

---

## *OTP PSD2 API production* documentation

---

Document version 1.4
----------------------

Date: 02.09.2024.
-------------------

## Document History

Version	Date	Description
1.0	14.01.2020	Initial version
1.2	02.06.2023.	PSU-ID flow added
1.3	03.06.2024.	Added cert renewal instructions
1.4	02.09.2024.	Payment without debtor account [chapter 7.7]

## Contents

- Document History.....2**
- 1. Definitions.....5**
- 2. Introduction .....5**
  - 2.1. Standards .....5
- 3. Production environment.....6**
  - 3.1. Introduction .....6
  - 3.2. Registration.....6
  - 3.3. Certificate renewal.....9
- 4. XS2A interface.....9**
  - 4.1. Pre-Authentication.....9
  - 4.2. Terms used in OAuth 2.0 .....11
  - 4.3. Getting OAuth token .....11
- 5. Strong Customer Authentication (SCA).....12**
  - 5.1. SCA Exemptions .....13
  - 5.2. Redirect approach.....13
    - 5.2.1. Redirect SCA Approach: Implicit Start of the Authorisation Process .....13
    - 5.2.2. Redirect SCA Approach: Explicit Start of the Authorisation Process .....14
- 6. Accounts endpoints .....15**
  - 6.1 Consent request.....15
    - 6.1.1 Consent request (using pre-step auth) .....16

6.1.2	Consent request PSU-ID flow for retail customers .....	17
6.1.3	Consent request PSU-ID flow for corporate customer .....	18
6.2	Get consent request.....	19
6.2.1	Get consent request (pre-auth flow) .....	19
6.2.2	Get consent request (PSU-ID flow for retail customers) .....	20
6.2.3	Get consent request (PSU-ID flow for corporate customers).....	21
6.3	Get consent status request .....	22
6.3.1	Get consent status (pre-auth) .....	22
6.3.2	Get consent status retail (PSU-ID flow retail customer) .....	23
6.3.3	Get consent status corporate (PSU-ID flow for corporate customer) .....	24
6.4	Delete consent.....	24
6.4.1	Delete consent (pre-auth flow) .....	25
6.4.2	Delete consent retail (PSU-ID flow) .....	25
6.4.3	Delete consent corporate (PSU-ID flow).....	26
6.5	Read account list.....	26
6.5.1	Read account list (pre-step flow).....	27
6.5.2	Read account list retail (PSU-ID flow retail customer) .....	28
6.5.3	Read account list corporate (PSU-ID flow corporate customer) .....	29
6.6	Read account details .....	30
6.7	Get balances .....	31
6.8	Get transactions list .....	32
6.9	Get transactions details .....	33
6.10	Start the authorization process for a consent (pre-step flow) .....	34
6.11	Start the authorization process for a consent (PSU flow retail customer) .....	34
6.12	Start the authorization process for a consent(PSU flow corporate customer) .....	35
6.13	Consent authorisation using Strong Customer Authentication (SCA) .....	36
6.13.1	Consent authorisations: redirect SCA approach .....	36
<b>7.</b>	<b>Payments endpoints .....</b>	<b>40</b>
7.1.	Payments initiation .....	40
7.2.	Get payment transaction status .....	41
7.3.	Get payment request .....	42
7.4.	Delete payment request .....	43

- 7.5. Update PSU data for payment initiation.....44
- 7.6. Payment authorisation using Strong Customer Authentication (SCA) .....45
  - 7.6.1. Payment authorisation: redirect SCA approach.....45
- 7.7 Payments without debtor account.....45
- 8. Miscellaneous .....46**

## 1. Definitions

This section offers explanations to the terminology used throughout the document.

<b>TPP (Third Party Provider)</b>	Third Party Provider (TPP) is the provider of an application which the user uses and is not offered by the bank. TPP is the client/consumer of the API and acts on behalf of the user through <b>consent</b> .
<b>PSU (Payment Service User) or user</b>	The user refers to the bank customer who uses the TPP application.
<b>ASPSP (Account Servicing Payments Service Provider)</b>	This is the account servicing provider, i.e, the OTP bank.
<b>PISP (Payment Initiation Service Provider)</b>	This is a service provider who can initiate a payment transaction on behalf of the customer.
<b>Sandbox</b>	Sandbox gives access to a small set of static data and it is used as an example to illustrate what would be returned when using the production API. The Sandbox can be reached within the developer portal at <a href="https://apiportal.sandbox.otpbanka.hr/portal/">https://apiportal.sandbox.otpbanka.hr/portal/</a>
<b>SCA (Strong Customer Authentication)</b>	The process of using a strong (2-factor) identification method to identify the customer.
<b>Authentication</b>	Authentication is the process of verifying that an individual is who it claims to be. This authentication is later used to grant authorization to specific data and functions within a system.
<b>Consent</b>	Consent is the agreement given by the PSU to the TPP to share data from the bank. Consent is stored by the bank and validated by the user according to PSD2. The consent may have a duration or just be used for a single API call.

## 2. Introduction

This document describes how TPPs can connect the PSD2 Solution of OTP sandbox API.

### 2.1. Standards

OTP bank PSD2 API follows standards described in the Berlin Group standard version 1.3.

From multiple options described in BGS we have selected to implement the following:

- Pre Step OAuth authorisation mode. It requires an authentication of a PSU in a pre-step, translating this authentication into an access token. Access token is mandatory for any other API call as described in BGS (4.3 Optional Usage of OAuth2 for PSU Authentication or Authorisation).
- OTP bank offers a redirect and decoupled integration methods as main way of integration for the TPP and the PSU.

The exposure of data is done through RESTful services. For the most part API encodes data in JavaScript Object Notation (**JSON**) format. In some cases **XML** may be used.

The API request and responses must use a UTF-8 character encoding, as is the default for JSON.

## 3. Production environment

### 3.1. Introduction

### 3.2. Registration

In order to use PSD2 services exposed by the bank, TPP needs to make a request to the specific endpoint in order to register itself and to get credentials that are needed for OAuth2 SCA. Endpoint that is used for TPP application registration is: POST /connect/register.

The payload of this request must be in JSON format and must contain following fields:

New users must fill in the following fields:

- **Redirect URIs** (*redirect\_uris*)  
*Required, list of URIs that TPP wants to register for redirection after successful completion of OAuth2 flow*
- **Post Logout Redirect URIs** (*post\_logout\_redirect\_uris*)  
*Optional, list of URIs that TPP wants to register for redirection after user logs out from the IAM application*
- **Logo URI** (*logo\_uri*)  
*Optional, URI to client logo*
- **Front Channel Logout URI** (*front\_channel\_logout\_uri*)  
*Specifies logout URI at client for HTTP based front-channel logout*
- **Back Channel Logout URI** (*back\_channel\_logout\_uri*)  
*Specifies logout URI at client for HTTP based back-channel logout*
- **Client URI** (*client\_uri*)  
*Optional, URI to further information about TPP*

Example payload:

```
{
  "post_logout_redirect_uris": [
    "https://www.getpostman.com/oauth2/callback"
  ],
  "client_uri": "https://www.uri.com",
  "logo_uri": "https://www.uri.com",
  "redirect_uris": [
    "https://www.getpostman.com/oauth2/callback"
  ]
}
```

In order to successfully perform Mutual TLS with the IAM application, TPP needs to provide X509 Certificate for authentication and to sign requests using private key that is associated with the public key from used certificates. To achieve this in Postman go to File->Settings. In new window click on Certificates tab. There is a button called Add Certificate under this tab.

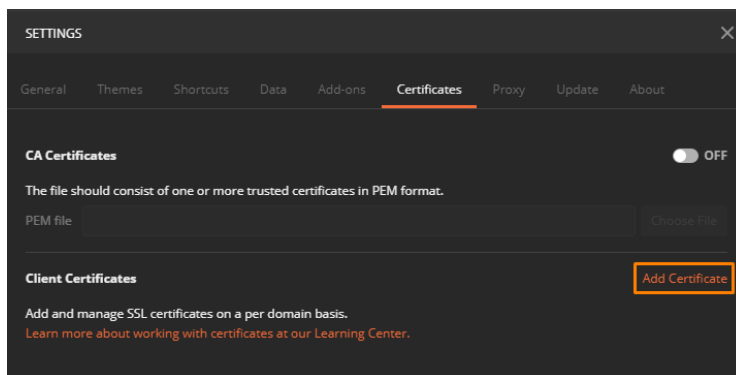


Figure 1 Adding certificate for Mutual TLS

Clicking on this button will open new window. In this window you need to fill in following fields:

- **Host**  
*Required, base path to the IAM application*
- **CRT file**  
*Path to the file that contains X509 Certificate in PEM format*
- **KEY file**  
*Path to the file that contains Private Key in PEM format*
- **PFX file**  
*Path to the file that contains both X509 Certificate and Private Key in PFX format*
- **Passphrase**  
*Passphrase for opening PFX file*

TPPs that have CRT and KEY files should not use **PFX file** and **Passphrase** fields, also, TPPs that have certificate in **PFX** format should not use **CRT file** and **KEY file** fields.

