User Guide for a TPP Developer

TBI BANK

Table of Contents

Developer Portal User Interface

Use the menu items in the Developer Portal user interface to register applications, work with APIs, and obtain support.

It is advised to use Chrome browser when using the Developer portal.

Browse the available APIs

You can browse available APIs and their code snippets through the Sandbox Developer Portal - https://dev.tbibank.bg/tbibank-sb/psd2. For the Production Developer Portal the address is https://dev.tbibank.bg/tbibank/psd2. Code snippets are sample code fragments that are generated automatically from REST APIs. They demonstrate how an API consumer can invoke and use an API operation.

From the Developer Portal menu, click **API Products**.

Click on any of the displayed products, then click the API that you are interested in to see the following information:

- Additional documentation on the selected API;
- Security scheme that is used for application authentication to access the API;
- Operations that belong to the API;
- Parameters that can be used to invoke the API;
- Example request and response in several programming languages.

Sign Up

You can create new developer accounts in the Developer Portal.

From the Developer Portal menu, click **Create account**.

Fill in the displayed form fields and submit it. You will receive an activation link to the provided email, which is valid for 48 hours. Click the activation link to complete the registration. Then you can log in to your developer account.

Organization

You must log in to view this page.

You can preview, add new organization, edit organization name and delete organization.

From the Developer Portal menu, click **Organization / My organization**.

View organization

In the **Manage** tab, you can preview organization along with a list of organization members.

In the **Analytics** tab, you can preview API statistics for the last 30 days and last 100 API calls.

Edit organization

From the ellipsis menu, click **Edit organization name**.

Enter the new name for your Developer organization in the Organization name field, and then click Submit button to change the name of your Developer organization, or Cancel button to cancel editing.

Invite your developers

You must be administrator or be granted with permission to view this page.

If you have permission, you can add users to your Developer organization. Those users can then access the Developer Portal and use the APIs that have been made available to the Developer organization.

From the **Members** of the organization list, click **Invite**.

Enter the email address of the new user in the Email address field. Assign a role to the new user. The available roles are:

- Administrator - Can create and edit organizations, create and edit applications and subscribe to Plans;
- Developer - Can create and edit applications and subscribe to Plans;
- Viewer - Can only view applications and application activity.

Click Submit button to invite the new user.

My account

You must log in to view this page.

From the Developer Portal menu, click **My account**.

View

From the left-hand side menu, click **View** if it is not selected by default.

Here you can preview you user data.

Edit

From the left-hand side menu, click **Edit**.

Here you can change multiple fields of your account that are enabled for edit.

Click Save button if you want to save changes or click Cancel button to ignore them.

Delete

At the bottom of your account edit form, you can find a **Delete account** button. If you want to delete your account in the Developer portal you can click it. After this you will not be able to log in with this account.

Change password

From the left-hand side menu, click **Change Password**.

To change the password of your account you must fill in:

- Current password;
- Password – your new password;
- Confirm password – type your new password again.

The new password must comply with the described password policy.

Click Submit button to change the password.

Register an application

You must log in to view this page.

You can preview, add new application, edit application name and delete application.

You must register an application and subscribe it to a plan in the Developer portal, before you can use an API.

From the Developer Portal menu, click **Apps**.

Create new application

To add new application, click **Create new app** button.

Fill in the fields in the Create a new application form.

In the Certificate field paste the public key of your QWAC. For the sandbox a test Qualified Website Authentication Certificates is provided to download and the password for the private key is 1234.

Click Submit button to create the application. Click Cancel button to ignore creation.

When you register an application you are provided with an API Key and Secret for the application. API Key is the Client ID of the application and may be referred as such further. You must keep these values safe, because you must supply the Client ID when you call an API that requires you to identify your application by using a client ID and client Secret.

Click Continue button to view the application Dashboard, which displays API statistics and API call for the application and the application Subscriptions.

Edit application

From the ellipsis menu, click **Edit**.

Modify required values of the application.

