9ac56e-dc91-4eca-91a6-e3b6f9aba714",
    "_links": {
        "scaRedirect": {
            "href": "https://psd2.i-unija.lt/account/ea9ac56e-dc91-4eca-91a6-e3b6f9aba714/"
        },
        "scaStatus": {
            "href": "/v1/consents/7a82a31b-80e0-4139-a6ee-381a768ec866/authorisations/
                ea9ac56e-dc91-4eca-91a6-e3b6f9aba714"
        }
    }
}
```


Update PSU data for consent (only for decoupled method)

Request PUT /v1/consents/{consent-id}/authorisations/{authorisation-id}

Path parameters

consent-id	The consent identification assigned to the created resource
authorisation-id	Authorisation object ID

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Request body

authenticationMethodId	mandatory	Select authentication method from list provided by start authorisation process response

Request example

```
{
  "authenticationMethodId": "{{authentication-method-id}}"
}
```

Response code

200 OK	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party
Aspsp-Sca-Approach	Possible values are: REDIRECT or DECOUPLED

Response example


```
{
    "chosenScaMethod": {
        "authenticationMethodId": "SmartId"
    },
    "challengeData": {
        "data": [
            "0000"
        ],
        "otpFormat": "integer",
        "additionalInformation": "Smart-ID parašas"
    },
    "_links": {
        "self": {
            "href": "/v1/consents/7a82a31b-80e0-4139-a6ee-381a768ec866"
        },
        "scaStatus": {
            "href": "/v1/consents/7a82a31b-80e0-4139-a6ee-381a768ec866/authorisations/..."
        },
        "status": {
            "href": "/v1/consents/7a82a31b-80e0-4139-a6ee-381a768ec866/status"
        }
    },
    "scaStatus": "started",
    "psuMessage": "Smart-ID parašas"
}
```


Read the SCA status of the consent authorization

Request GET /v1/consents/{consent-id}/authorisations/{authorisation-id}

Path parameters

consent-id	**The consent identification assigned to the created resource**
authorisation-id	Authorisation object ID (in case of payment authorisation)

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Response code

200 OK	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example

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

Get Consent Authorisation Sub-Resources

Request GET /v1/consents/{consent-id}/authorisations

Path parameters

consent-id	**The consent identification assigned to the created resource**

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Response code

200 OK	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example


```
{
    "authorisationIds": [
        "69e31e32-96cd-439d-9eb3-d964b7fc855c",
        "f0de36e2-1c7e-42c0-9927-4f9e7aa78edc",
        "ef338d0f-2488-4ded-810a-eb04464db5b7"
    ]
}
```


Delete consent

Request DELETE /v1/consents/{consent-id}

Path parameters

consent-id	The consent identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token

Response code

204 No content	The request has succeeded

Response header

X-Request-ID	The server has successfully fulfilled the request and that there is no additional content to send in the response payload body

Read account list

Request GET /v1/accounts

Query parameters

withBalance	If contained, this function reads the list of accessible payment accounts including the booking balance, if granted by the PSU in the related consent and available by the ASPSP.

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Consent-Id	mandatory	Shall be contained since "Establish Consent Transaction" was performed via this API before.
Authorization	mandatory	Oauth2 authorization bearer token

Response code

200 OK	The request has succeeded

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example (withBalance = true)


```
{
    "accounts": [
        {
            "resourceId": "c1c042c0-5d96-649a-e053-030ca8c05a26",
            "iban": "LTXXXXXXXXXXXXXXXXXX",
            "currency": "EUR",
            "name": "einamoji s-ta",
            "ownerName": "Jonas Jonaitis",
            "product": "CurrentMember",
            "cashAccountType": "CACC",
            "status": "enabled",
            "bic": "LCKULT22XXX",
            "usage": "PRIV",
            "balances": [
                {
                    "balanceAmount": {
                        "currency": "EUR",
                        "amount": "361.2"
                    },
                    "balanceType": "interimAvailable",
                    "referenceDate": "2021-05-19"
                }
            ],
            "_links": {
                "balances": {
                    "href": "/v1/accounts/c1c042c0-5d96-649a-e053-030ca8c05a26/balances"
                },
                "transactions": {
                    "href": "/v1/accounts/c1c042c0-5d96-649a-e053-030ca8c05a26/transactions"
                }
            }
        }
    ]
}
```


Read account details

Request GET /v1/accounts/{account-id}

Path parameters

account-id	The account identification assigned to the created resource

Query parameters

withBalance	If contained, this function reads the list of accessible payment accounts including the booking balance, if granted by the PSU in the related consent and available by the ASPSP.

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Consent-ID	mandatory	Shall be contained since "Establish Consent Transaction" was performed via this API before.
Authorization	mandatory	Oauth2 authorization bearer token

Response code

200 OK	The request has succeeded

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example


```
{
    "resourceId": "c1c042c0-5d96-649a-e053-030ca8c05a26",
    "iban": "LTXXXXXXXXXXXXXXXXXX",
    "currency": "EUR",
    "name": "einamoji s-ta",
    "ownerName": "Jonas Jonaitiss",
    "product": "CurrentMember",
    "cashAccountType": "CACC",
    "status": "enabled",
    "bic": "LCKULT22XXX",
    "usage": "PRIV",
    "balances": [
        {
            "balanceAmount": {
                "currency": "EUR",
                "amount": "361.2"
            },
            "balanceType": "interimAvailable",
            "referenceDate": "2021-05-19"
        }
    ],
    "_links": {
        "balances": {
            "href": "/v1/accounts/c1c042c0-5d96-649a-e053-030ca8c05a26/balances"
        },
        "transactions": {
            "href": "/v1/accounts/c1c042c0-5d96-649a-e053-030ca8c05a26/transactions"
        }
    }
}
```

Get balances

Request GET /v1/accounts/{account-id}/balances

Path parameters

account-id	**The account identification assigned to the created resource**

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Consent-ID	mandatory	
Authorization	mandatory	Oauth2 authorization bearer token

Response code

200 OK	The request has succeeded

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example