If the request was successful, TPP will get a response that looks similar to this example:

```
{
  "client_id": "63.certificate",
  "client_secret": "Certificate thumbprint",
  "client_name": "63 Certificate Client",
  "grant_types": "authorization_code,password,client_credentials",
  "scope": "PSD2 PIS:<paymentId> AIS:<consentId>",
  "client_uri": "https://www.uri.com",
  "logo_uri": "https://www.uri.com",
  "redirect_uris": [
    "https://www.getpostman.com/oauth2/callback"
  ],
  "post_logout_redirect_uris": [
    "https://www.getpostman.com/oauth2/callback"
  ],
  "front_channel_logout_uri": null,
  "back_channel_logout_uri": null
}
```

Figure 2 Registration response

This response contains data that will be needed later for starting the OAuth2 flow.

Response contains following fields:

- **Client Id**  
*Id of client that was created for TPP during registration*
- **Client Secret**  
*Secret for the created client. If this field has value "Certificate Thumbprint" that means that secret for the created client is thumbprint from certificate that was used for TPP registration*
- **Client Name**  
*Friendly client name*
- **Grant Types**  
*Allowed grant types*
- **Scope**  
*Allowed scopes*
- **Client URI**
- **Logo URI**
- **Redirect URIs**
- **Post Logout Redirect URIs**
- **Front Channel Logout URI**
- **Back Channel Logout URI**



## 3.3 Certificate renewal

### Renewed certificate:

If TPP certificate has expired, TPP should renew it and then make a new request with a new certificate, as described previously, to an endpoint: POST `{{lamBaseUrl}}/connect/register`. If a new certificate is attached and valid and if the Issuer and Subject match to previously registered one, TPP certificate will be updated in our system.

### New certificate:

If the Issuer or Subject of the certificate has changed, you should complete full registration, as previously described. It will be considered as a new `client_id`.

## 4. XS2A interface

### 4.1. Pre-Authentication

We will follow the pre-authentication approach of Berlin Group. It requires an authentication of a PSU in a pre-step, translating this authentication into an access token. This corresponds to the regular behaviour of OTP online banking. Access token is mandatory for any other API call as described in BGS (4.3 Optional Usage of OAuth2 for PSU Authentication or Authorisation). Please note that supported scope for pre-step authentication is **OTP.PSD2** in a Sandbox environment and in a production environment.

After successful registration the TPP can request AIS, PIS or PIIS services. The information about authorization server can be accessed via <https://iam.otpbanka.hr/.well-known/openid-configuration> link.

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 3.2. After registering application client will get *clientId* (Figure 2). These param should be passed to the Oauth server's `/authorization` and `/token` endpoints.

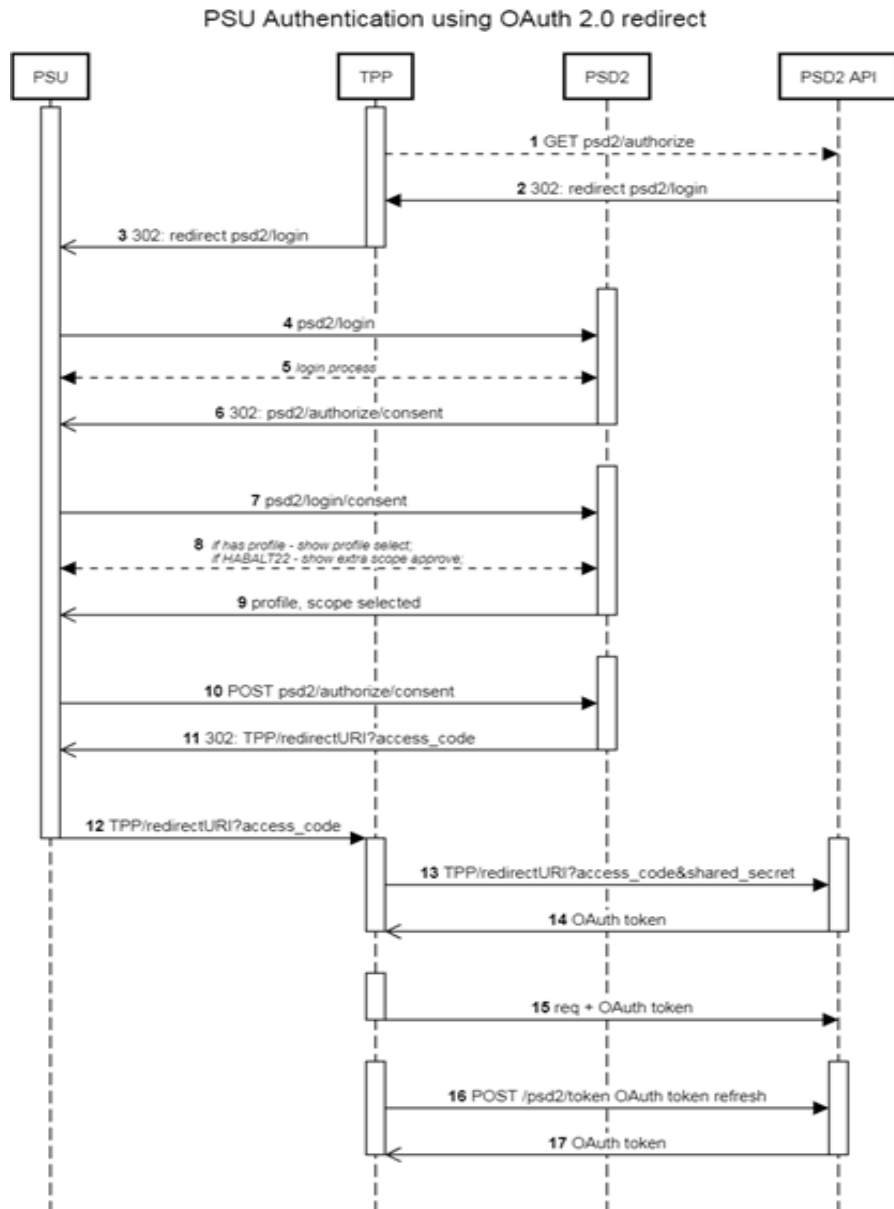


Figure 3 PSU Authentication using OAuth 2.0 redirect

## 4.2. Terms used in OAuth 2.0

Definition	Description
<b>OAuth 2.0</b>	OAuth 2.0 is an authorisation framework that enables applications to obtain limited access to user accounts on an HTTP service.
<b>OAuth 2.0 flow</b>	Since the API is built for backend communication and not designed to handle direct client communication it is mandated that the <code>grant_type</code> will be <code>authorization_code</code> and will require the use of a <code>client_secret</code> and a <code>client_id</code> .
<b>Client_secret</b>	A client secret is obtained by registering in the OTP Banking PSD2 API(Figure 2)
<b>Client_id</b>	A client Id is obtained by registering in the OTP Banking PSD2 API(Figure 2)
<b>Scope</b>	A scope according to OAuth 2.0 defines what data can be accessed. API supported scopes are PIS,AIS.
<b>Redirect_uri</b>	The <code>redirect_uri</code> is an address used by OAuth providers as a location to deliver the <code>access_token</code> by means of a browser redirect.
<b>State</b>	In OAuth 2.0 a random string is defined as state. State is provided and verified by TPP. The main purpose is to avoid some cross site request forgery (CSRF) attacks.

## 4.3. Getting OAuth token

OAuth2 tokens are issued requesting authorization server's special endpoints. Server could be accessed via <https://iam.otpbanka.hr> 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 token issuing.

### Authorization endpoint GET /connect/authorize

<code>response_type</code>	mandatory	„code“ is only supported as response type
<code>grant_type</code>	mandatory	„code“ is only supported as grant type
<code>client_id</code>	mandatory	Generated application clientid
<code>scope</code>	mandatory	Scope should be „OTP.PSD2“
<code>state</code>	mandatory	A dynamical value set by the TPP and used to prevent XSRF attacks.
<code>redirect_uri</code>	mandatory	the URI of the TPP where the OAuth2server is redirecting the PSU's user agent after the authorization.

