TPP USER GUIDE

Document version	1.0
Release date	11.06.2019.

Glossary / List of Abbreviations and Terms Used in this Document

Abbreviation / Term	Expansion / Description
TPP	Third party provider. A legal entity acting either as AISP, PISP or both
AIS	Account information services
PIS	Payment instruction services
User	Individual that is registered at both TPP and Bank and has granted consent to TPP
IAM	Identity access manager
CA	Certification authority
OAuth 2	Industry-standard protocol for authorization
SCA	Strong customer authentication

Contents

1. Background

This document is used to describe the functionalities of the Asseco PSD2 enabler IAM application.

1.1. Purpose

Purpose of this document is to provide details on how to use the product from a functionality point of view.

1.2. Intended audience

Main audience of this document are TPPs that want to register their application at bank in order to use PSD2 API methods exposed by the bank.

1.3. Scope

Descriptions in this document describe the following processes and flows:

1. TPP application registration
2. Creating and authorizing AIS and PIS resources using OAuth2 SCA flow
3. Creating and authorizing AIS and PIS resources using Redirect flow

2. Registration

Goal

Register your application in order to gain access to PSD2 API exposed by the bank.

Preconditions

Obtained valid X509 Certificate from trusted CA that satisfies requirements stated in ETSI TS 119 495 V1.2.1 directive.

Installed Postman or similar application.

How to access

Through the API call from TPP application, Postman client or similar apps.

Overview

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:

- **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.

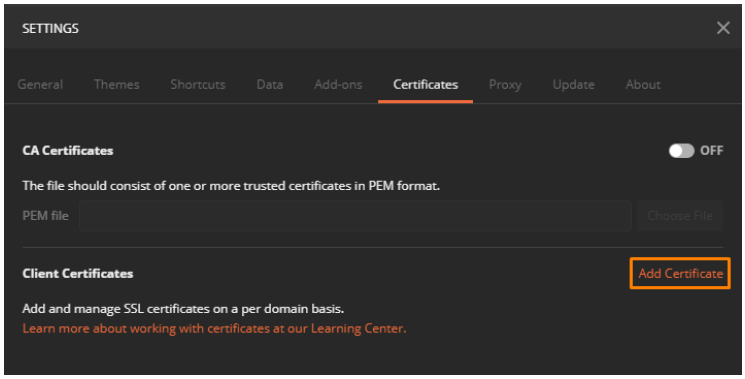

Figure 1: Adding certificate for Mutual TLS

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.

After this setup is complete TPP should send a registration request to the IAM application. 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
}
```


This response contains data that will be needed later for starting the OAuth2 flow for authorizing AIS and PIS resources. 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. Creating and authorizing AIS and PIS resources using the OAuth2 flow

Goal

Goal of this section is to successfully create AIS resource (consent) and PIS resource (payment), to start authorization for created resources and to authorize resources using the OAuth2 protocol.

Creating and authorizing AIS resources

Creating AIS consent

In order to read account details, transactions, balances or initiate payments, TPP needs to get consent from user. First step in doing this is creation of consent resource. To do this TPP has to make call to *POST /v1/*consents endpoint. Request should have payload that is similar to this (for full description of payload and headers refer to Berlin Group NextGen PSD2 Documentation):

```
{
  "access": {
    "availableAccounts": "allAccounts"
  },

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

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

Starting authorization for AIS consent

When consent resource is successfully created, TPP has to make call to: *POST /v1/consents/{consentId}/authorizations* endpoint where *consentId* is id of the consent that was previously created. If the authorization was successfully created, response payload should be similar to this:

```
{
  "scaStatus": "received",
  "authorizationId": "748e51cf66e94f5c97b32777a4ce0813",
  "scaMethods": [
    {
      "authenticationVersion": "1.00",
      "authenticationMethodId": "SCA Method 3",
      "name": "SCA Method 3"
    }
  ],
  "choosenScaMethod": null,
  "_links": {
    "scaRedirect": {
      "href": "https://iam-
sandbox.medirect.psd2enabler.asseco.rs/consent/login?consentId=0ebc4812-2998-4018-
b610-
ead5291134a2&authorizationId=748e51cf66e94f5c97b32777a4ce0813&returnUrl=https://www.go
ogle.com"
    },
    "scaOAuth": {
      "href": "https://iam-sandbox.medirect.psd2enabler.asseco.rs/"
    },
    "scaStatus": {
      "href": "v1/consents/0ebc4812-2998-4018-b610-
ead5291134a2/authorisations/748e51cf66e94f5c97b32777a4ce0813"
    }
  }
}
```


AIS consent authorization using OAuth2 protocol

For authorizing Consent using OAuth2 flow, TPP needs URL from *scaOAuth* field. To start OAuth2 protocol, TPP has to redirect client from its application to IAM application. Endpoint on which user needs to be redirected is *scaOAuth/connect/authorize,* where *scaOAuth* is value of corresponding field in the response of the request for starting the authorization. User should be redirected with following query parameters:

- **Grant Type** (grant_type) **–** This field needs be equal to *code*
- **Response Type** (response_type) – This field needs to be equal to *code*
- **Redirect URL** (redirect_uri) **–** URL on which TPP wants user to be redirected after finishing SCA, should be equal to some of URLs that are provided on TPP application registration under the *Redirect URIs* field
- **Client ID** (client_id)**–** Id of client that was created for TPP on TPP registration
- **Scope** (scope) **–** Scope should have value that equals to AIS:<consentId> where <consentId> should be replaced with id of the consent that we want user to authorize

Redirect URL should be in format similar to this:

https://iam-sandbox.psd2enabler.asseco.rs/connect/authorize?client_id=client.id&scope=AIS:e3bf80a0-996e-47e5-8840-b3b83eaa29ed&redirect_uri=https://www.redirect.com/oauth-callback&grant_type=code&response_type=code

If user has done authentication successfully, user will be redirected to the URI that TPP provided in *redirect_uri* field with following parameters in string format:

- **Code** (code) – one time code that will be used for obtaining access token by TPP
- **Scope** (scope) – scopes that were granted
- **Session State** (sessionState) – this field can be omitted

TPP should make a request for access token in this callback method. This access token will be used as authorization data that is required for consent authorization. To obtain access token, TPP has to send a request to POST *scaOAuth/connect/token/mtls,* with content type *application/x-www-form-urlencoded* and following parameters in request body:

- **Client ID** (client_id) - Id of client that was created for TPP on TPP registration
- **Scope** (scope) – this field should be equal to the scope parameter received in callback
- **Code** (code) – code that was received in callback
- **Redirect URI** (redirect_uri) – redirect URI that was used I */connect/authorize* request
- **Grant Type** (grant_type) – This field needs to be equal to authorization_code

As in guide for TPP application registration, TPP should add certificate that will be used for Mutual TLS. Response body of the successful request will contain access token.

Response example:


```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9l
IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```


Finishing authorization

In order to finish consent authorization, TPP needs to send a request to PUT */v1/consents/{consentId}/authorisations/{authorisationId}*. Consent ID is id of consent that is authorizing, and authorizationId is id of authorization resource that was created for consent authorization.

Request body of this field has to be in application/json format and must contain field “scaAuthenticationData”. Value of this field has to be equal to the access token that was obtained through the OAuth2 protocol.

Request body example:


```
{
  "scaAuthenticationData":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9l
IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
}
```

Creating and authorizing PIS resources

Creating payment resource

To create payment TPP has to make call to *POST /v1/{payment-service}/{payment-product}* endpoint.

Request example (for full description of payload and headers refer to Berlin Group NextGen PSD2 Documentation):


```
{
  "endToEndIdentification": "EI25125767734",
  "debtorAccount": {
    "iban": "MT61MBWM07113430000002143800000"
  },
  "instructedAmount": {
    "currency": "EUR",
    "amount": "10"
  },
  "creditorAccount": {
    "iban": "MT61MBWM07113430000002143800011"
  },
  "creditorName": "John Doe",
  "creditorAddress": {
    "street": "Street",
    "buildingNumber": "1",
    "country": "RS",
    "city": "Belgrade"
  },
  "requestedExecutionDate": "2019-12-30T10:02:29.073Z",
  "requestedExecutionTime": "2019-12-30T10:02:29.073Z",
  "endDate": ""
}
```


As in almost all previous requests, TPP should add certificate that will be used for Mutual TLS.

Starting authorization for payment resource

When payment resource is successfully created, TPP has to make call to: POST */v1/{payment-service}/{payment-product}/{paymentId}/authorisations* endpoint where *paymentId* is id of the payment resource that was previously created. If the authorization was successfully created, response payload should be similar to this:

```
{
  "scaStatus": "received",
  "authorizationId": "fa8cf3568d134adaaf79d7d1f2c0695c",
  "scaMethods": null,
  "choosenScaMethod": null,
  "_links": {
    "scaRedirect": {
      "href": "https://iam-
sandbox.medirect.psd2enabler.asseco.rs/payment?transferId=783867aab4bc439291c6c5e2e6b3
db6f&authorizationId=fa8cf3568d134adaaf79d7d1f2c0695c&returnUrl=https://www.google.com
&abandonUrl=https://www.bing.com"
    },
    "scaOAuth": {
      "href": "https://iam-sandbox.medirect.psd2enabler.asseco.rs/"
    }
  }
}
```


Payment resource authorization using OAuth2 protocol

For authorizing payment resource using OAuth2 flow, TPP needs URL from *scaOAuth field.* To start OAuth2 protocol, TPP needs to redirect client from its application to IAM application. Endpoint on which user needs to be redirected is *scaOAuth/connect/authorize,* where *scaOAuth* is value of corresponding field in the response of the request for starting the authorization. User should be redirected with following query parameters:

- **Grant Type** (grant_type) **–** This field needs be equal to *code*
- **Response Type** (response_type) – This field needs to be equal to *code*
- **Redirect URL** (redirect_uri) **–** URL on which TPP wants user to be redirected after finishing SCA, should be equal to some of URLs that are provided on TPP application registration under the *Redirect URIs* field
- **Client ID** (client_id)**–** Id of client that was created for TPP on TPP registration
- **Scope** (scope) **–** Scope should have value that equals to PIS:<paymentId> where <paymentId> should be replaced with id of the payment resource that we want for user to authorize

Redirect URL should be in format similar to this:

https://iam-sandbox.medirect.psd2enabler.asseco.rs/connect/authorize?client_id=id.client&scope=PIS:783867aab4bc439291c6c5e2e6b3db6f&redirect_uri=https://www.returnurl.com/oauthcallback&grant_type=code&response_type=code

If user has done authentication successfully, user will be redirected to the URI that TPP provided in *redirect_uri* field with following parameters in string format:

- **Code** (code) – one time code that will be used for obtaining access token by TPP
- **Scope** (scope) – scopes that were granted
- **Session State** (sessionState) – this field can be omitted

TPP should make a request for access token in this callback method. This access token will be used as authorization data that is required for payment resource authorization. To obtain access token, TPP has to send a request to POST *scaOAuth/connect/token/mtls,* with content type *application/x-www-form-urlencoded* and following parameters in request body:

- **Client ID** (client_id) - Id of client that was created for TPP on TPP registration
- **Scope** (scope) – this field should be equal to the scope parameter received in callback
- **Code** (code) – code that was received in callback
- **Redirect URI** (redirect_uri) – redirect URI that was used I */connect/authorize* request
- **Grant Type** (grant_type) – This field needs to be equal to authorization_code

As in guide for TPP application registration, TPP should add certificate that will be used for Mutual TLS. Response body of the successful request will contain access token.

Response example:


```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9l
IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```


Finishing authorization

In order to finish payment resource authorization, TPP has to send a request to PUT */v1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}*. Payment id is id of payment resource that is authorizing, and *authorization id* is id of authorization resource that was created for payment resource authorization.

Request body of this field has to be in *application/json* format and must contain field *scaAuthenticationData*. Value of this field has to be equal to the access token that was obtained through the OAuth2 protocol.

Request body example:


```
{
  "scaAuthenticationData":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9l
IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
}
```

4. AIS and PIS resources authorization using the Redirect flow

Goal

Goal of this section is to show how to authorize AIS and PIS resources using the Redirect SCA flow.

Authorizing AIS consent

Creating consent resource

Consent resource should be created the same way as in the third section of this guide, except that now TPP has to provide redirect URL on which it want for user to be redirected after successful consent authorization. Redirect URL has to be provided through the request header on consent creation. Header name is *TPP-Redirect-URI.* Optionally, TPP can provide redirect URL on which it wants for user to be redirected if SCA was unsuccessful. This URL can be provided through the *TPP-Nok-Redirect-URI* header.

Starting authorization for AIS consent

TPP should start authorization the same way as in third section of this guide.

Starting redirect flow

To start redirect flow, TPP should redirect user to URL from *scaRedirect* field in response body of the request for creating authorization resource. This will redirect user to IAM application where user can perform SCA.

Difference between redirect flow and OAuth2 flow is that now TPP won't obtain an access token and doesn't need to send a request to PUT: */v1/consents/{consentId}/authorisations/{authorisationId}.* This is because in redirect flow, IAM application automatically authorizes consent resource when user authenticates itself successfully.

Authorizing payment resource

Creating payment resource

Payment resource should be created the same way as in the third section of this guide, except that now TPP has to provide redirect URL on which it wants for user to be redirected after successful payment authorization. Redirect URL has to be provided through the request header on payment resource creation. Header name is *TPP-Redirect-URI.* Optionally, TPP can provide redirect URL on which it wants for user to be redirected if SCA was unsuccessful. This URL can be provided through the *TPP-Nok-Redirect-URI* header.

Starting authorization for payment resource

TPP should start authorization the same way as in third section of this guide.

Starting redirect flow

To start redirect flow, TPP should redirect user to URL from *scaRedirect* field in response body of the request for creating authorization resource. This will redirect user to IAM application where user can perform SCA.

Difference between redirect flow and OAuth2 flow is that now TPP won't obtain an access token and doesn't need to send a request to PUT */v1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}*.

This is because in redirect flow, IAM application automatically authorizes payment resource when user authenticates itself successfully.

Support: support.api@otpbanka.hr