OTP PSD2 API production documentation

Document version	1.1
Release date	14.01.2020.

Document History

Version	Date	Description
1.1	14.01.2020	Initial version

Contents

1. Definitions

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

TPP (Third Party Provider)	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 **consent**.
PSU (Payment Service User) or user	The user refers to the bank customer who uses the TPP application.
ASPSP (Account Servicing Payments Service Provider)	This is the account servicing provider, i.e, the OTP bank.
PISP (Payment Initiation Service Provider)	This is a service provider who can initiate a payment transaction on behalf of the customer.
Sandbox	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 https://apiportal.sandbox.otpbanka.hr/portal/
SCA (Strong Customer Authentication)	The process of using a strong (2-factor) identification method to identify the customer.
Authentication	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.
Consent	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 prod 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:

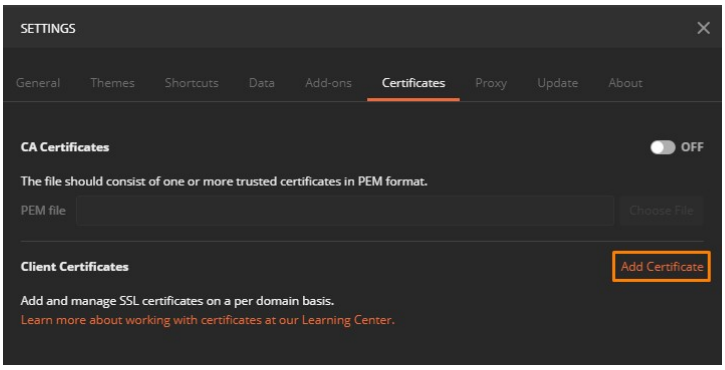
New users must fill in the following fields:

- **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.



CA Certificates

The file should consist of one or more trusted certificates in PEM format.

Client Certificates

Add and manage SSL certificates on a per domain basis.
Learn more about working with certificates at our Learning Center.

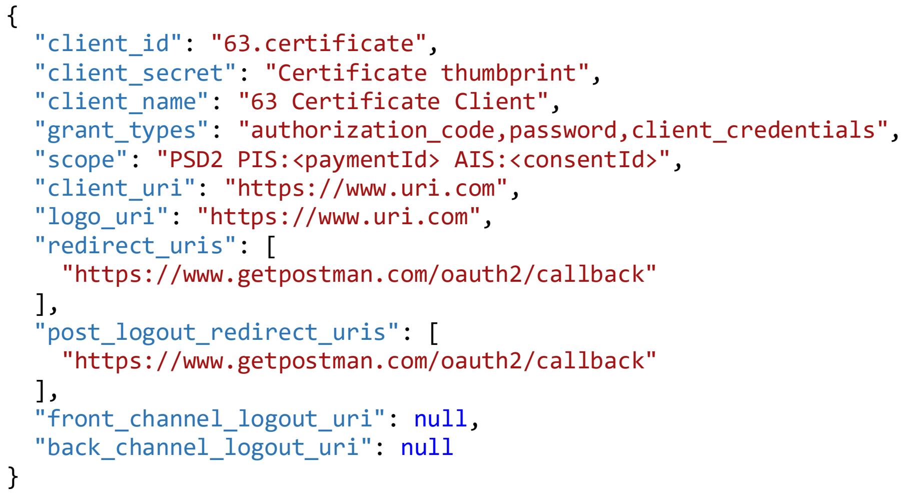
Figure 1 Adding certificate for Mutual TLS

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**

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 c*lientId* (Figure 2). These param should be passed to the Oauth server's /authorization and /token endpoints.