```
{
    "account": {
        "iban": "LTXXXXXXXXXXXXXXXX",
        "currency": "EUR"
    },
    "balances": [
        {
            "balanceAmount": {
                "currency": "EUR",
                "amount": "487.02"
            },
            "balanceType": "interimAvailable",
            "referenceDate": "2021-01-05"
        }
    ]
}
```

Get transactions list

Request GET /v1/accounts/{account-id}/transactions

Path parameters

account-id	The account identification assigned to the created resource

Query parameters

dateFrom	Starting date (inclusive the date dateFrom) of the transaction list, mandated if no delta access is required.
dateTo	End date (inclusive the data dateTo) of the transaction list, default is "now" if not given.
bookingStatus	Permitted codes are „booked“ , „pending“ and „both“
withBalance	If contained, this function reads the list of transactions including the booking balance, if granted by the PSU in the related consent and available by the ASPSP.
size	Number of transactions to return
page	Page of results to return
query	Query for search in creditor, debtor (name or IBAN) or description.

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Consent-ID	mandatory	
Authorization	mandatory	Oauth2 authorization bearer token
Accept	mandatory	in current version of API application/json is only supported

Response code

200 OK	**The request has succeeded**

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example


```
{
    "account": {
        "iban": "LTXXXXXXXXXXXXXXXXX",
        "currency": "EUR"
    },
    "transactions": {
        "pending": [
            {
                "transactionId": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
                "endToEndId": "LT-XXXXXXXXXXXXXXX",
                "bookingDate": "2021-04-27",
                "valueDate": "2021-04-27",
                "transactionAmount": {
                    "currency": "EUR",
                    "amount": "-0.36"
                },
                "creditorName": "Swedbank AB",
                "creditorAccount": {
                    "iban": "DEXXXXXXXXXXXXXXXXXXXXXXXX",
                    "currency": "EUR"
                },
                "remittanceInformationUnstructured": "Paskirtis uz daikta 445566"
            }
        ],
        "_links": {
            "next": {
                "href": "/v1/accounts/c1c06522-444f-75cd-e053-030ca8c0e192/transactions/?dateFrom=2021-02-28&dateTo=2021-05-01&
                    bookingStatus=BOTH&page=1&size=15&query=food+and+drinks"
            },
            "last": {
                "href": "/v1/accounts/c1c06522-444f-75cd-e053-030ca8c0e192/transactions/?dateFrom=2021-02-28&dateTo=2021-05-01&
                    bookingStatus=BOTH&page=21&size=15&query=food+and+drinks"
            },
            "account": {
                "href": "/v1/accounts/c1c06522-444f-75cd-e053-030ca8c0e192"
            },
            "first": {
                "href": "/v1/accounts/c1c06522-444f-75cd-e053-030ca8c0e192/transactions/?dateFrom=2021-02-28&dateTo=2021-05-01&
                    bookingStatus=BOTH&page=0&size=15&query=food+and+drinks"
            }
        }
    }
}
```


Get transaction details

Request GET /v1/accounts/{account-id}/transactions/{transaction-id}

Path parameters

account-id	The account identification assigned to the created resource
transaction-id	This identification is given by the attribute resourceId of the corresponding entry of a transaction list.

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Consent-ID	mandatory	
Authorization	mandatory	Oauth2 authorization bearer token

Response code

200 OK	**The request has succeeded**

Response header

Content-type	Possible value: application/json
X-Request-ID	**ID of the request, unique to the call, as determined by the initiating party**

Response example