Click Submit button to save new values. Click Cancel button to ignore edit.

Change application icon

From the ellipsis menu, click **Upload image**.

Upload new image or delete the existing one.

Click Submit button to save new value. Click Cancel button to ignore.

Delete application

From the ellipsis menu, click **Delete**.

Click Delete button to delete the application.

When an application is deleted, this operation cannot be undone.

Select a plan

Plans specify the limitations and subscription details of how developers can use API Products.

From the Developer Portal menu, click **API Products**.

Choose a Product or an API from a Product you want to subscribe to and click **Subscribe** button from the plan that is suitable for you. You can preview details of the plan.

Select the application that you want to use with this Plan, and click **Select App** button.

Confirm the application plan subscription with **Next** button and complete the action.

Support

You must log in to view this page.

If you experience problems with API consumption while testing them, you can contact our support.

From the footer of the page click **Contact Us**.

Fill in the required fields and describe the problem that you have.

Click Send message button to submit the question.

Sandbox

The sandbox helps developers to become familiar with the published APIs and facilitates the preparation for API usage in product environment.

Sandbox follows BISTRA Standard and current API implementation and services that exist are based on 1.2 release of BISTRA.

The API sandbox consists of static mocked data for

- Account Information Services (AIS) as defined by article 67 in the PSD2 Directive;
- Payment Initiation services (PIS) as defined by Article 66 in the PSD2 Directive;
- Confirmation on the availability of funds (CoF) as defined by Article 65 in the PSD2 Directive;
- Consent services to create consent.

Please note that the information is not real-time at this point.

The communication between the TPP and the ASPSP is always secured by using TLS version 1.2 or higher.

No signing of the request data is mandated in the current implementation of the sandbox.

The sandbox uses following accounts:

IBAN	Currency	Comment
BG30TBIB93101060018902	BGN	Debtor account in BGN
BG98TBIB93104460018901	EUR	Debtor account in EUR
BG63TBIB93104160018901	USD	Debtor account in USD
BG65BANK99991111111111	BGN	
DE68640901001111111111	EUR	
TR4811111012345678912345 67	USD	

Account Information Services

Balances and fixed number of transaction are predefined on following PSU accounts:

IBAN	Currency	Resource Id
BG30TBIB93101060018902	BGN	3dc3d5b3-7023-4848-9853-f5400a64e80f
BG98TBIB93104460018901	EUR	3dc3d5b3-7023-4848-9853-c4511b46f08e

When the current date is in the period of the requested transactions list, then transactions in the response will be returned, otherwise it will be empty.

Payment Initiation services

When you initiate payments following accounts may be used:

IBAN	Currency	Comment
BG30TBIB93101060018902	BGN	Debtor account in BGN
BG98TBIB93104460018901	EUR	Debtor account in EUR
BG63TBIB93104160018901	USD	Debtor account in USD

Initiated payments will be stored in the sandbox for maximum 4 hours.

There are predefined examples of initiated payment requests for the supported payment products:

Product	Resource Id
domestic-credit-transfers-bgn	1234-rtgsct-983
domestic-budget-transfers-bgn	1234-bgnbdg-983
sepa-credit-transfers	1234-wertiq-983
cross-border-transfers	1234-cbdtra-983

Confirmation on the availability of funds

When you make request of this kind following fields are not allowed:

- cardNumber;
- pan.

If you make request for amount, regardless of the currency, less than or equal to 500, then funds are available, otherwise funds are not available.

Consents

There is a predefined resource for a consent with resource id: 1234-consent-983.