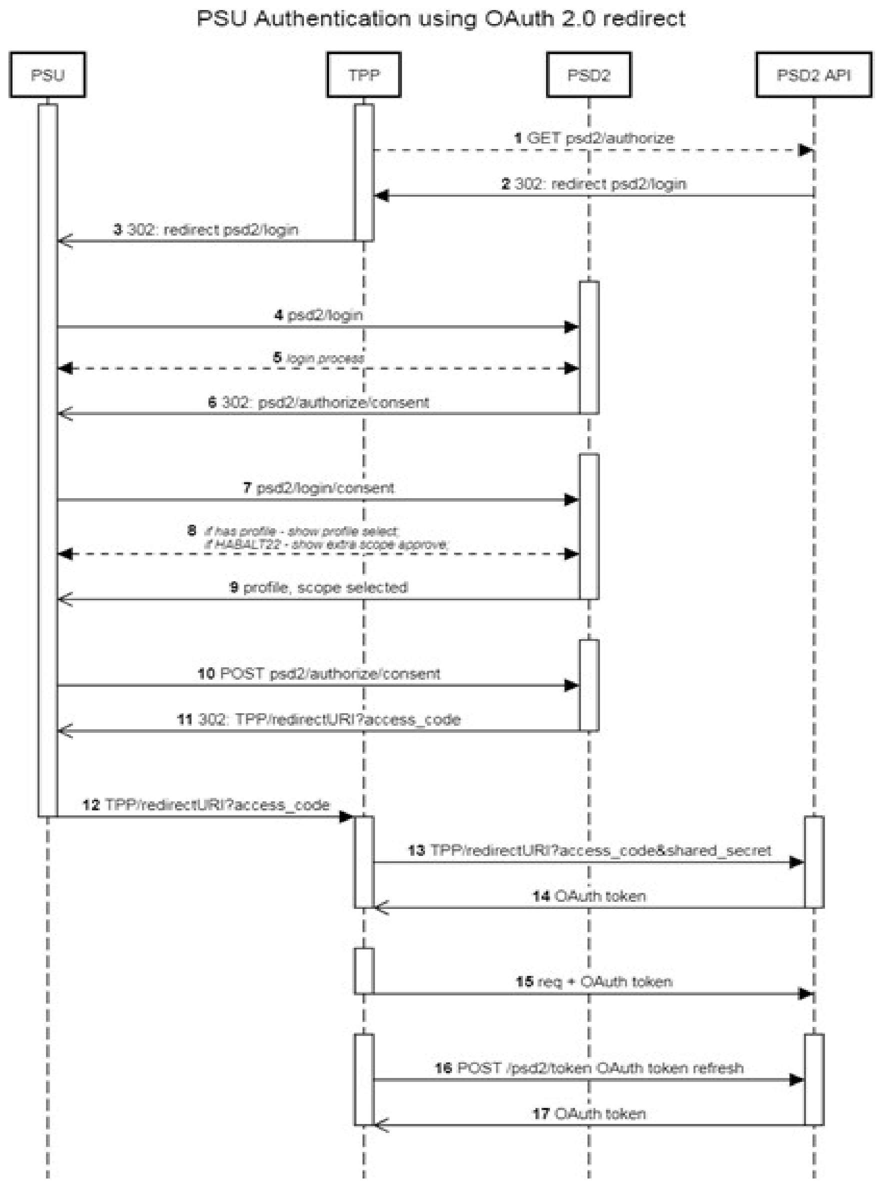

PSU Authentication using OAuth 2.0 redirect
PSU
TPP
PSD2
PSD2 API
1 GET psd2/authorize
2 302: redirect psd2/login
3 302: redirect psd2/login
4 psd2/login
5 login process
6 302: psd2/authorize/consent
7 psd2/login/consent
8 if has profile - show profile select;
if HABALT22 - show extra scope approve;
9 profile, scope selected
10 POST psd2/authorize/consent
11 302: TPP/redirectURI?access_code
12 TPP/redirectURI?access_code
13 TPP/redirectURI?access_code&shared_secret
14 OAuth token
15 req + OAuth token
16 POST /psd2/token OAuth token refresh
17 OAuth token

Figure 3 PSU Authentication using OAuth 2.0 redirect

4.2. Terms used in OAuth 2.0

Definition	Description
OAuth 2.0	OAuth 2.0 is an authorisation framework that enables applications to obtain limited access to user accounts on an HTTP service.
OAuth 2.0 flow	Since the API is built for backend communication and not designed to handle direct client communication it is mandated that the grant_type will be authorization_code and will require the use of a client_secret and a client_id.
Client_secret	A client secret is obtained by registering in the OTP Banking PSD2 API(Figure 2)
Client_id	A client Id is obtained by registering in the OTP Banking PSD2 API(Figure 2)
Scope	A scope according to OAuth 2.0 defines what data can be accessed. API supported scopes are PIS,AIS.
Redirect_uri	The redirect_uri is an address used by OAuth providers as a location to deliver the access_token by means of a browser redirect.
State	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

response_type	mandatory	„code“ is only supported as response type
grant_type	mandatory	„code“ is only supported as grant type
client_id	mandatory	Generated application clientId
scope	mandatory	Scope should be „OTP.PSD2“
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 --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.

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

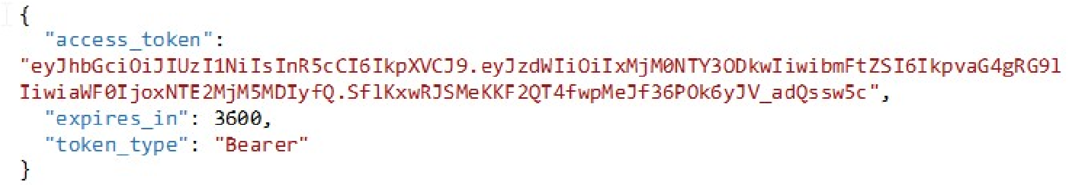
code	mandatory	code that was received in callback
client_id	mandatory	Generated application clientId from developer's portal
scope	mandatory	This field should be equal to the scope parameter received in callback
grant_type	mandatory	This field needs to be equal to „authorization_code"
redirect_uri	mandatory	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.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9l
IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c",
  "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-redirected to the TPP’s system.

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

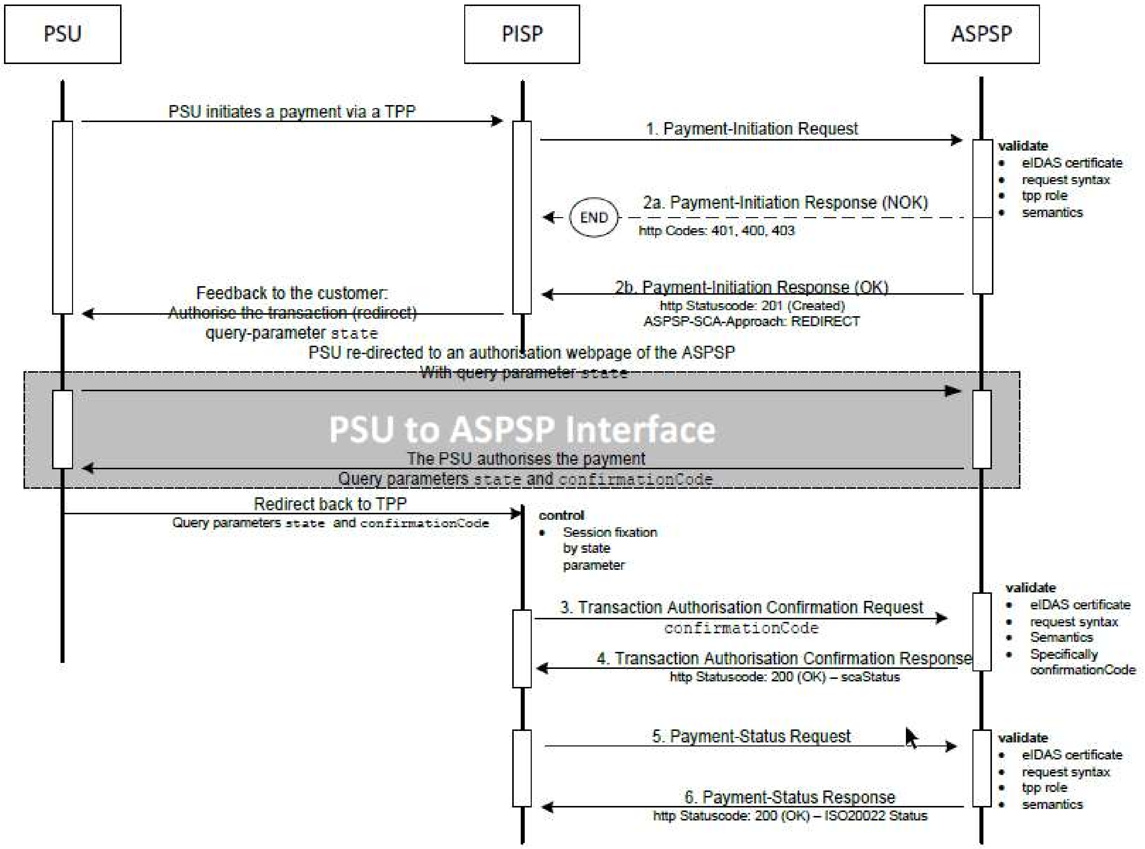

PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
Feedback to the customer:
Authorise the transaction (redirect)
query-parameter state
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
PSU re-directed to an authorisation webpage of the ASPSP
With query parameter state
PSU to ASPSP Interface
The PSU authorises the payment
Query parameters state and confirmationCode
Redirect back to TPP
Query parameters state and confirmationCode
control
• Session fixation by state parameter
validate
• eIDAS certificate
• request syntax
• Semantics
• Specifically confirmationCode
3. Transaction Authorisation Confirmation Request
confirmationCode
4. Transaction Authorisation Confirmation Response
http Statuscode: 200 (OK) – scaStatus
5. Payment-Status Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
6. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

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-authorisation-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.