```
{
    "transactionDetails": {
        "transactionId": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
        "endToEndId": "LT-1234567890",
        "bookingDate": "2021-04-19",
        "valueDate": "2021-04-19",
        "transactionAmount": {
            "currency": "EUR",
            "amount": "-0.36"
        },
        "creditorName": "Swedbank AB",
        "creditorAccount": {
            "iban": "DEXXXXXXXXXXXXXXXXXXXXXX",
            "currency": "EUR"
        },
        "remittanceInformationUnstructured": "Paskirtis uz daikta 445566"
    }
}
```


3.5 PIS endpoints

Payment states transitions

During the payment process payment, states could be changed eventually (Figure 15). Some state changes could be achieved via external endpoint call and other changes are achieved during an internal isolated process. After successful payment initiation call, payment state will be changed to **PDGN** (pending). This

means that payment was created but not authorized yet by the PSU and will not be proceeded further. In this state, a payment could be canceled using delete payment method without any authorization need. After payment deletion, the state will be changed to **CANC** (canceled). From this state, no other state could be reached. The payment should be created from the beginning. After successful payment authorization, one of the two states could be reached. To which state will be transited to depends on some predefined internal conditions. If no conditions are applied to the payment an **ACSC** (accepted settlement completed) state will be set. This means that the payment settlement on the debtor 's account has been completed. The payment is ready to be executed. If some validation and execution error occur during the process, then the payment will be rejected and the state will be changed to **RJCT** (rejected) state. From this point, no further state changes could be reached. Such payment should be initialized from the beginning. If some internal conditions are applied, then in such case **PATC** (partially accepted technically correct) state will be set. This means that some internal approvement process should be applied. During this state, payment could be canceled at any time, but the cancelation process should be authorized by the PSU via start the authorization process for the cancellation of the addressed payment endpoint. If the TPP tries to cancel payment via delete endpoint, then it will receive a link inside JSON response to the payment cancellation endpoint. Also, payment could be rejected if some validation error occurred during execution process otherwise current **ACFC** state will be changed to final **ACSC** state.

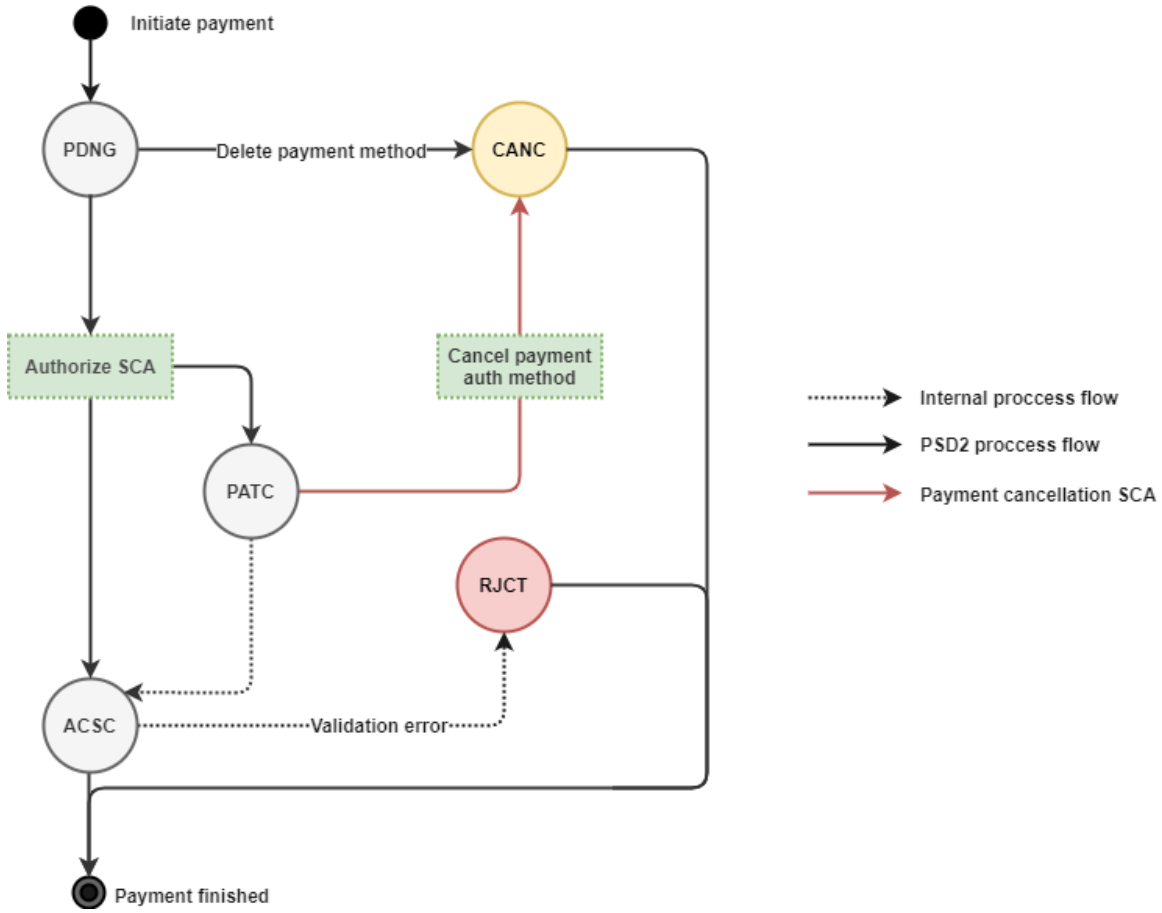

Figure 15. Payment states transition during payment process

Payment authorizations: redirect SCA approach

During this approach TPP must send *Tpp-Redirect-Preffered* header set to true. This means that payment will be authorized in redirect approach. Also, there are two ways how payment authorization object will be created in redirect manner: implicit and explicit. Implicit method will create authorization object during *initiate payment* call. No sequential calls are needed. A *scaRedirect* steering link will be added to the *initiate payment* JSON response. Following this redirect link a PSU will be redirect to the LCKU payment summary and SCA selection and approval form where PSU must enter their PIN2 credentials. Also, *Aspsp-Sca-Approach: REDIRECT* header will be added to the response. Using explicit method TPP will have to make additional call for consent authorization object creation. A separate call *starts the authorisation process for a payment* will create consent authorization object and return *scaRedirect* steering link inside JSON response. Same as in implicit method following this redirect link will redirect PSU to the LCKU payment summary and SCA selection, approval form. It is highly recommended to use implicit method with SCA redirect approach.

Figure 16. Payment approves using redirect method.

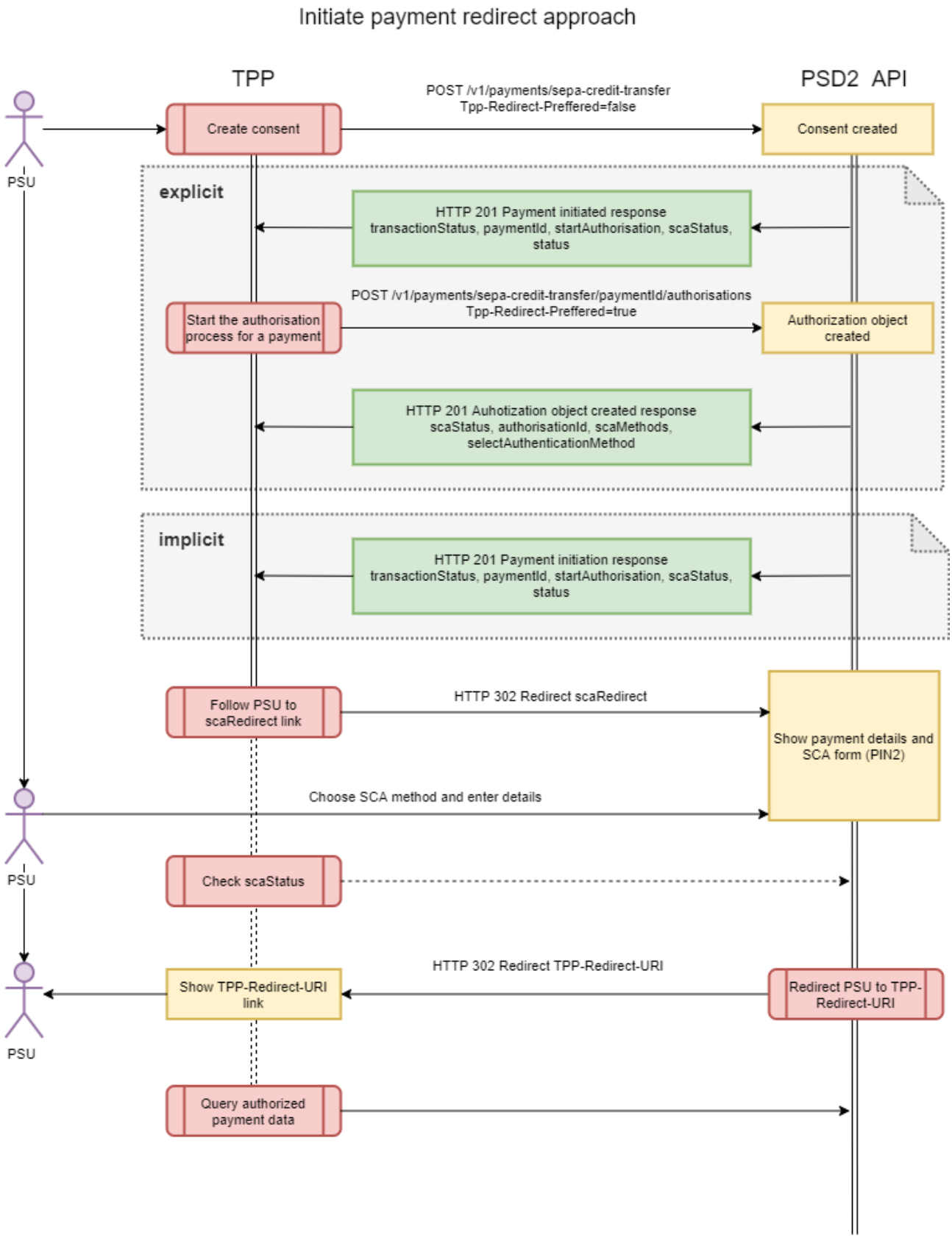

Figure 17. Initiate payment redirect approach

Payment authorizations: decoupled SCA approach

The main difference of *redirect* approach from *decoupled* is that PSU must enter their credential details in ASPSP environment. In *decoupled* approach an explicit authorisation method only exists this means that TPP has always to make additional calls to the API after *create consent* call execution. In the first step TPP has to call *initiate payment* endpoint without *Top-Redirect-Preffered* header or setting this header value to false. In response TPP will get *startAuthorisation* steering link. In the second step TPP must *start authorization process for a payment initiation* using HTTP POST method. After executing this call TPP will receive a list of available SCA methods inside *scaMethods* array and *selectAuthenticationMethod* hyperlink in the JSON response. SCA methods list should be depicted in TPP environment so that PSU could select preferred SCA method (mobile signature, smart ID etc.).