### CURL authorization call:

```
curl --location --request GET
'https://iam.otpbanka.hr/connect/authorize?client_id=<client_id>&redirect_uri=<redirec
t_uri>&response_type=code&grant_type=code&scope=openid&state=<state>&scope=<scope>'
```

Executing this call will generate 302 response with Location header, redirect the client app to the login form where the user has to authorize himself.

After successful user authorization client app will be redirected to the `redirect_uri` parameter with following parameters in string format.

## OTP Identity Access Manager: TPP User Guide

code	one time code that will be used for obtaining access token by TPP
state	A dynamical value set by the TPP and used to prevent XSRF attacks.
scope	scopes that were granted
session_state	this field can be omitted

After this step TPP can request token by calling token issuing endpoint.

### Token endpoint POST /connect/token/mtls

code	mandatory	code that was received in callback
client_id	mandatory	Generated application clientId from developer's portal
scope	mandatory	This field should be equal to the scope parameter received in callback
grant_type	mandatory	This field needs to be equal to „authorization_code“
redirect_uri	mandatory	redirect URI that was used I /connect/authorize request

#### CURL token call:

```
curl --location --request POST 'https://iam.otpbanka.hr/connect/token/mtls' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=<client_id>' \
--data-urlencode 'scope=<scope>' \
--data-urlencode 'grant_type=authorization_code' \
--data-urlencode 'code=<code>' \
--data-urlencode 'redirect_uri=<redirect_uri>' \
--key client.key \
--cert client.crt \
```

As in guide for TPP application registration, TPP should add certificate that will be used for Mutual TLS.

#### Token response example

```
{
  "access_token":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG91IiwiaWF0IjoxNTE2MzIyMDIyLjE1fQ.Sf1KxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQs sw5c",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```

## 5. Strong Customer Authentication (SCA)

By PSD2 directive some API calls requires Strong Customer Authentication (PSU must approve request by using PIN2). SCA can be implemented either by using redirect or decoupled methods.

In redirect mode PSD2 API generates links and TPP must redirect PSU to these pages. In these pages corresponding payment/consent/... information will be displayed to PSU and PSU will authorise using security device and PIN2.

In decoupled integration mode SCA is performed without displaying bank pages. In decoupled mode PSD2 API generates SCA requests to third party identity provider, PSU gets details of requested authorisation action in his security device and approve it using PIN2.

### 5.1. SCA Exemptions

Whitelisting, low-value transaction are also not supported in the first approach of the XS2A interface.

### 5.2. Redirect approach

In redirect integration method TPP needs to provide redirects URLs in the TPP-Redirect-URI header in case when SCA is required. Redirects happen after the PSU has completed the SCA process in OTP PSD2 API pages and have to be redirected to TPP. Please note that TPP-Redirect-URI header is required for all SCA requests in redirect integration method.

#### 5.2.1. Redirect SCA Approach: Implicit Start of the Authorisation Process

If request requires PSU SCA, ASPSP might start the authorisation process **implicitly** (Figure 4) in case of no additional data is needed from the TPP or if TPP-explicit-authorisation-preferred is omitted or false. In response `_links` element `scaRedirect` link is returned. Redirect the PSU using `scaRedirect` to OTP bank environment, where PSU completes SCA flow. Please note that TPP-Redirect-URI header is required for all SCA requests in redirect integration method.

After successfully completing the SCA flow the PSU will be redirected to the URL provided earlier in the TPP-Redirect-URI with query parameter `confirmationCode`. An authorisation confirmation request is requested by the OTP and after the session is re-directed to the TPP's system.

If authorisation has failed, new authorisation may be created and processed.

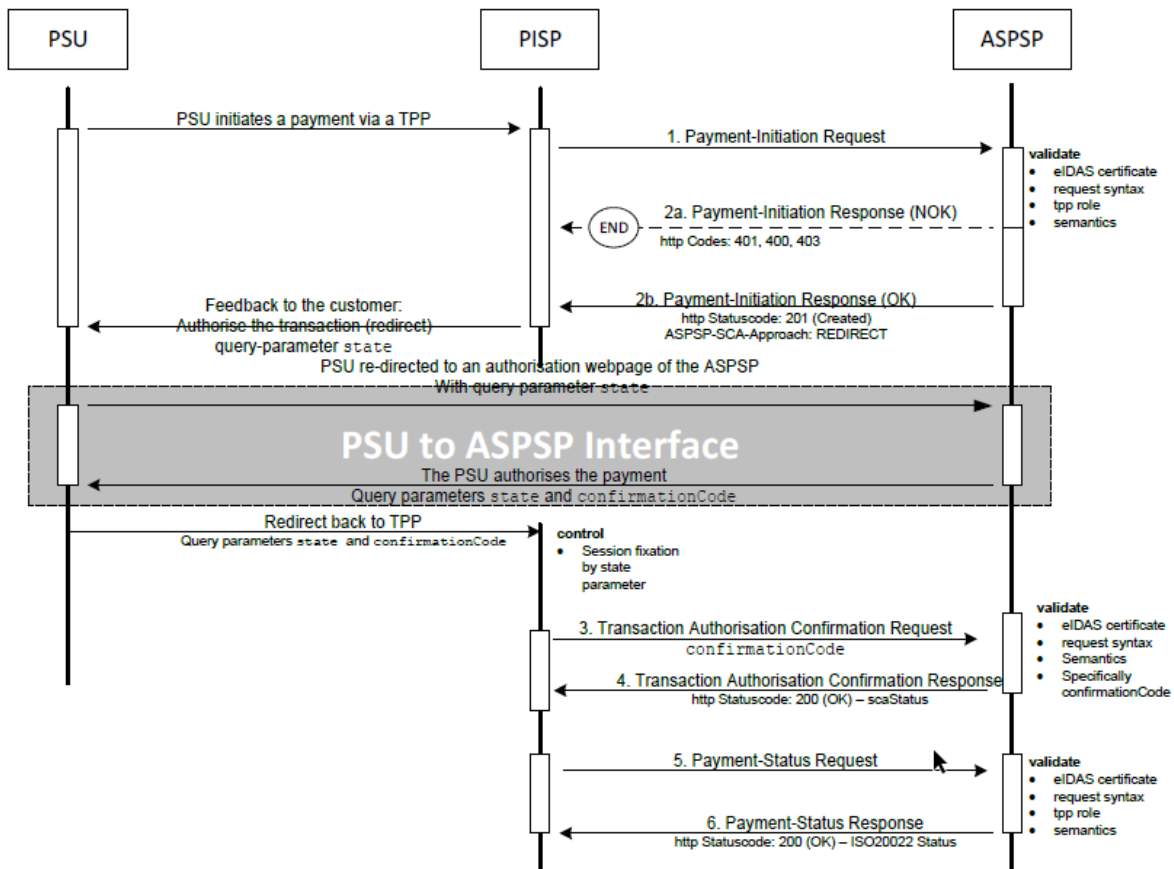


Figure 4 Redirect SCA Approach: Implicit Start of the Authorisation Process

### 5.2.2. Redirect SCA Approach: Explicit Start of the Authorisation Process

If TPP-explicit-authorization-preferred is set to true, **explicit** authorisation is started. Explicit authorisation allows more detailed control of authorisation process needed for decoupled integration method or countersigning. It is not advised for redirect integration. In such case response will have steering link in *startAuthorisation* parameter and TPP must request it. This will start authorisation and return scaRedirect link. More technical details are available on Developer Portal. Explicit authorisation should only be used when decoupled approach is selected or when countersigning of the payment is required.

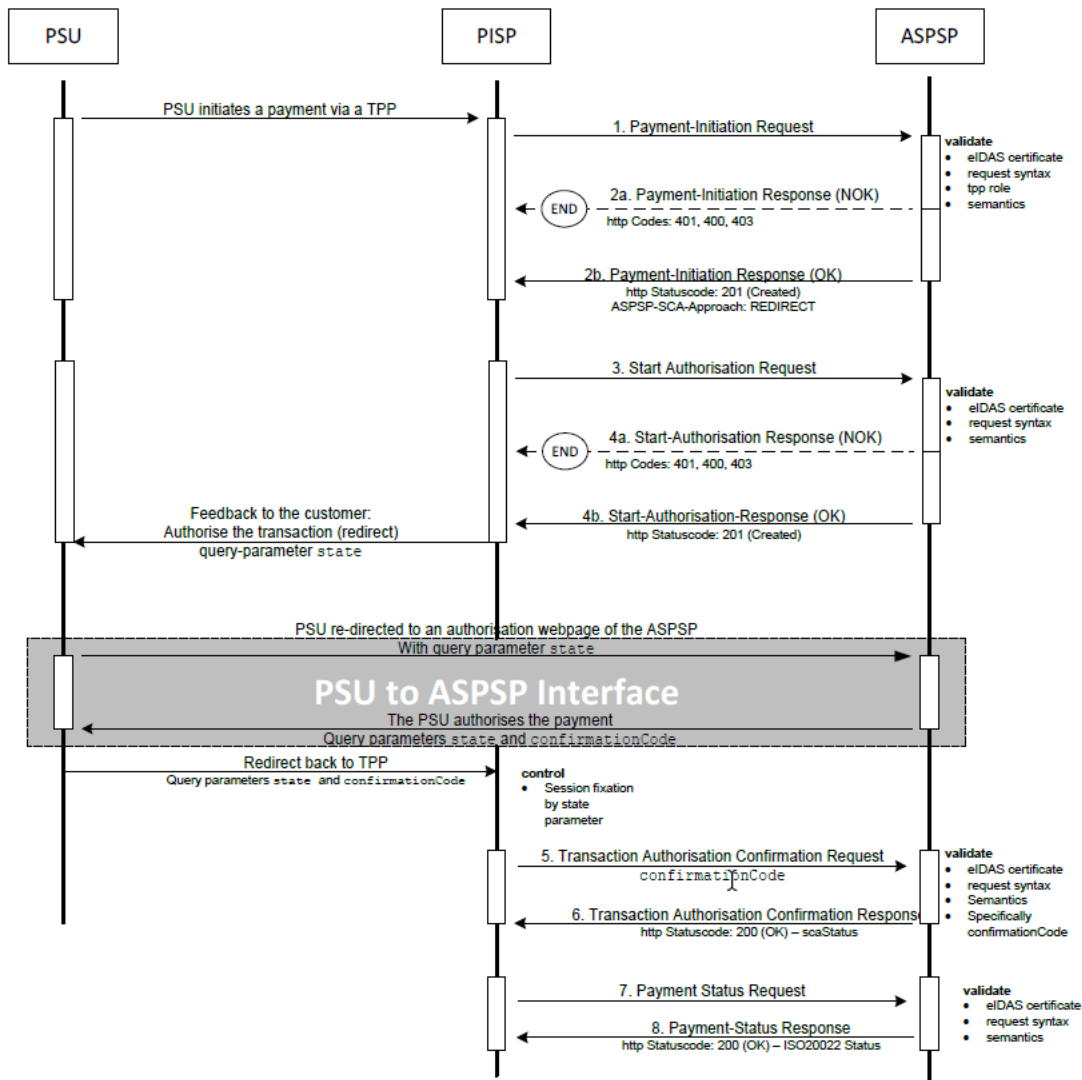


Figure 5 Redirect SCA Approach: Explicit Start of the Authorisation Process

## 6. Accounts endpoints

### 6.1 Consent request

In order to read account details, transactions, balances or initiate payments, TPP needs to get consent from user. An AI role is needed for accessing this endpoint. First step in doing this is creation of consent resource. For creating a consent, a SCA will always be necessary. Consent can be created two-ways: using pre-step auth or PSU-ID flow. We'll explain both scenarios:

## 6.1.1 Consent request (using pre-step auth)

Request is done by sending JWT bearer token in header parameters.

### Request POST /v1/consents/

#### Request header

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>Authorization</b>	mandatory	Oauth2 authorization bearer token
<b>TPP-Redirect-Preferred</b>	optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach.
<b>TPP-Redirect-URI</b>	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
<b>Content-Type</b>	mandatory	Content type application/json

#### Request body

<b>access</b>	mandatory	Requested access services
<b>recurringIndicator</b>	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
<b>validUntil</b>	mandatory	This parameter is requesting a valid until date for the requested consent.
<b>frequencyPerDay</b>	mandatory	This field indicates the requested maximum frequency for an access without PSU involvement per day.
<b>combinedServiceIndicator</b>	mandatory	If true indicates that a payment initiation service will be addressed in the same "session",

#### Request example