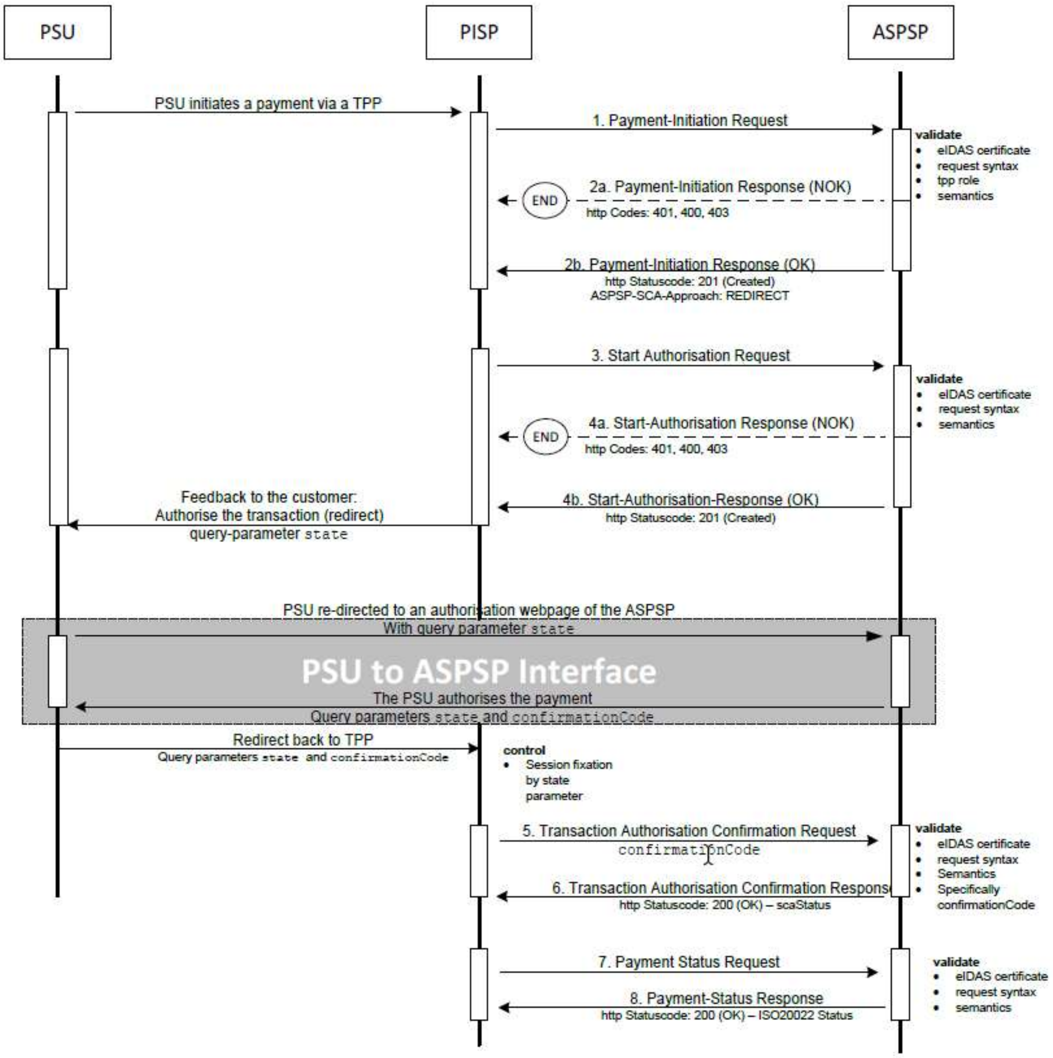

PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
3. Start Authorisation Request
validate
• eIDAS certificate
• request syntax
• semantics
END
4a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
Feedback to the customer:
Authorise the transaction (redirect)
query-parameter state
4b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
PSU re-directed to an authorisation webpage of the ASPSP
With query parameter state
PSU to ASPSP Interface
The PSU authorises the payment
Query parameters state and confirmationCode
Redirect back to TPP
Query parameters state and confirmationCode
control
• Session fixation by state parameter
5. Transaction Authorisation Confirmation Request
confirmationCode
6. Transaction Authorisation Confirmation Respons
http Statuscode: 200 (OK) – scaStatus
validate
• eIDAS certificate
• request syntax
• Semantics
• Specifically confirmationCode
7. Payment Status Request
8. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status
validate
• eIDAS certificate
• request syntax
• semantics

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

6. Accounts endpoints

5.1. 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.

A consent can become invalid, if:

- the PSU, TPP or ASPSP (OTP) revokes the consent.

- the consent was created for a specific period of time (validUntil).