```
{
    "scaStatus": "received",
    "authorisationId": "9effc5fa-2439-4cc0-95a8-bdbd27d16d1a",
    "scaMethods": [
        {
            "authenticationType": "PUSH_OTP",
            "authenticationMethodId": "SmartId",
            "name": "Smart-ID",
            "explanation": "Norėdami prisijungti su Smart-ID turite atsisiųsti nemokamą programėlę į savo
                išmanųjį telefoną ar planšetinį kompiuterį."
        }
    ],
    "_links": {
        "selectAuthenticationMethod": {
            "href": "/v1/payments/sepa-credit-transfers/
                7FDE2B42D9687B7F2254D9FDBD72E8636CAC4D3AA4C323C0B8D9D8AE48820ACB/authorisations/
                9effc5fa-2439-4cc0-95a8-bdbd27d16d1a"
        },
        "scaStatus": {
            "href": "/v1/payments/sepa-credit-transfers/
                7FDE2B42D9687B7F2254D9FDBD72E8636CAC4D3AA4C323C0B8D9D8AE48820ACB/authorisations/
                9effc5fa-2439-4cc0-95a8-bdbd27d16d1a"
        }
    }
}
```


After PSU selects method TPP should initiate *update PSU data for payment initiation* call executing *selectAuthenticationMethod* link with PUT HTTP method and JSON request body with *authenticationMethodId* which contains method ID from the SCA methods list .

```
{
    "authenticationMethodId": "{{authentication-method-id}}"
}
```

During this call ASPSP must initialize internal SCA providers process which will push OTP challenge data to the PSU device and adds same challenge code data to the JSON response of the *update PSU data for payment* request. PSU must confirm this challenge using PIN2 code. If the confirmation was successful payment transaction status will be changed from *PDNG (Pending)* to *ACSC (AcceptedSettlementCompleted)* and authorization object will be *finalized*. If the authorization is unsuccessful payment transaction status

will not change but authorization object status will be changed to failed. In this case TPP should start authorization process from the second step: *start the authorization process for a payment initiation*. More information about request and response structure could be found in the 58 page.

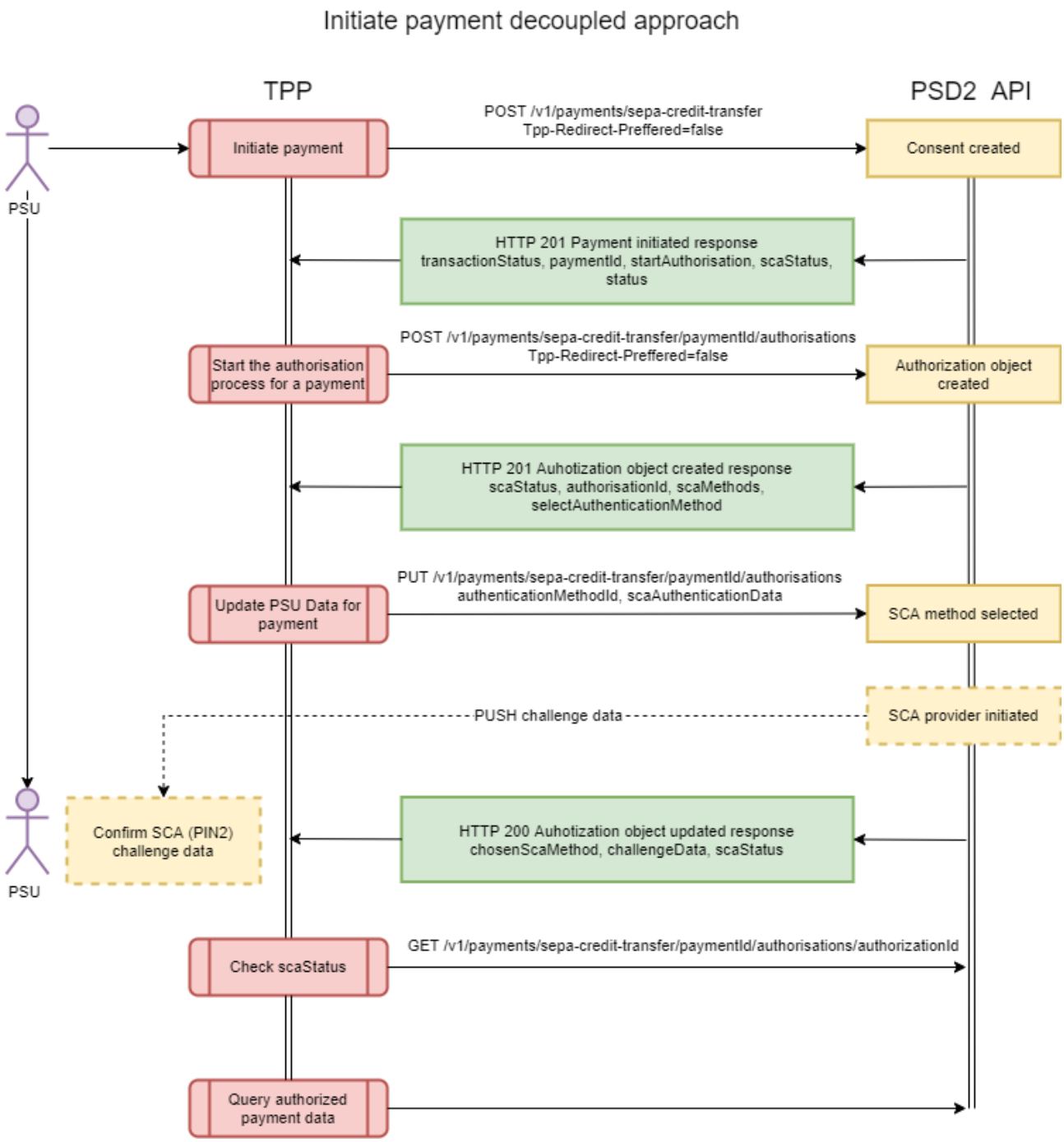

Figure 18. Initiate payment decoupled approach

Payment initiation

Request POST /v1/payments/{payment-product}

Path parameters

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
PSU-IP-Address	mandatory	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. If not available, the TPP shall use the IP Address used by the TPP when submitting this request.
TPP-Redirect-Preferred	optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach. If it equals "false", the TPP prefers not to be redirected for SCA. The ASPSP will then choose between the Embedded or the Decoupled SCA approach, depending on the choice of the SCA procedure by the TPP/PSU.
TPP-Nok-Redirect-URI	optional	If this URI is contained, the TPP is asking to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the redirect SCA method
TPP-Redirect-URI		URI of the TPP, where the transaction flow shall be redirected to after a Redirect. Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true".

Request body

endToEndIdentification	optional	SEPA end to end reference id field
debtorAccount	mandatory	Debtor account object with iban and currency elements
instructedAmount	mandatory	Instructed payment amount has amount and currency elements
creditorAccount	mandatory	Creditor account object with iban and currency elements
creditorAgent	optional	
creditorName	mandatory	Title/name of the creditor
creditorAddress	optional	
remittanceInformationUnstructured	optional	

Request example (remittance unstructured)