```
{
  "access": {
    "availableAccounts": "allAccounts"
  },
  "recurringIndicator": "false",
  "validUntil": "2019-12-30T10:02:29.073Z",
  "frequencyPerDay": "30",
  "combinedServiceIndicator": "false"
}
```

### Response POST /v1/consents/

#### Response code

<b>201 Created</b>	The request has been fulfilled and has resulted in one or more new resources being created
--------------------	--

#### Response header



<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Aspsp-Sca-Approach</b>	Possible values are: REDIRECT or DECOUPLED
<b>Content-Type</b>	Content type application/json

**Response example**

```

{
  "consentStatus": "received",
  "consentId": "58f686ea-cd47-4501-92e1-eae26398bbee",
  "scaMethods": [],
  "chosenScaMethod": null,
  "challengeData": {
    "data": "Default challenge"
  },
  "_links": {
    "scaRedirect": {
      "href": ""
    },
    "self": {
      "href": "v1/consents/58f686ea-cd47-4501-92e1-eae26398bbee"
    },
    "status": {
      "href": "v1/consents/58f686ea-cd47-4501-92e1-eae26398bbee/status"
    },
    "scaStatus": {
      "href": "v1/consents/58f686ea-cd47-4501-92e1-eae26398bbee/authorisations/bff84163af3e43b98e6c3d8ea49b1326"
    }
  }
}

```

6.1.2 Consent request PSU-ID flow for retail customers

We need to set PSU-ID variable in headers. PSU-ID contains token serial number.

**Request POST /v1/consents/**

**Request header**

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	mandatory	Token SN from retail customer
<b>TPP-Redirect-Preferred</b>	optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach.
<b>TPP-Redirect-URI</b>	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
<b>Content-Type</b>	mandatory	Content type application/json

**Request body**

<b>access</b>	mandatory	Requested access services
<b>recurringIndicator</b>	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
<b>validUntil</b>	mandatory	This parameter is requesting a valid until date for the requested consent.
<b>frequencyPerDay</b>	mandatory	This field indicates the requested maximum frequency for an access without PSU involvement per day.
<b>combinedServiceIndicator</b>	mandatory	If true indicates that a payment initiation service will be addressed in the same "session",

## Request example

```
curl --location 'https://api.test.otpbanka.hr/v1/consents' \
--header 'Content-Type: application/json' \
--header 'X-Request-ID: 260ff652-69b3-4429-99fe-182572b5d58b' \
--header 'PSU-IP-Address: 10.1.2.3' \
--header 'TPP-Explicit-Authorisation-Preferred: true' \
--header 'Tpp-Redirect-Preffered: true' \
--header 'TPP-Redirect-URI: https://api.test.psd2cloud.asseco-see.hr' \
--header 'PSU-ID: <TOKEN_SN_NUMBER >' \
--data '{
  "access": {
    "availableAccounts": "allAccounts"
  },
  "recurringIndicator": "TRUE",
  "validUntil": "2023-06-30",
  "frequencyPerDay": "4",
  "combinedServiceIndicator": "false"
}'
```

### 6.1.3 Consent request PSU-ID flow for corporate customer

#### Request POST /v1/consents/

##### Request header

Header	Requirement	Description
<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	mandatory	a. For standard token users: <b>Token SN - corporate customer</b> b. For PKI USB stick or PKI smart card users: <b>Personal identification number (OIB) from the person - owner of the USB stick</b>
<b>PSU-CORPORATE-ID</b>	mandatory	Corporate customer OIB (Croatian VAT number)
<b>TPP-Redirect-Preferred</b>	optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach.
<b>TPP-Redirect-URI</b>	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
<b>Content-Type</b>	mandatory	Content type application/json

##### Request body

Field	Requirement	Description
<b>access</b>	mandatory	Requested access services
<b>recurringIndicator</b>	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
<b>validUntil</b>	mandatory	This parameter is requesting a valid until date for the requested consent.
<b>frequencyPerDay</b>	mandatory	This field indicates the requested maximum frequency for an access without PSU involvement per day.
<b>combinedServiceIndicator</b>	mandatory	If true indicates that a payment initiation service will be addressed in the same "session",

## Request example