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"
Content-Type	mandatory	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 without PSU involvement per day.
combinedServiceIndicator	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/

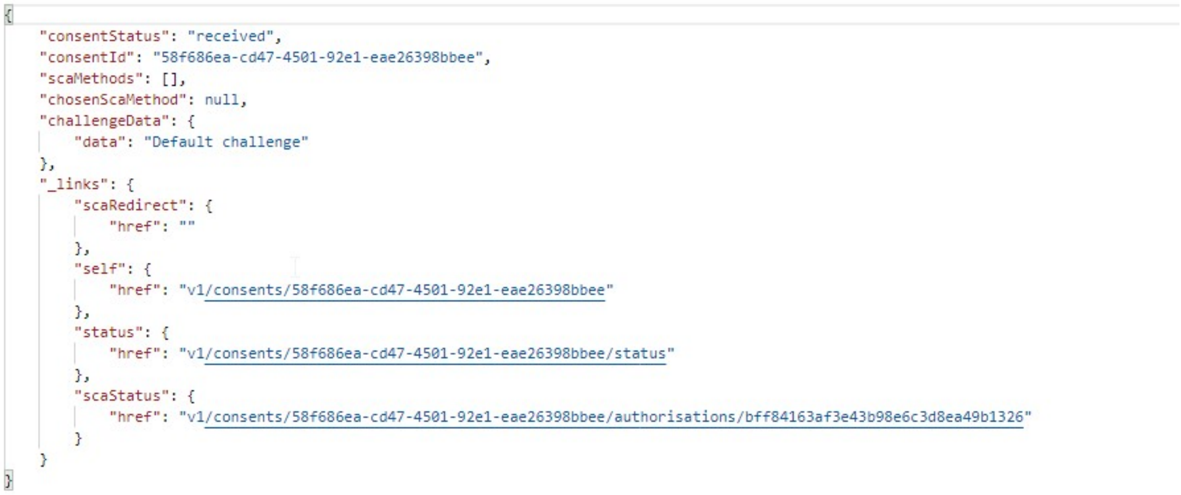
Response code

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
Content-Type	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"
        }
    }
}
```

5.2. Get consent request

Returns the content of an account information consent object.

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

Response GET /v1/consents/{consentId}

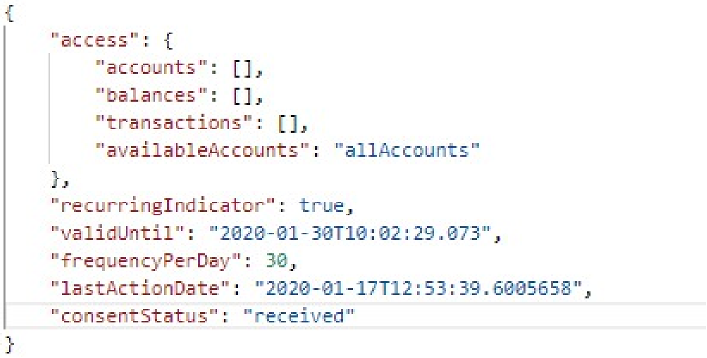
Response code

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": "2020-01-30T10:02:29.073",
    "frequencyPerDay": 30,
    "lastActionDate": "2020-01-17T12:53:39.6005658",
    "consentStatus": "received"
}
```

5.3. Get consent status request

Returns the content of an account information consent object.

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

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

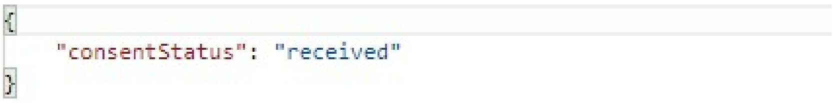
Response code

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



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

5.4. Delete consent

Delete content.

Request DELETE /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
Authorization	Mandatory	Oauth2 authorization bearer token
Content-Type	mandatory	Content type application/json

Response DELETE /v1/consents/{consentId}

Response code

204 No content	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

5.5. 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.

Request GET /v1/accounts

Query parameter

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
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	mandatory	Content type application/json
Consent-ID	mandatory	Identification of the consent for this access as granted by the PSU.

Response GET /v1/accounts

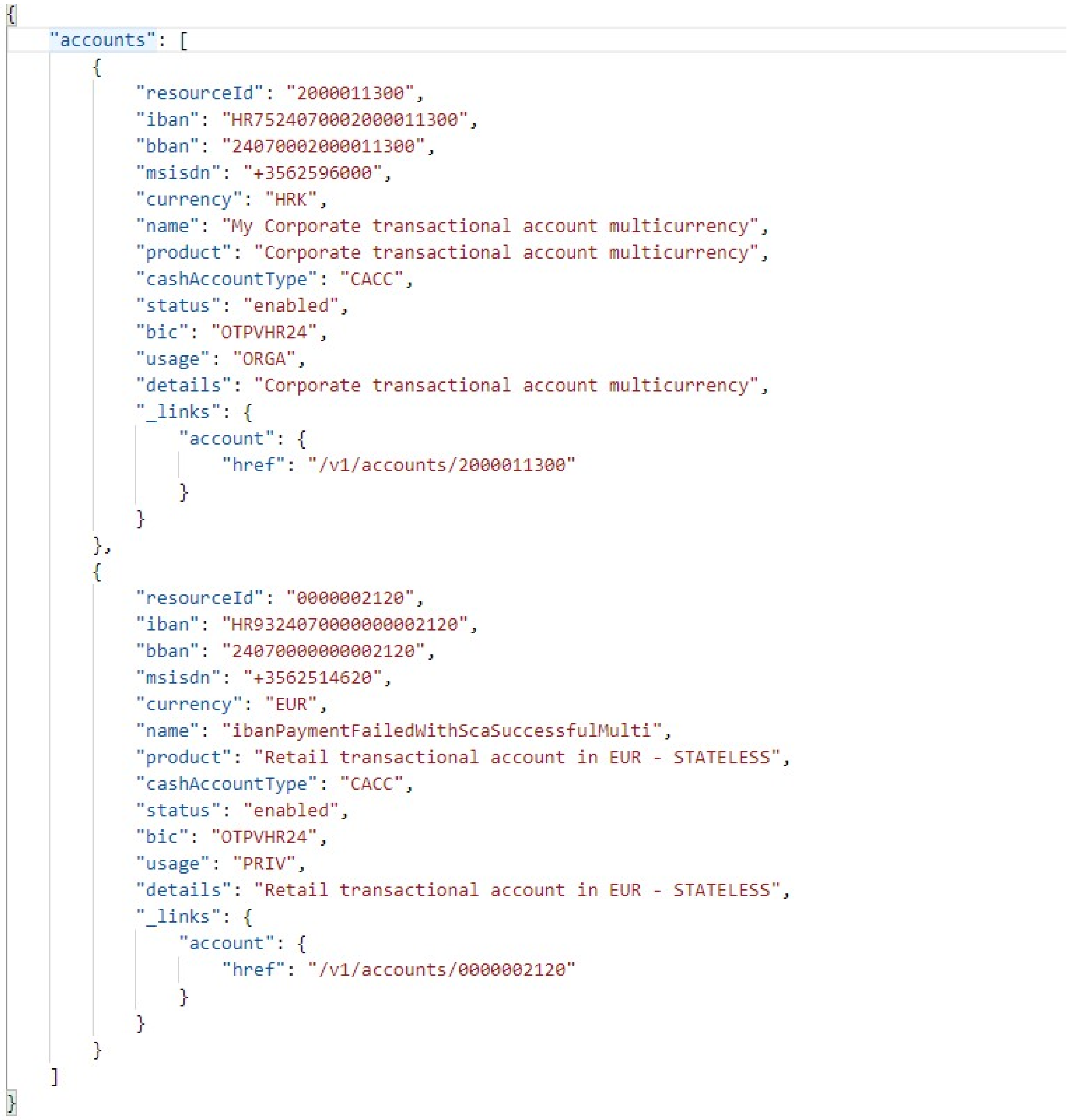
Response code

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



```
{
    "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": "HR9324070000000002120",
            "bban": "24070000000002120",
            "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"
                }
            }
        }
    ]
}
```

5.6. 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*, respectively the OAuth2 access token.