```
{
    "endToEndIdentification": "LT-1234567890",
    "debtorAccount": {
        "currency": "EUR",
        "iban": "LTXXXXXXXXXXXXXXXXXX"
    },
    "instructedAmount": {
        "amount": 0.36,
        "currency": "EUR"
    },
    "creditorAccount": {
        "currency": "EUR",
        "iban": "DEXXXXXXXXXXXXXXXXXX"
    },
    "creditorName": "Swedbank AB",
    "creditorAddress": {
        "buildingNumber": "25-96",
        "townName": "Kaunas",
        "country": "LT",
        "postCode": 90233,
        "streetName": "St 111"
    },
    "remittanceInformationUnstructured": "Paskirtis uz daikta 445566"
}
```


Request example (remittance structured)


```
{
    "endToEndIdentification": "LT-1234567890",
    "debtorAccount": {
        "iban": "LTXXXXXXXXXXXXXXXXXX",
        "currency": "EUR"
    },
    "instructedAmount": {
        "currency": "EUR",
        "amount": "0.36"
    },
    "creditorAccount": {
        "iban": "DEXXXXXXXXXXXXXXXXXX",
        "currency": "EUR"
    },
    "creditorName": "Swedbank AB",
    "creditorAddress": {
        "streetName": "St 111",
        "buildingNumber": "25-96",
        "townName": "Kaunas",
        "postCode": "90233",
        "country": "LT"
    },
    "remittanceInformationStructured": {
        "reference": "1001 testas"
    }
}
```

Response code

201 Created	The request has been fulfilled and has resulted in one or more new resources being created

Response header

Location	Location of the created resource (if created)
X-Request-ID	ID of the request, unique to the call, as determined by the initiating party
Aspsp-Sca-Approach	Possible values are: REDIRECT or DECOUPLED

Response example (TPP-Redirect-Preferred = true/false/null, TPP-Explicit-Authorisation-Preferred = true)


```
{
    "transactionStatus": "PDNG",
    "paymentId": "69258CDC1527AF1FF32EB0203B8619A62A35DE3B64ED4C929B3EC14A3BD44EEA",
    "transactionFeeIndicator": false,
    "_links": {
        "self": {
            "href": "/v1/payments/sepa-credit-transfers/
                69258CDC1527AF1FF32EB0203B8619A62A35DE3B64ED4C929B3EC14A3BD44EEA"
        },
        "startAuthorisation": {
            "href": "/v1/payments/sepa-credit-transfers/
                69258CDC1527AF1FF32EB0203B8619A62A35DE3B64ED4C929B3EC14A3BD44EEA/authorisations"
        },
        "status": {
            "href": "/v1/payments/sepa-credit-transfers/
                69258CDC1527AF1FF32EB0203B8619A62A35DE3B64ED4C929B3EC14A3BD44EEA/status"
        }
    }
}
```

Response example (TPP-Redirect-Preferred = true, TPP-Explicit-Authorisation-Preferred = false)


```
{
    "transactionStatus": "PDNG",
    "paymentId": "7FDE2B42D9687B7F2254D9FDBD72E8636CAC4D3AA4C323C0B8D9D8AE48820ACB",
    "transactionFeeIndicator": false,
    "_links": {
        "self": {
            "href": "/v1/payments/sepa-credit-transfers/
                7FDE2B42D9687B7F2254D9FDBD72E8636CAC4D3AA4C323C0B8D9D8AE48820ACB"
        },
        "scaStatus": {
            "href": "/v1/payments/sepa-credit-transfers/
                7FDE2B42D9687B7F2254D9FDBD72E8636CAC4D3AA4C323C0B8D9D8AE48820ACB/authorisations/
                885f4b72-1674-46de-8f1a-8f6d5596002b"
        },
        "scaRedirect": {
            "href": "https://psd2.i-unija.lt/payment/885f4b72-1674-46de-8f1a-8f6d5596002b/"
        },
        "status": {
            "href": "/v1/payments/sepa-credit-transfers/
                7FDE2B42D9687B7F2254D9FDBD72E8636CAC4D3AA4C323C0B8D9D8AE48820ACB/status"
        }
    }
}
```


Get payment transaction status

Request GET /v1/payments/{payment-product}/{payment-id}/status

Path parameters

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Accept	optional	JSON only supported

Response code

200 OK	**The request has succeeded**

Response header

X-Request-ID	**ID of the request, unique to the call, as determined by the initiating party**

Response example