```
curl --location 'https://api.test.otpbanka.hr/v1/consents' \
--header 'Content-Type: application/json' \
--header 'X-Request-ID: 260ff652-69b3-4429-99fe-182572b5d58b' \
--header 'PSU-IP-Address: 10.1.2.3' \
--header 'TPP-Explicit-Authorisation-Preferred: true' \
--header 'Tpp-Redirect-Preferred: true' \
--header 'TPP-Redirect-URI: https://api.test.psd2cloud.asseco-see.hr' \
--header 'PSU-ID: <TOKEN_SN_NUMBERE_HERE>' \
--header 'PSU-ID-CORPORATE: <CORPORATE CUSTOMER OIB>' \

--data '{
  "access": {
    "availableAccounts": "allAccounts"
  },
  "recurringIndicator": "TRUE",
  "validUntil": "2023-06-30",
  "frequencyPerDay": "4",
  "combinedServiceIndicator": "false"
}'
```

## 6.2 Get consent request

Returns the content of an account information consent object.

### 6.2.1 Get consent request (pre-auth flow)

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

**Path parameter**

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

**Request header**

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

**Response GET /v1/consents/{consentId}**

**Response code**

<b>200 Ok</b>	The request has succeeded
---------------	---------------------------

**Response header**

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

**Response example**

```
{
  "access": {
    "accounts": [],
    "balances": [],
    "transactions": [],
    "availableAccounts": "allAccounts"
  },
  "recurringIndicator": true,
  "validUntil": "2020-01-30T10:02:29.073",
  "frequencyPerDay": 30,
  "lastActionDate": "2020-01-17T12:53:39.6005658",
  "consentStatus": "received"
}
```

6.2.2 Get consent request (PSU-ID flow for retail customers)

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

**Path parameter**

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

**Request header**

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	Mandatory	Token SN from retail customer
<b>Content-Type</b>	mandatory	Content type application/json

## Response GET /v1/consents/{consentId}

### 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
Content-Type	Content type application/json

### Response example

```
{
  "access": {
    "accounts": [],
    "balances": [],
    "transactions": [],
    "availableAccounts": "allAccounts"
  },
  "recurringIndicator": true,
  "validUntil": "2023-06-30",
  "frequencyPerDay": 4,
  "lastActionDate": "2023-05-31",
  "consentStatus": "received"
}
```

## 6.2.3 Get consent request (PSU-ID flow for corporate customers)

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

#### Path parameter

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
PSU-ID	Mandatory	<ol style="list-style-type: none"> <li>For standard token users: <b>Token SN - corporate customer</b></li> <li>For PKI USB stick or PKI smart card users: <b>Personal identification number (OIB) from the person - owner of the USB stick</b></li> </ol>
PSU-CORPORATE-ID	mandatory	Corporate customer OIB (Croatian VAT number)
Content-Type	mandatory	Content type application/json

### Response GET /v1/consents/{consentId}

## 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
Content-Type	Content type application/json

## Response example

```
{
  "access": {
    "accounts": [],
    "balances": [],
    "transactions": [],
    "availableAccounts": "allAccounts"
  },
  "recurringIndicator": true,
  "validUntil": "2023-06-30",
  "frequencyPerDay": 4,
  "lastActionDate": "2023-05-31",
  "consentStatus": "received"
}
```

## 6.3 Get consent status request

Returns the content of an account information consent object.

### 6.3.1 Get consent status (pre-auth)

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

#### Path parameter

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
--------------	-----------	--

<b>Authorization</b>	Mandatory	Oauth2 authorization bearer token
<b>Content-Type</b>	mandatory	Content type application/json

**Response GET /v1/consents/{consentId}/status**

**Response code**

<b>200 Ok</b>	The request has succeeded
---------------	---------------------------

**Response header**

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

**Response example**

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

6.3.2 Get consent status retail (PSU-ID flow retail customer)

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

**Path parameter**

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

**Request header**

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	Mandatory	Token SN for retail customer
<b>Content-Type</b>	mandatory	Content type application/json

**Response GET /v1/consents/{consentId}/status**

**Response code**

<b>200 Ok</b>	The request has succeeded
---------------	---------------------------

**Response header**

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

## Response example

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

### 6.3.3 Get consent status corporate (PSU-ID flow for corporate customer)

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

##### Path parameter

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

##### Request header

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	mandatory	<ul style="list-style-type: none"> <li>a. For standard token users: <b>Token SN - corporate customer</b></li> <li>b. For PKI USB stick or PKI smart card users: <b>Personal identification number (OIB) from the person - owner of the USB stick</b></li> </ul>
<b>PSU-CORPORATE-ID</b>	mandatory	Corporate customer OIB (Croatian VAT number)
<b>Content-Type</b>	mandatory	Content type application/json

#### Response GET /v1/consents/{consentId}/status

##### Response code

<b>200 Ok</b>	The request has succeeded
---------------	---------------------------

##### Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

## Response example

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

### 6.4 Delete consent

Delete consent.



6.4.1 Delete consent (pre-auth flow)

**Request DELETE /v1/consents/{consentId}**

**Path parameter**

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

**Request header**

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

**Response DELETE /v1/consents/{consentId}**

**Response code**

<b>204 No content</b>	The request has succeeded
-----------------------	---------------------------

**Response header**

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

6.4.2 Delete consent retail (PSU-ID flow)

**Request DELETE /v1/consents/{consentId}**

**Path parameter**

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

**Request header**

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	Mandatory	Token SN for retail customer
<b>Content-Type</b>	mandatory	Content type application/json

**Response DELETE /v1/consents/{consentId}**

**Response code**

<b>204 No content</b>	The request has succeeded
-----------------------	---------------------------

**Response header**

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

## 6.4.3 Delete consent corporate (PSU-ID flow)

### Request DELETE /v1/consents/{consentId}

#### Path parameter

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

#### Request header

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	Mandatory	<ul style="list-style-type: none"> <li>a. For standard token users: <b>Token SN - corporate customer</b></li> <li>b. For PKI USB stick or PKI smart card users: <b>Personal identification number (OIB) from the person - owner of the USB stick</b></li> </ul>
<b>PSU-CORPORATE-ID</b>	mandatory	Corporate customer OIB (Croatian VAT number)
<b>Content-Type</b>	mandatory	Content type application/json

### Response DELETE /v1/consents/{consentId}

#### Response code

<b>204 No content</b>	The request has succeeded
-----------------------	---------------------------

#### Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

## 6.5 Read account list

Reads a list of bank accounts, with balances where required. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system.

## 6.5.1 Read account list (pre-step flow)

### Request GET /v1/accounts

#### Query parameter

<b>withBalance</b>	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

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>Authorization</b>	mandatory	Oauth2 authorization bearer token
<b>Content-Type</b>	mandatory	Content type application/json
<b>Consent-ID</b>	mandatory	Identification of the consent for this access as granted by the PSU.

### Response GET /v1/accounts

#### Response code

<b>200 OK</b>	The request has succeeded
---------------	---------------------------

#### Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

**Response**

**example**

```

{
  "accounts": [
    {
      "resourceId": "2000011300",
      "iban": "HR7524070002000011300",
      "bban": "24070002000011300",
      "msisdn": "+3562596000",
      "currency": "HRK",
      "name": "My Corporate transactional account multicurrency",
      "product": "Corporate transactional account multicurrency",
      "cashAccountType": "CACC",
      "status": "enabled",
      "bic": "OTPVHR24",
      "usage": "ORGA",
      "details": "Corporate transactional account multicurrency",
      "_links": {
        "account": {
          "href": "/v1/accounts/2000011300"
        }
      }
    },
    {
      "resourceId": "0000002120",
      "iban": "HR93240700000000002120",
      "bban": "240700000000002120",
      "msisdn": "+3562514620",
      "currency": "EUR",
      "name": "ibanPaymentFailedWithScaSuccessfulMulti",
      "product": "Retail transactional account in EUR - STATELESS",
      "cashAccountType": "CACC",
      "status": "enabled",
      "bic": "OTPVHR24",
      "usage": "PRIV",
      "details": "Retail transactional account in EUR - STATELESS",
      "_links": {
        "account": {
          "href": "/v1/accounts/0000002120"
        }
      }
    }
  ]
}

```

6.5.2 Read account list retail (PSU-ID flow retail customer)

**Request GET /v1/accounts**

**Query parameter**

<b>withBalance</b>	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

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	Mandatory	Token SN retail customer
<b>Content-Type</b>	mandatory	Content type application/json
<b>Consent-ID</b>	mandatory	Identification of the consent for this access as granted by the PSU.

## Response GET /v1/accounts

### Response code

<b>200 OK</b>	The request has succeeded
---------------	---------------------------

### Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

## 6.5.3 Read account list corporate (PSU-ID flow corporate customer)

### Request GET /v1/accounts

#### Query parameter

<b>withBalance</b>	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

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	Mandatory	<ul style="list-style-type: none"> <li>a. For standard token users: <b>Token SN - corporate customer</b></li> <li>b. For PKI USB stick or PKI smart card users: <b>Personal identification number (OIB) from the person - owner of the USB stick</b></li> </ul>
<b>PSU-CORPORATE-ID</b>	mandatory	Corporate customer OIB (Croatian VAT number)
<b>Content-Type</b>	mandatory	Content type application/json
<b>Consent-ID</b>	mandatory	Identification of the consent for this access as granted by the PSU.

### Response GET /v1/accounts

#### Response code

<b>200 OK</b>	The request has succeeded
---------------	---------------------------

## Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

## 6.6 Read account details

Reads details about an account, with balances where required. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system. The addressed details of this account depends then on the stored consent addressed by *consentId*.

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

#### Path parameter

<b>accountId</b>	The account identification assigned to the created resource
------------------	---

#### Query parameter

<b>withBalance</b>	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

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>Consent-ID</b>	Mandatory	Consent ID for request
<b>Content-Type</b>	mandatory	Content type application/json

### Response GET /v1/accounts/{account-id}

#### Response code

<b>200 Ok</b>	The request has succeeded
---------------	---------------------------

#### Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

#### Response example