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

Path parameter

accountId	The account identification assigned to the created resource

Query parameter

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

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

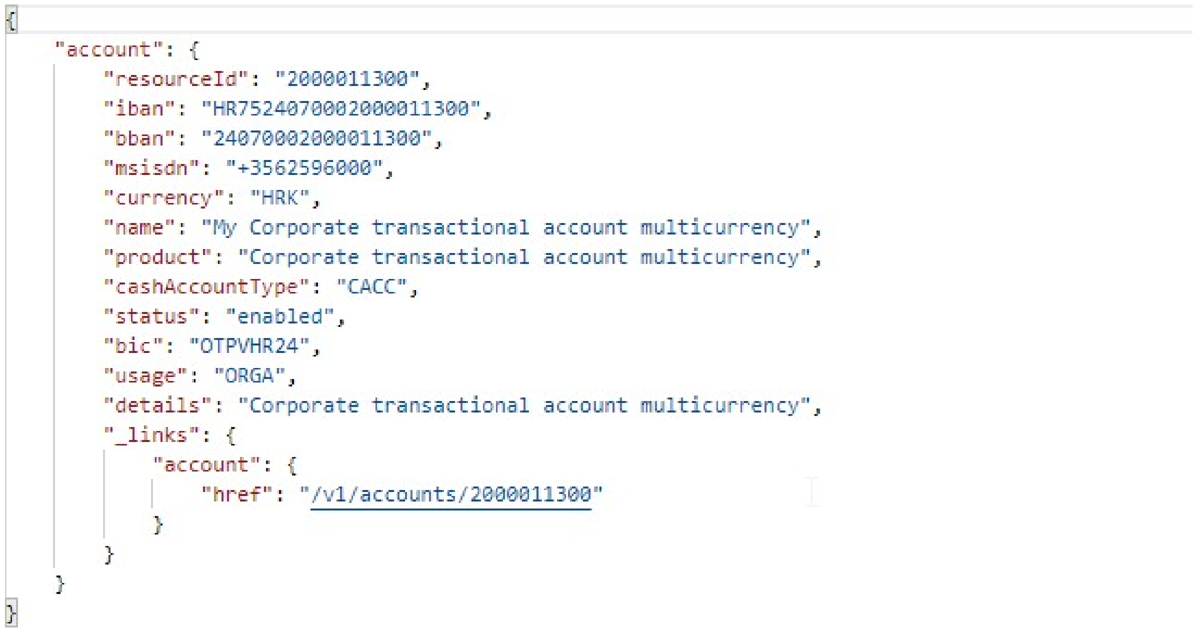
Response code

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



```
{
    "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"
            }
        }
    }
}
```

5.7. Get balances

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

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

Path parameter

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

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

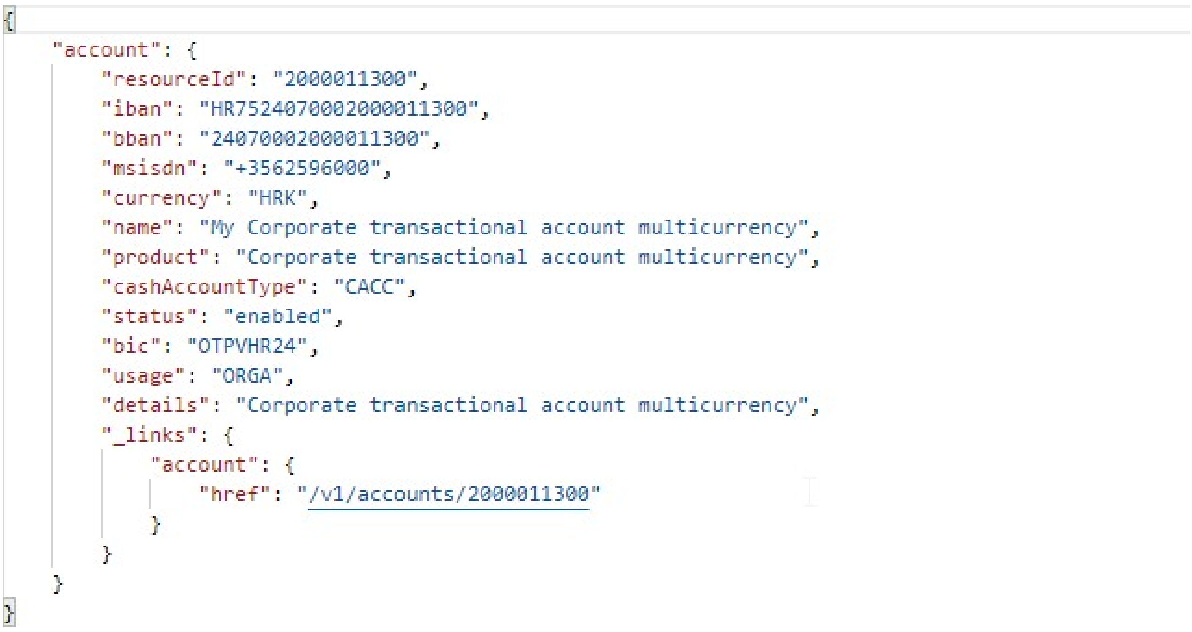
Response code

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



```
{
    "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"
            }
        }
    }
}
```

5.8. 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

accountId	The account identification assigned to the created resource

Query parameter

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.