```
{
    "transactionStatus": "PDNG"
}
```

Get payment request

Request GET /v1/payments/{payment-product}/{payment-id}

Path parameters

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token

Response code

200 OK	**The request has succeeded**

Response header

X-Request-ID	**ID of the request, unique to the call, as determined by the initiating party**

Response example (remittance unstructured)


```
{
    "endToEndIdentification": "LT-1234567890",
    "debtorAccount": {
        "iban": "LTXXXXXXXXXXXXXXXXXX",
        "currency": "EUR"
    },
    "instructedAmount": {
        "currency": "EUR",
        "amount": "0.36"
    },
    "creditorAccount": {
        "iban": "DEXXXXXXXXXXXXXXXXXX",
        "currency": "EUR"
    },
    "creditorName": "Swedbank AB",
    "creditorAddress": {
        "streetName": "St 111",
        "buildingNumber": "25-96",
        "townName": "Kaunas",
        "postCode": "90233",
        "country": "LT"
    },
    "remittanceInformationUnstructured": "Paskirtis uz daikta 445566",
    "transactionStatus": "PDNG"
}
```

Response example (remittance unstructured)


```
{
    "endToEndIdentification": "LT-1234567890",
    "debtorAccount": {
        "iban": "LTXXXXXXXXXXXXXXXXXXX",
        "currency": "EUR"
    },
    "instructedAmount": {
        "currency": "EUR",
        "amount": "0.36"
    },
    "creditorAccount": {
        "iban": "DEXXXXXXXXXXXXXXXXXXX",
        "currency": "EUR"
    },
    "creditorName": "Swedbank AB",
    "creditorAddress": {
        "streetName": "St 111",
        "buildingNumber": "25-96",
        "townName": "Kaunas",
        "postCode": "90233",
        "country": "LT"
    },
    "remittanceInformationStructured": {
        "reference": "1001 testas"
    },
    "transactionStatus": "PDNG"
}
```


Delete payment

Request DELETE /v1/ payments/{payment-product}/{payment-id}

Path parameters

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token

Response code

204 No content	The request has succeeded

Response header

X-Request-ID	The server has successfully fulfilled the request and that there is no additional content to send in the response payload body

Start the authorization process for a payment initiation

Request POST /v1/payments/{payment-product}/{payment-id}/authorisations

Path parameters

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
TPP-Redirect-Preferred	optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach.
TPP-Redirect-URI	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
TPP-Nok-Redirect-URI	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true". Needed for not ok redirect URI
Content-Type	optional	Content type application/json

Response code

201 Created	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party
Aspsp-Sca-Approach	Possible values are REDIRECT or DECOUPLED

Response example (TPP-Redirect-Preferred = false)


```
{
    "scaStatus": "received",
    "authorisationId": "9effc5fa-2439-4cc0-95a8-bdbd27d16d1a",
    "scaMethods": [
        {
            "authenticationType": "PUSH_OTP",
            "authenticationMethodId": "SmartId",
            "name": "Smart-ID",
            "explanation": "Norėdami prisijungti su Smart-ID turite atsisiųsti nemokamą programėlę į savo
                išmanųjį telefoną ar planšetinį kompiuterį."
        }
    ],
    "_links": {
        "selectAuthenticationMethod": {
            "href": "/v1/payments/sepa-credit-transfers/
                7FDE2B42D9687B7F2254D9FDBD72E8636CAC4D3AA4C323C0B8D9D8AE48820ACB/authorisations/
                9effc5fa-2439-4cc0-95a8-bdbd27d16d1a"
        },
        "scaStatus": {
            "href": "/v1/payments/sepa-credit-transfers/
                7FDE2B42D9687B7F2254D9FDBD72E8636CAC4D3AA4C323C0B8D9D8AE48820ACB/authorisations/
                9effc5fa-2439-4cc0-95a8-bdbd27d16d1a"
        }
    }
}
```


Response example (TPP-Redirect-Preferred = true, TPP-Redirect-URI=http://....)


```
{
    "scaStatus": "received",
    "authorisationId": "0be11929-b918-47a4-a210-884470d5c1be",
    "_links": {
        "scaRedirect": {
            "href": "https://psd2.i-unija.lt/payment/0be11929-b918-47a4-a210-884470d5c1be/"
        },
        "scaStatus": {
            "href": "/v1/payments/sepa-credit-transfers/
                7FDE2B42D9687B7F2254D9FDBD72E8636CAC4D3AA4C323C0B8D9D8AE48820ACB/authorisations/
                0be11929-b918-47a4-a210-884470d5c1be"
        }
    }
}
```

Update PSU data for payments (only for decoupled method)

Request PUT /v1/payments/{payment-product}/{payment-id}/authorisations/{authorisation-id}

Path parameters

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource
authorisation-id	Authorisation object ID

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Request body

authenticationMethodId	mandatory	Select authentication method from list provided by start authorisation process response

Request example

```
{
    "authenticationMethodId": "{{authentication-method-id}}"
}
```

Response code

200 OK	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party
Aspsp-Sca-Approach	Possible values are: REDIRECT or DECOUPLED

Response example


```
{
    "chosenScaMethod": {
        "authenticationMethodId": "SmartId"
    },
    "challengeData": {
        "data": [
            "8915"
        ],
        "otpFormat": "integer",
        "additionalInformation": "Smart-ID parašas"
    },
    "_links": {
        "self": {
            "href": "/v1/payments/sepa-credit-transfers/
                02027E45BAD0698E05B51F22786E66FE2A4AE213E8019F533305F8A7B3E32A4F"
        },
        "scaStatus": {
            "href": "/v1/payments/sepa-credit-transfers/
                02027E45BAD0698E05B51F22786E66FE2A4AE213E8019F533305F8A7B3E32A4F/authorisations/
                95002e3f-2bff-4b41-869e-3bdd221dd510"
        },
        "status": {
            "href": "/v1/payments/sepa-credit-transfers/
                02027E45BAD0698E05B51F22786E66FE2A4AE213E8019F533305F8A7B3E32A4F/status"
        }
    },
    "scaStatus": "started",
    "psuMessage": "Smart-ID parašas"
}
```


Read the SCA Status of the payment authorisation

Request GET /v1/payments/{payment-product}/{payment-id}/authorisations/{authorisation-id}

Path parameters

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource
authorisation-id	Authorisation object ID (in case of payment authorisation)

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token

Content-Type	optional	Content type application/json

Response code

200 OK	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example

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

Get Payment Authorisation Sub-Resources

Request GET /v1/payments/{payment-product}/{payment-id}/authorisations

Path parameters.

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Response code

200 OK	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example