```

{
  "account": {
    "resourceId": "2000011300",
    "iban": "HR7524070002000011300",
    "bban": "24070002000011300",
    "msisdn": "+3562596000",
    "currency": "HRK",
    "name": "My Corporate transactional account multicurrency",
    "product": "Corporate transactional account multicurrency",
    "cashAccountType": "CACC",
    "status": "enabled",
    "bic": "OTPVHR24",
    "usage": "ORGA",
    "details": "Corporate transactional account multicurrency",
    "_links": {
      "account": {
        "href": "/v1/accounts/2000011300"
      }
    }
  }
}

```

### 6.7 Get balances

Reads account data from a given account addressed by "account-id".

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

##### Path parameter

<b>accountId</b>	The account identification assigned to the created resource
------------------	---

##### Request header

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>Consent-ID</b>	Mandatory	Consent ID for request
<b>Content-Type</b>	mandatory	Content type application/json

#### Response GET /v1/accounts/{account-id}/balances

##### Response code

<b>200 Ok</b>	The request has succeeded
---------------	---------------------------

##### Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

**Response example**

```

{
  "account": {
    "resourceId": "2000011300",
    "iban": "HR7524070002000011300",
    "bban": "24070002000011300",
    "msisdn": "+3562596000",
    "currency": "HRK",
    "name": "My Corporate transactional account multicurrency",
    "product": "Corporate transactional account multicurrency",
    "cashAccountType": "CACC",
    "status": "enabled",
    "bic": "OTPVHR24",
    "usage": "ORGA",
    "details": "Corporate transactional account multicurrency",
    "_links": {
      "account": {
        "href": "/v1/accounts/2000011300"
      }
    }
  }
}

```

6.8 Get transactions list

Reads account data from a given account addressed by "account-id".

**Request GET /v1/accounts/{account-id} /transactions/ {query-parameters}**

**Path parameter**

<b>accountId</b>	The account identification assigned to the created resource
------------------	---

**Query parameter**

<b>dateFrom</b>	Starting date (inclusive the date dateFrom) of the transaction list, mandated if no delta access is required.
<b>dateTo</b>	End date (inclusive the data dateTo) of the transaction list, default is "now" if not given.
<b>bookingStatus</b>	Permitted codes are „booked“ , „pending“ and „both“
<b>withBalance</b>	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.

**Request header**

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>Consent-ID</b>	Mandatory	Id from consent
<b>Content-Type</b>	mandatory	Content type application/json

**Response GET /v1/accounts/{account-id} /transactions/ {query-parameters}**



## Response code

200 Ok	The request has succeeded
--------	---------------------------

## Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

## Response example

### 6.9 Get transactions details

Reads transaction details from a given transaction addressed by "transactionId" on a given account addressed by "account-id".

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

##### Path parameter

<b>accountId</b>	The account identification assigned to the created resource
<b>transactionId</b>	This identification is given by the attribute resourceId of the corresponding entry of a transaction list.

##### Request header

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>Consent-ID</b>	Mandatory	Id from consent
<b>Content-Type</b>	mandatory	Content type application/json

#### Response GET /v1/accounts/{account-id} /transactions/ {query-parameters}

##### Response code

200 Ok	The request has succeeded
--------	---------------------------

##### Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

**Response example**

6.10 Start the authorization process for a consent (pre-step flow)

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

**Path parameter**

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

**Request header**

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>Authorization</b>	Mandatory	Oauth2 authorization bearer token
<b>TPP-Redirect-Preferred</b>	optional	If it equals "true", the TPP prefers a redirect approach.
<b>TPP-Redirect-URI</b>	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
<b>Content-Type</b>	mandatory	Content type application/json

**Response POST /v1/consents/{consent-id}/authorisations**

**Response code**

<b>201 Created</b>	The request has been fulfilled and has resulted in one or more new resources being created