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

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

5.9. 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

accountId	The account identification assigned to the created resource
transactionId	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
Authorization	mandatory	Oauth2 authorization bearer token
Content-Type	mandatory	Content type application/json

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

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

5.10. Start the authorization process for a consent

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

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
Authorization	Mandatory	Oauth2 authorization bearer token
TPP-Redirect-Preferred	optional	If it equals "true", the TPP prefers a redirect approach.
TPP-Redirect-URI	conditional	Mandated for the Redirect SCA Approach (including OAuth2 SCA approach), specifically when TPP-Redirect-Preferred equals "true"
Content-Type	mandatory	Content type application/json

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

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
Content-Type	Content type application/json

Response example

5.11. 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..

5.11.1. Consent authorisations: redirect SCA approach

During this approach TPP has to 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 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.

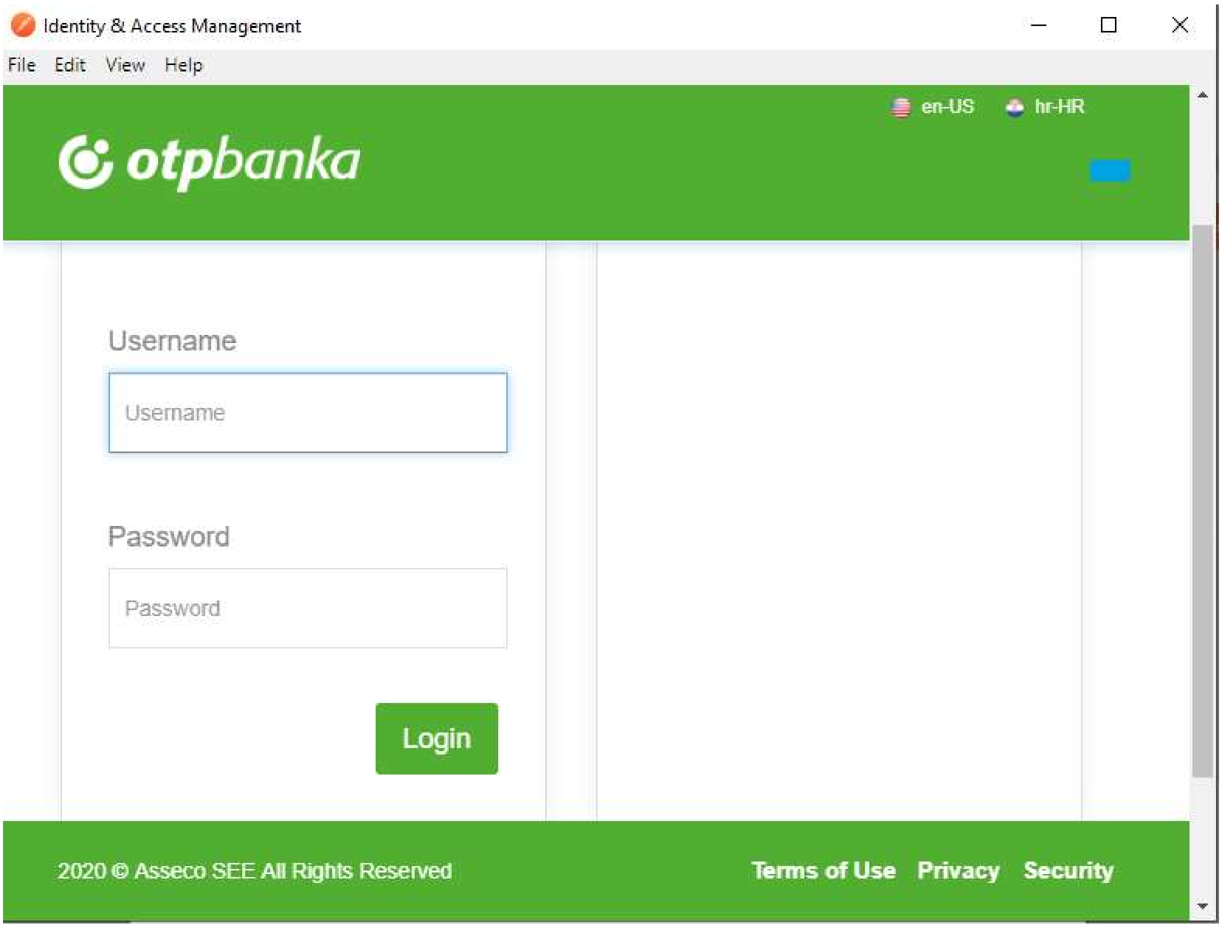
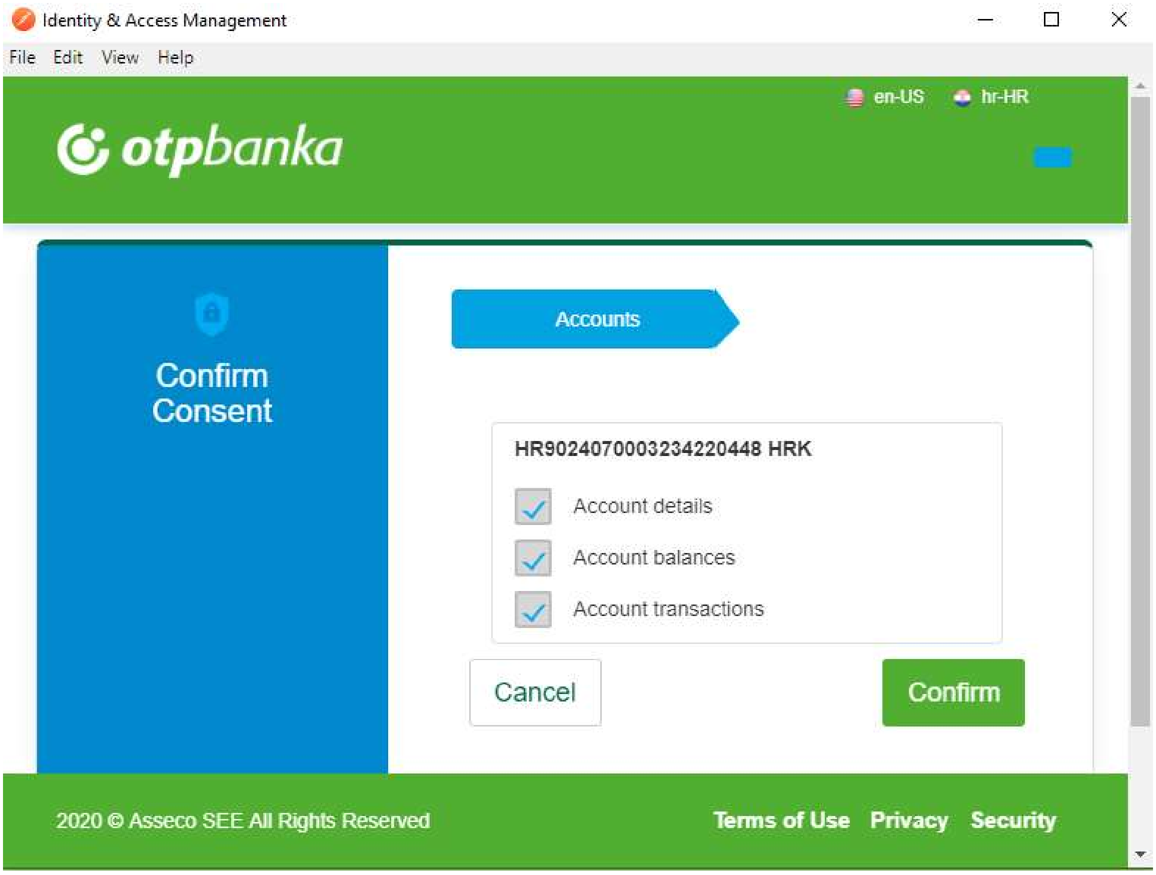