```
{
    "authorisationIds": [
        "69e31e32-96cd-439d-9eb3-d964b7fc855c",
        "f0de36e2-1c7e-42c0-9927-4f9e7aa78edc",
        "ef338d0f-2488-4ded-810a-eb04464db5b7"
    ]
}
```


Start the authorization process for the cancellation of the addressed payment

Request POST /v1/payments/{payment-product}/{payment-id}/cancellation-authorisations

Path parameters

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
TPP-Redirect-Preferred	optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach.
TPP-Redirect-URI	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
TPP-Nok-Redirect-URI	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true". Needed for not ok redirect URI
Content-Type	optional	Content type application/json

Response code

201 Created	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party
Aspsp-Sca-Approach	Possible values are: REDIRECT or DECOUPLED

Response example (TPP-Redirect-Preferred = false)


```
{
    "scaStatus": "received",
    "authorisationId": "1f45d9e8-6c48-485e-8a7d-da96f7bca8b2",
    "scaMethods": [
        {
            "authenticationType": "PUSH_OTP",
            "authenticationMethodId": "SmartId",
            "name": "Smart-ID",
            "explanation": "Norėdami prisijungti su Smart-ID turite atsisiųsti nemokamą programėlę į savo
                išmanųjį telefoną ar planšetinį kompiuterį."
        }
    ],
    "_links": {
        "selectAuthenticationMethod": {
            "href": "/v1/payments/sepa-credit-transfers/
                2723ECBD45F0F5BED899EB3D9F52D546CD1A8B6F81F7EF3C1EAC0F884FD8A8E6/cancellation-authorisations/
                1f45d9e8-6c48-485e-8a7d-da96f7bca8b2"
        },
        "scaStatus": {
            "href": "/v1/payments/sepa-credit-transfers/
                2723ECBD45F0F5BED899EB3D9F52D546CD1A8B6F81F7EF3C1EAC0F884FD8A8E6/authorisations/
                1f45d9e8-6c48-485e-8a7d-da96f7bca8b2"
        }
    }
}
```


Response example (TPP-Redirect-Preferred = true, TPP-Redirect-URI=http://....)


```
{
    "scaStatus": "received",
    "authorisationId": "d2a3e63f-0dcd-46fb-a1dd-3ae64361be32",
    "_links": {
        "scaRedirect": {
            "href": "https://psd2.i-unija.lt/payment/cancellation/d2a3e63f-0dcd-46fb-a1dd-3ae64361be32/"
        },
        "scaStatus": {
            "href": "/v1/payments/sepa-credit-transfers/
                2723ECBD45F0F5BED899EB3D9F52D546CD1A8B6F81F7EF3C1EAC0F884FD8A8E6/authorisations/
                d2a3e63f-0dcd-46fb-a1dd-3ae64361be32"
        }
    }
}
```

Update PSU data for payment initiation cancellation (only for decoupled method)

Request PUT /v1/ payments/{payment-product}/{payment-id}/cancellation-authorisations/{authorisation-id}

Path parameters

payment-id	The payment identification assigned to the created resource
authorisation-id	Authorization object ID

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Request body

authenticationMethodId	mandatory	Select authentication method from list provided by start authorization process response

Request example

```
{
    "authenticationMethodId": "{{authentication-method-id}}"
}
```

Response code

200 OK	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party
Aspsp-Sca-Approach	Possible values are: REDIRECT or DECOUPLED

Response example


```
{
    "chosenScaMethod": {
        "authenticationMethodId": "SmartId"
    },
    "challengeData": {
        "data": [
            "5510"
        ],
        "otpFormat": "integer",
        "additionalInformation": "Smart-ID parašas"
    },
    "_links": {
        "self": {
            "href": "/v1/payments/sepa-credit-transfers/
                18E193CBB801020C6C70DED8BFC8A9814396914ADC7E69EB4C3833432A8D1F6F"
        },
        "scaStatus": {
            "href": "/v1/payments/sepa-credit-transfers/
                18E193CBB801020C6C70DED8BFC8A9814396914ADC7E69EB4C3833432A8D1F6F/cancellation-authorisations/
                4a2fa6bc-32f2-4157-8090-14b6b662f9d8"
        },
        "status": {
            "href": "/v1/payments/sepa-credit-transfers/
                18E193CBB801020C6C70DED8BFC8A9814396914ADC7E69EB4C3833432A8D1F6F/status"
        }
    },
    "scaStatus": "started",
    "psuMessage": "Smart-ID parašas"
}
```


Read the SCA Status of the payment cancellation authorisation

Request GET /v1/payments/{payment-product}/{payment-id}/cancellation-authorisations/{cancellation-id}

Path parameters

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource
cancellation-id	Authorisation object ID (in case of payment cancellation authorisation)

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party

Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Response code

200 OK	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example

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

Get Payment Authorisation Cancellation Sub-Resources

Request GET /v1/payments/{payment-product}/{payment-id}/cancellation-authorisations

Path parameters.

payment-product	The addressed payment product endpoint, e.g., for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers
payment-id	The payment identification assigned to the created resource

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Response code

200 OK	The request has been fulfilled and has resulted in one or more new resources being created

Response header

X-Request-ID	ID of the request, unique to the call, as determined by the initiating party

Response example


```
{
    "authorisationIds": [
        "69e31e32-96cd-439d-9eb3-d964b7fc855c",
        "f0de36e2-1c7e-42c0-9927-4f9e7aa78edc",
        "ef338d0f-2488-4ded-810a-eb04464db5b7"
    ]
}
```


3.6 PIISP endpoints

Confirmation of funds request

Request GET /v1/funds-confirmation

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token

Request body

cardNumber	optional	Card Number of the card issued by the PIISP.
account	mandatory	PSU's account number.
payee	optional	The merchant where the card is accepted as an information to the PSU.
instructedAmount	mandatory	Transaction amount to be checked within the funds check mechanism.

Request example