--------------------	--

**Response header**

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Aspsp-Scs-Approach</b>	Possible values are: REDIRECT or DECOUPLED
<b>Content-Type</b>	Content type application/json

6.11 Start the authorization process for a consent (PSU flow retail customer)

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

**Path parameter**

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

**Request header**

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

<b>PSU-ID</b>	Mandatory	Token SN
<b>TPP-Redirect-Preferred</b>	optional	If it equals "true", the TPP prefers a redirect approach.
<b>TPP-Redirect-URI</b>	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
<b>Content-Type</b>	mandatory	Content type application/json

**Response POST /v1/consents/{consent-id}/authorisations**

**Response code**

<b>201 Created</b>	The request has been fulfilled and has resulted in one or more new resources being created
--------------------	--

**Response header**

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Aspsp-Sca-Approach</b>	Possible values are: REDIRECT or DECOUPLED
<b>Content-Type</b>	Content type application/json

6.12 [Start the authorization process for a consent\(PSU flow corporate customer\)](#)

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

**Path parameter**

<b>consentId</b>	The consent identification assigned to the created resource
------------------	---

**Request header**

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>PSU-ID</b>	Mandatory	<ul style="list-style-type: none"> <li>a. For standard token users: <b>Token SN - corporate customer</b></li> <li>b. For PKI USB stick or PKI smart card users: <b>Personal identification number (OIB) from the person - owner of the USB stick</b></li> </ul>
<b>PSU-CORPORATE-ID</b>	Mandatory	Customer OIB
<b>TPP-Redirect-Preferred</b>	optional	If it equals "true", the TPP prefers a redirect approach.
<b>TPP-Redirect-URI</b>	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
<b>Content-Type</b>	mandatory	Content type application/json

**Response POST /v1/consents/{consent-id}/authorisations**

**Response code**

<b>201 Created</b>	The request has been fulfilled and has resulted in one or more new resources being created
--------------------	--

## Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Aspsp-Sca-Approach</b>	Possible values are: REDIRECT or DECOUPLED
<b>Content-Type</b>	Content type application/json

### 6.13 Consent authorisation using Strong Customer Authentication (SCA)

Account information can only be requested after a consent has been created. The PreAuth is not sufficient to authorize a consent..

#### 6.13.1 Consent authorisations: redirect SCA approach

During this approach TPP has to send Tpp-Redirect-Preferred 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 OTP bank login form(Figure 13).

After successful login consent summary and approval form will be displayed where PSU has to approval 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 start the authorisation process for a consent will create consent authorization object and return scaOauth steering link inside JSON response. Same as in implicit method following this redirect link will redirect PSU to the OTP bank consent summary and SCA selection, approval form. It's highly recommended to use implicit method with SCA redirect approach.

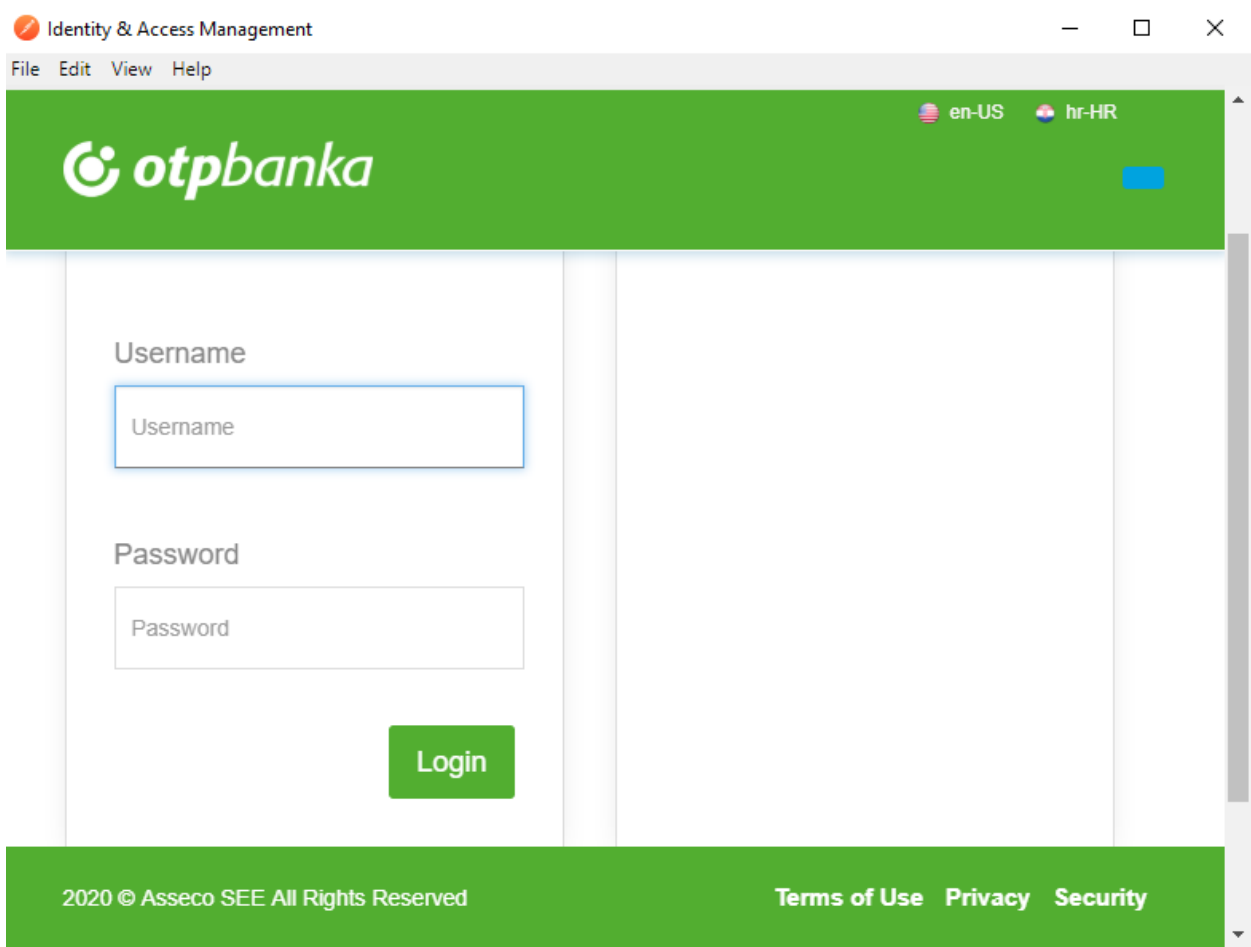


Figure 4 OTP bank login form

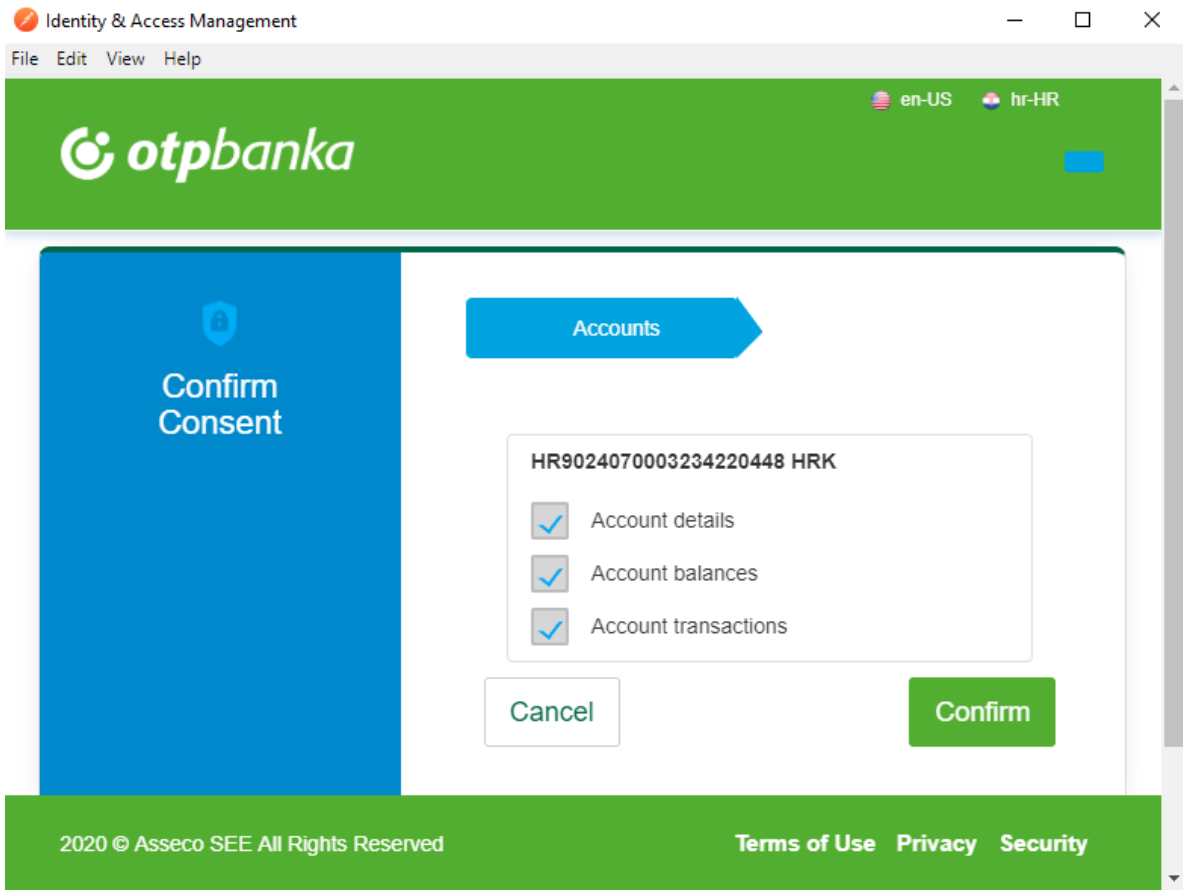
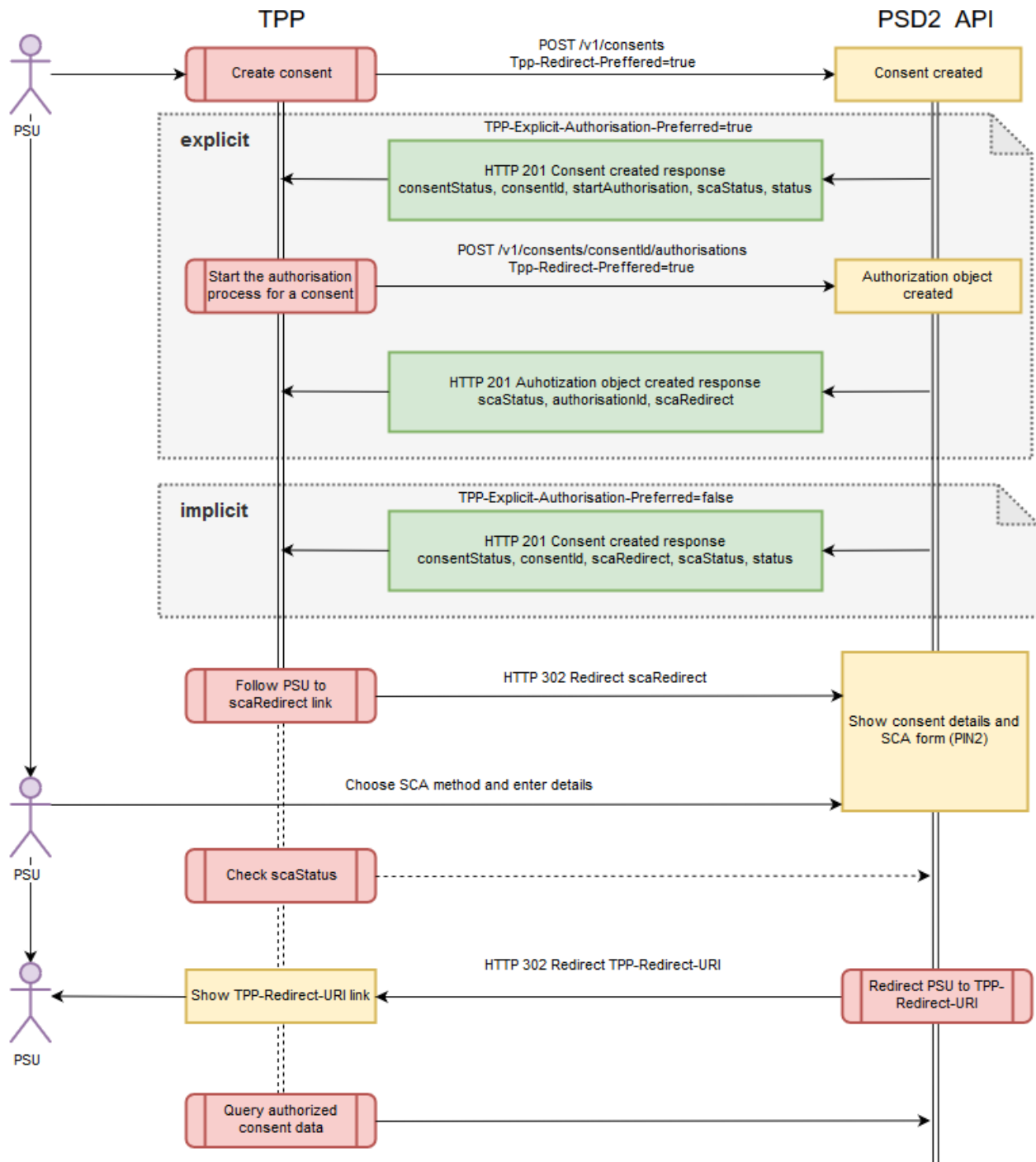


Figure 5 Confirm Consent form

Create consent redirect approach



## 7. Payments endpoints

### 7.1. Payments initiation

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

##### Path parameter

<b>payment-product</b>	The addressed payment product endpoint, e.g. for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers, domestic-payment, instant-domestic-credit-transfers, cross-border-credit-transfers
------------------------	---

##### Request header

<b>X-Request-ID</b>	mandatory	ID of the request, unique to the call, as determined by the initiating party
<b>Authorization</b>	mandatory	Oauth2 authorization bearer token
<b>Content-Type</b>	mandatory	Content type application/json
<b>PSU-IP-Address</b>	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.

##### Request body example for sepa-credit transfer

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

##### Request example for sepa-credit transfer