Figure 4 OTP bank login form



Identity & Access Management
File Edit View Help
en-US hr-HR
otpbanka
Confirm Consent
Accounts

HR9024070003234220448 HRK

☑ Account details

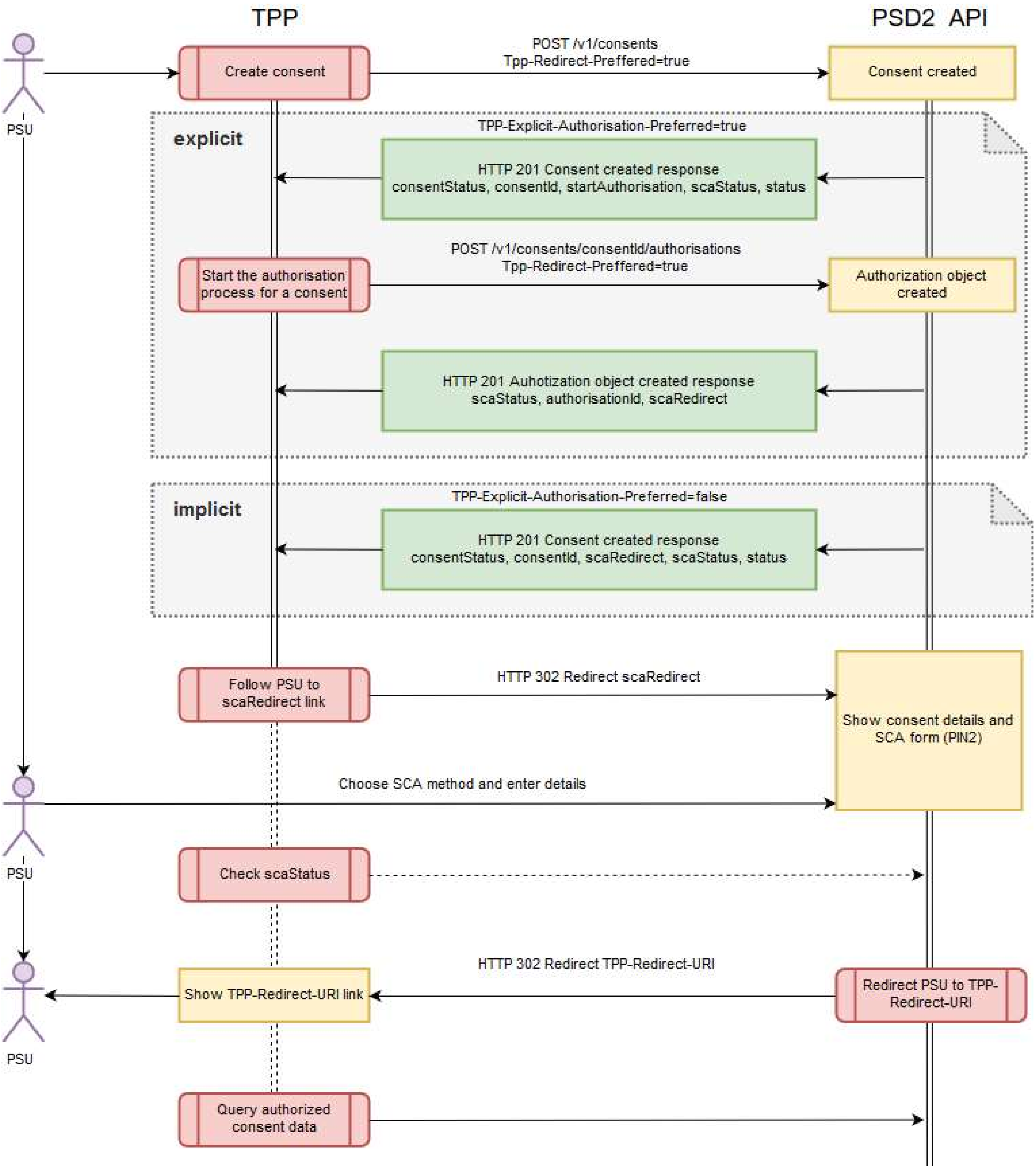
☑ Account balances

☑ Account transactions

Cancel
Confirm
2020 © Asseco SEE All Rights Reserved
Terms of Use Privacy Security

Figure 5 Confirm Consent form

Create consent redirect approach


TPP
PSD2 API
PSU
Create consent
POST /v1/consents
Tpp-Redirect-Preffered=true
Consent created
explicit
TPP-Explicit-Authorisation-Preferred=true
HTTP 201 Consent created response
consentStatus, consentId, startAuthorisation, scaStatus, status
POST /v1/consents/consentId/authorisations
Tpp-Redirect-Preffered=true
Start the authorisation process for a consent
Authorization object created
HTTP 201 Auhotization object created response
scaStatus, authorisationId, scaRedirect
implicit
TPP-Explicit-Authorisation-Preferred=false
HTTP 201 Consent created response
consentStatus, consentId, scaRedirect, scaStatus, status
Follow PSU to scaRedirect link
HTTP 302 Redirect scaRedirect
Show consent details and SCA form (PIN2)
PSU
Choose SCA method and enter details
Check scaStatus
HTTP 302 Redirect TPP-Redirect-URI
PSU
Show TPP-Redirect-URI link
Redirect PSU to TPP-Redirect-URI
Query authorized consent data