```
{
  "instructedAmount": {
    "amount": "10.01",
    "currency": "EUR"
  },
  "payee": "Check24",
  "account": {
    "bban": "1111111111",
    "currency": "EUR",
    "iban": "LTXXXXXXXXXXXXXXXXXX"
  }
}
```

Response code

200 OK	**The request has succeeded**

Response header

X-Request-ID	**ID of the request, unique to the call, as determined by the initiating party**

Response example

```
{
    "fundsAvailable": true
}
```

4. Extended PSD2 services

4.1 Recent beneficiaries

Request GET /v1/recent-beneficiaries

Query parameters

accountID	An IBAN of an account

Request header

X-Request-ID	mandatory	ID of the request, unique to the call, as determined by the initiating party
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	optional	Content type application/json

Response code

200 OK	**The request has succeeded**

Response header

X-Request-ID	**ID of the request, unique to the call, as determined by the initiating party**

Response example

```
{
    "recentBeneficiaries": [
        {
            "debtorAccount": {
                "iban": "LT045000000000000000"
            },
            "creditorAccount": {
                "iban": "LT560000000000000000"
            },
            "creditorName": "LCKU union"
        }
    ]
}
```

5. Additional info

5.1 Error codes

Global errors

HTTP code	TPP Code	TPP text	Description
400	SCA_METHOD_UNKNOWN	Unsupported authentication method	Unsupported authentication method for PSU.
401	SERVICE_BLOCKED	Unknown certificate	QWAC certificate revoked or invalid.
401	SERVICE_BLOCKED	Internet bank agreement is not active	User internet bank agreement is not yet active. As a result, internet bank cannot be used.
401	TOKEN_UNKNOWN	Unknown or expired access token	Access token expired. Refresh token grant should be executed.
401	TOKEN_INVALID	Unknown refresh token	Refresh token expired. User should relog in.
403	SERVICE_BLOCKED	Forbidden	User is authenticated but has no role to access resource.
404	RESOURCE_NOT_FOUND	The addressed resource not found	The addressed resource not found.
500	INTERNAL_ERROR	Internal error	Internal error occurred.

Authentication and authorization

HTTP code	TPP Code	TPP text	Description
400	FORMAT_ERROR	Unknown authorization	Invalid authorization id.
400	FORMAT_ERROR	Failed authorization	User initiated login but failed it.
400	FORMAT_ERROR	Expired authorization	User initiated login but failed to complete it in a time.
400	FORMAT_ERROR	Unknown authorization code	Authorization code was returned but system failed to exchange it to access token in a time.
401	PSU_CREDENTIALS_INVALID	PSU credentials invalid	User provided incorrect PSU ID or personal code.
403	SERVICE_BLOCKED	User blocked	User is blocked.
403	SERVICE_BLOCKED	Refused authorization	User initiated login but refused it in Smart ID / Mobile ID application.

Consents

HTTP code	TPP Code	TPP text	Description
400	CONSENT_UNKNOWN	Consent unknown	Consent was not found, incorrect consent id.
401	CONSENT_EXPIRED	Consent expired	Consent expired and new should be initiated.
401	CONSENT_INVALID	Consent invalid	Consent was invalidated.

Accounts

HTTP code	TPP Code	TPP text	Description
400	FORMAT_ERROR	Missing bookingStatus	Missing *bookingStatus* query parameter.
400	FORMAT_ERROR	Invalid bookingStatus value	*bookingStatus* query parameter value is not *both*, *pending*, *booked* or *information*.
400	FORMAT_ERROR	size should be a positive number	*size* query parameter value Is zero or

			negative.
400	FORMAT_ERROR	page should be a positive number or zero	*page* query parameter value is negative.
400	PERIOD_INVALID	dateFrom must be a past or present date	*dateFrom* query parameter value is from future.
400	PERIOD_INVALID	dateTo must be a past or present date	*dateTo* query parameter value is from future.
400	PERIOD_INVALID	dateFrom must be earlier or equal to dateTo	*dateFrom* query parameter value is after *dateTo* parameter value.
400	PERIOD_INVALID	No older than 90 days transactions are available	*dateFrom* query parameter value signals about access to transactions older than 90 days.

Payments

HTTP code	TPP Code	TPP text	Description
400	FORMAT_ERROR	Unknown payment	Invalid payment id
400	FORMAT_ERROR	Unknown authorization	Invalid payment authorisation id.
400	PAYMENT_FAILED	Failed authorization	User initiated payment authorisation but failed it.
400	PAYMENT_FAILED	Expired authorization	User initiated payment authorisation but failed to complete it in a time.
400	PAYMENT_FAILED	Refused authorization	User initiated payment authorisation but refused it in Smart ID / Mobile ID application
400	PAYMENT_FAILED	Insufficient funds for paying	Insufficient funds.
400	PAYMENT_FAILED	Debtor account number invalid or missing	Missing debtor account number or it is invalid.
400	PAYMENT_FAILED	Creditor account number invalid or missing	Missing creditor account number or it is invalid.
400	PAYMENT_FAILED	Creditor name is invalid or missing	Missing creditor name or it is invalid.
400	PAYMENT_FAILED	Debtor account number closed	Payment being initiated from closed debtor account.
400	PAYMENT_FAILED	Creditor account number closed	Payment being initiated from closed creditor account.
400	PAYMENT_FAILED	Debtor account blocked	Payment being initiated from blocked debtor account.
400	PAYMENT_FAILED	Debtor account currency is invalid or missing	Missing debtor account currency.
400	PAYMENT_FAILED	Creditor account currency is invalid or missing	Missing creditor account currency.
400	PAYMENT_FAILED	Payment forbidden on this type of account	Unsupported payment type being initiated using specified debtor account.
400	PAYMENT_FAILED	Payment amount exceeds single transfer limit	Payment request amount exceeds maximum single transfer limit.
400	PAYMENT_FAILED	Payment amount exceeds daily transfer limit	Payment request amount exceeds maximum daily transfer limit.
400	PAYMENT_FAILED	Creditor account cannot match Debtor account	Creditor and debtor account numbers matches.
400	PAYMENT_FAILED	Creditor bank is not SEPA reachable	Payment being initiated to creditor's bank that does not support SEPA.
400	PAYMENT_FAILED	Structured or unstructured remittance information required	Structured or unstructured remittance information required.
400	PAYMENT_FAILED	Remittance information structure does not comply with rules for payment type	Structured or unstructured remittance information does not comply with rules.
400	PAYMENT_FAILED	Unable to process payment	Unable to process payment due to unknown reason.
403	SERVICE_BLOCKED	Incomplete KYC	KYC is missing or incomplete.