```
[
  {
    "debtorAccount": {
      "iban": "HR7524070002000011300"
    },
    "instructedAmount": {
      "currency": "EUR",
      "amount": "0.01"
    },
    "creditorAccount": {
      "iban": "HR9024070003234220448"
    },
    "creditorName": "Test PSD2 Interface"
  }
]
```

##### Response POST /v1/payments/{payment-product}



## Response code

<b>201 Created</b>	The request has been fulfilled and has resulted in one or more new resources being created
--------------------	--

## Response header

<b>Location</b>	Location of the created resource (if created)
<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Aspsp-Sca-Approach</b>	Possible values are: REDIRECT or DECOUPLED
<b>Content-Type</b>	Content type application/json

## Response example

```

{
  "transactionStatus": "RCVD",
  "paymentId": "d30b410bc6634fc2b8fe78f6cac1b137",
  "transactionFees": {
    "amount": "0.01",
    "currency": "EUR"
  },
  "transactionFeeIndicator": false,
  "scaMethods": [],
  "challengeData": {
    "data": "Default challenge"
  },
  "_links": {
    "scaRedirect": {
      "href": ""
    },
    "self": {
      "href": "v1/payments/sepa-credit-transfers/d30b410bc6634fc2b8fe78f6cac1b137"
    },
    "status": {
      "href": "v1/payments/sepa-credit-transfers/d30b410bc6634fc2b8fe78f6cac1b137/status"
    },
    "scaStatus": {
      "href": "v1/payments/sepa-credit-transfers/d30b410bc6634fc2b8fe78f6cac1b137/authorisations/185a13adc0a8496d99839076839251da"
    }
  }
}

```

## 7.2. Get payment transaction status

### Request POST /v1/payments/{payment-product}/{payment-id}/status

#### Path parameter

<b>payment-product</b>	The addressed payment product endpoint, e.g. for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers,domestic-payment, instant-domestic-credit-transfers, cross-border-credit-transfers
<b>payment-id</b>	The consent identification assigned to the created resource

#### Request header

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

<b>PSU-IP-Address</b>	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.
-----------------------	-----------	---

## Response POST /v1/payments/{payment-product}/{payment-id}/status

### Response code

<b>200 Ok</b>	The request has succeeded
---------------	---------------------------

### Response header

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

### Response example

```
{
  "transactionStatus": "RCVD"
}
```

## 7.3. Get payment request

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

#### Path parameter

<b>payment-product</b>	The addressed payment product endpoint, e.g. for SEPA Credit Transfers (SCT). The supported product is: sepa-credit-transfers,domestic-payment, instant-domestic-credit-transfers, cross-border-credit-transfers
<b>payment-Id</b>	The consent identification assigned to the created resource

#### Request header

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

### Response POST /v1/payments/{payment-product}/{payment-id}

## 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
Content-Type	Content type application/json

## Response example

```
{
  "debtorAccount": {
    "iban": "HR7524070002000011300"
  },
  "instructedAmount": {
    "amount": "0.01",
    "currency": "EUR"
  },
  "creditorAccount": {
    "iban": "HR9024070003234220448"
  },
  "creditorName": "Test PSD2 Interface",
  "transactionStatus": "RCVD"
}
```

## 7.4. Delete payment request

It initiates the cancellation of a payment. Depending on the payment-service, the payment-product and the ASPSP's implementation, this TPP call might be sufficient to cancel a payment. If an authorisation of the payment cancellation is mandated by the ASPSP, a corresponding hyperlink will be contained in the response message.

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

#### Path parameter

payment-product	The addressed payment product endpoint
payment-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	mandatory	Content type application/json

**Response DELETE /v1/payments/{payment-product}/{payment-id}**

**Response code**

<b>204 No content</b>	The request has succeeded
-----------------------	---------------------------

**Response header**

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Content-Type</b>	Content type application/json

7.5. Update PSU data for payment initiation

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

**Path parameter**

<b>payment-product</b>	The addressed payment product endpoint.
<b>payment-id</b>	The consent identification assigned to the created resource
<b>authorisation-id</b>	Authorisation object ID

**Request header**

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

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

**Response code**

<b>200 Ok</b>	The request has been fulfilled and has resulted in one or more new resources being created
---------------	--

**Response header**

<b>X-Request-ID</b>	ID of the request, unique to the call, as determined by the initiating party
<b>Aspsp-Sca-Approach</b>	Possible values are: REDIRECT or DECOUPLED
<b>Content-Type</b>	Content type application/json

**Response example**

```
{
  "ScaStatus": "finalised",
  "ChosenScaMethod": null,
  "ChallengeData": null,
  "Links": null,
  "Pain002Response": null
}
```

## 7.6. Payment authorisation using Strong Customer Authentication (SCA)

To confirm payments, a SCA takes place for each transaction. The PreAuth is not sufficient to authorize a payment. After the authorization of a payment, the API responds if the payment was accepted or declined, similar to the OTP Online Banking. It is not possible to send an automated confirmation that the payment was booked successfully because of the OTP batch-booking approach.

### 7.6.1. Payment authorisation: redirect SCA approach

During this approach TPP has to send *Tpp-Redirect-Preferred* 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 redirected to the bank payment summary and SCA selection and approval form where PSU has to 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 start 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 OTP bank payment summary and SCA selection, approval form. It's highly recommended to use implicit method with SCA redirect approach.

## 7.7 Payments without debtor account

Additional possibility for TPP and PSU clients is creating a payment resource without providing *debtorAccount* object in JSON. In that case, PSU will have option to select any IBAN\* it has access to, after completing SCA. This option helps TPP clients to have a more flexible payment initiation and helps PSU to select any IBAN\* available as debtor account, during the authorization of the payment. This feature is currently available on OAuth2 Redirect approach.

\*IBAN must be in the same currency as the currency of the payment order.

## 8. Miscellaneous