4. Payments endpoints

4.1. Payments initiation

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

Path parameter

payment-product	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

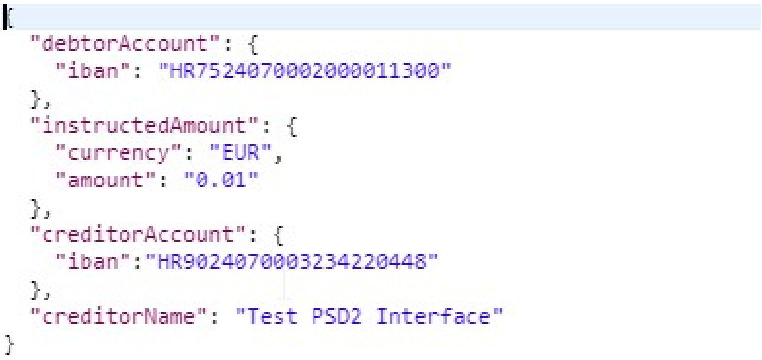
Request header

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
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.

Request body example for sepa-credit transfer

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 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}

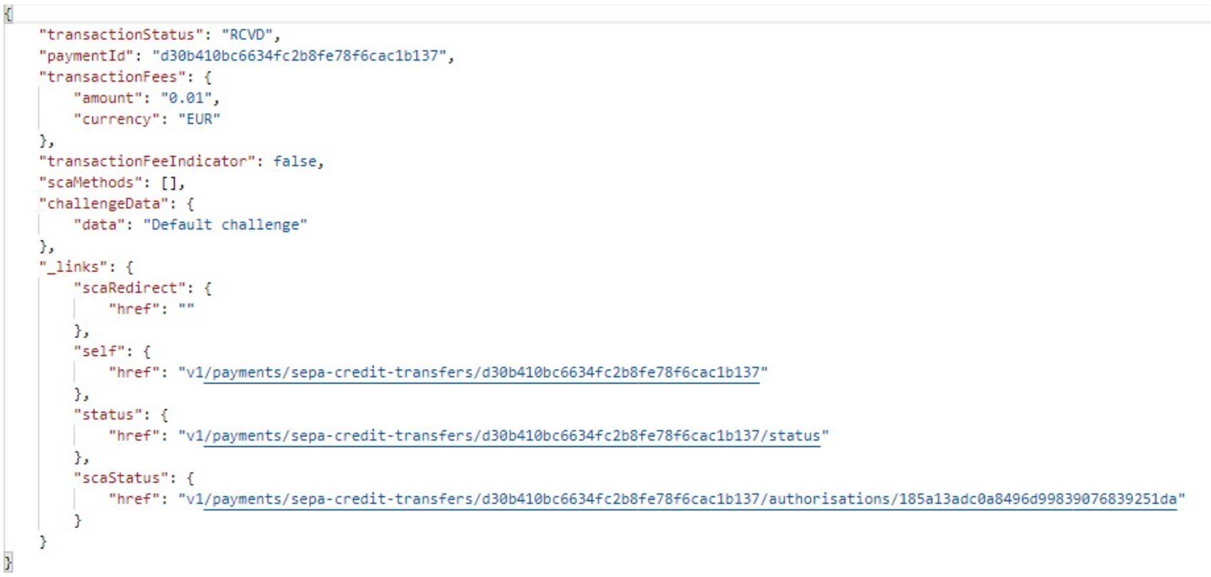
Response code

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
Content-Type	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"
        }
    }
}
```

4.2. Get payment transaction status

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

Path parameter

payment-product	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
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

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.

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

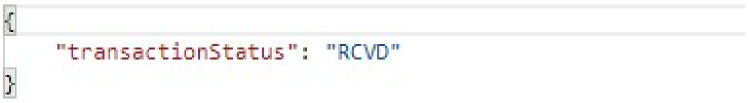
Response code

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



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

4.3. Get payment request

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

Path parameter

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

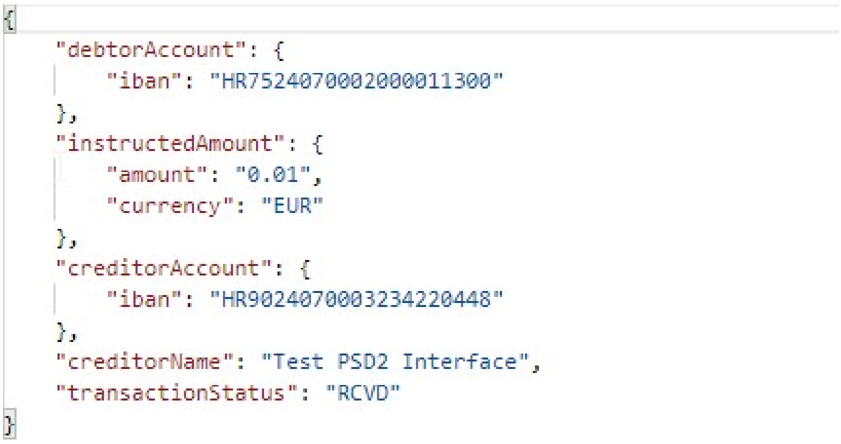
Response code

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"
}
```

4.4. 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

204 No content	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

4.5. Update PSU data for payment initiation

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

Path parameter

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

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

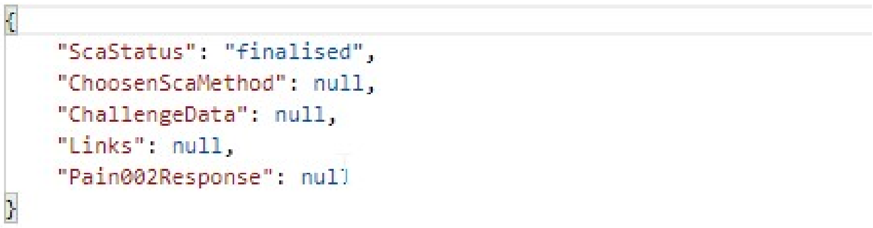
Response code

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
Content-Type	Content type application/json

Response example



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

4.6. 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 success-fully because of the OTP batch-booking approach.

4.6.1. Payment authorisation: redirect SCA approach

During this approach TPP has to 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 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.

5. Miscellaneous