Additional required parameters by the ASPSP

Service	Parameter Name	Location	Note
CoF	Consent-ID	Header	
CoF	PSU-ID	Header	Username that PSU user to log in to

			ASPSP's internet banking
PIS	PSU-ID	Header	Username that PSU user to log in to ASPSP's internet banking
AIS	PSU-ID	Header	Username that PSU user to log in to ASPSP's internet banking
Consent	PSU-ID	Header	Username that PSU user to log in to ASPSP's internet banking

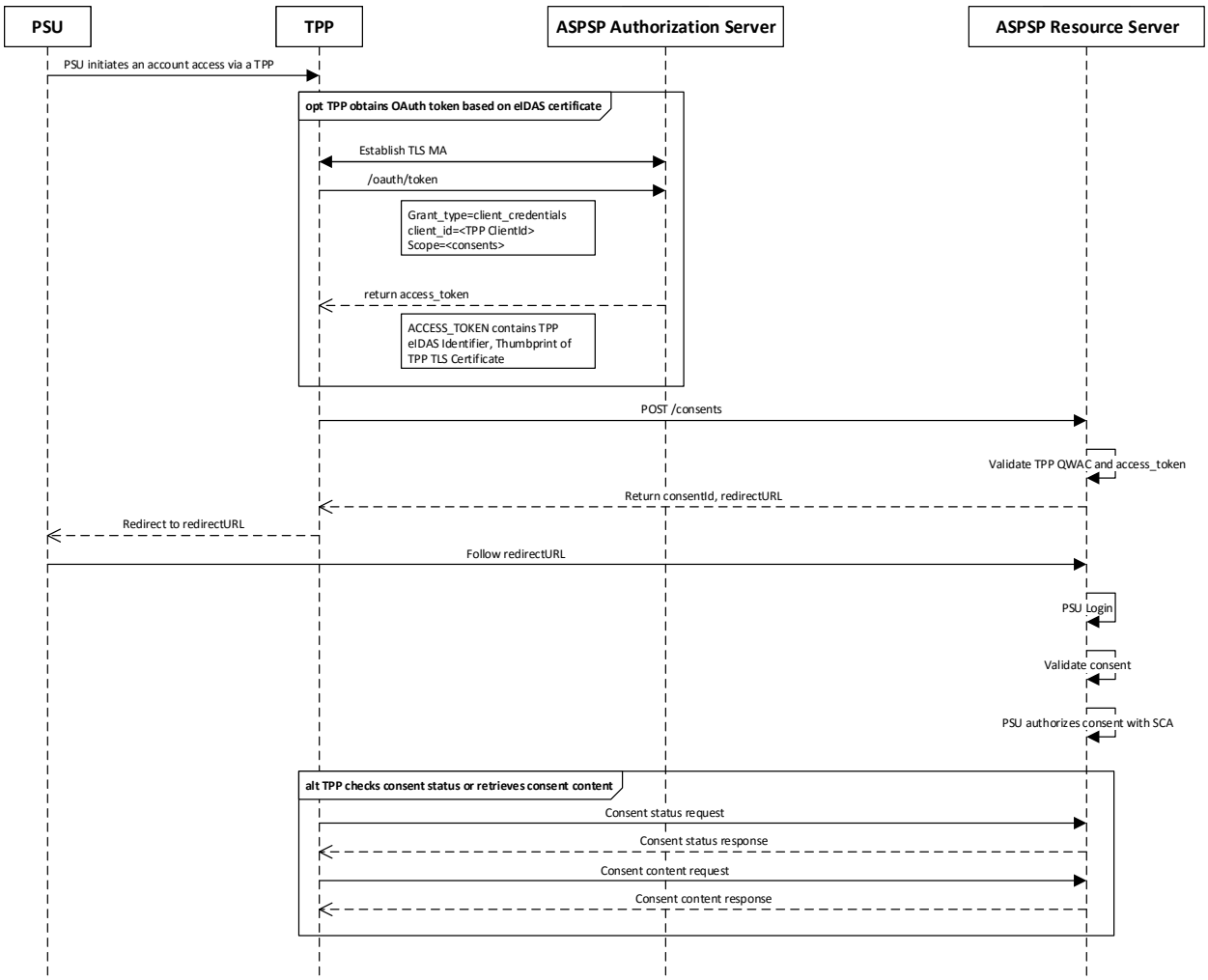
SCA Methods

Supported SCA methods by the ASPSP are:

- redirect.

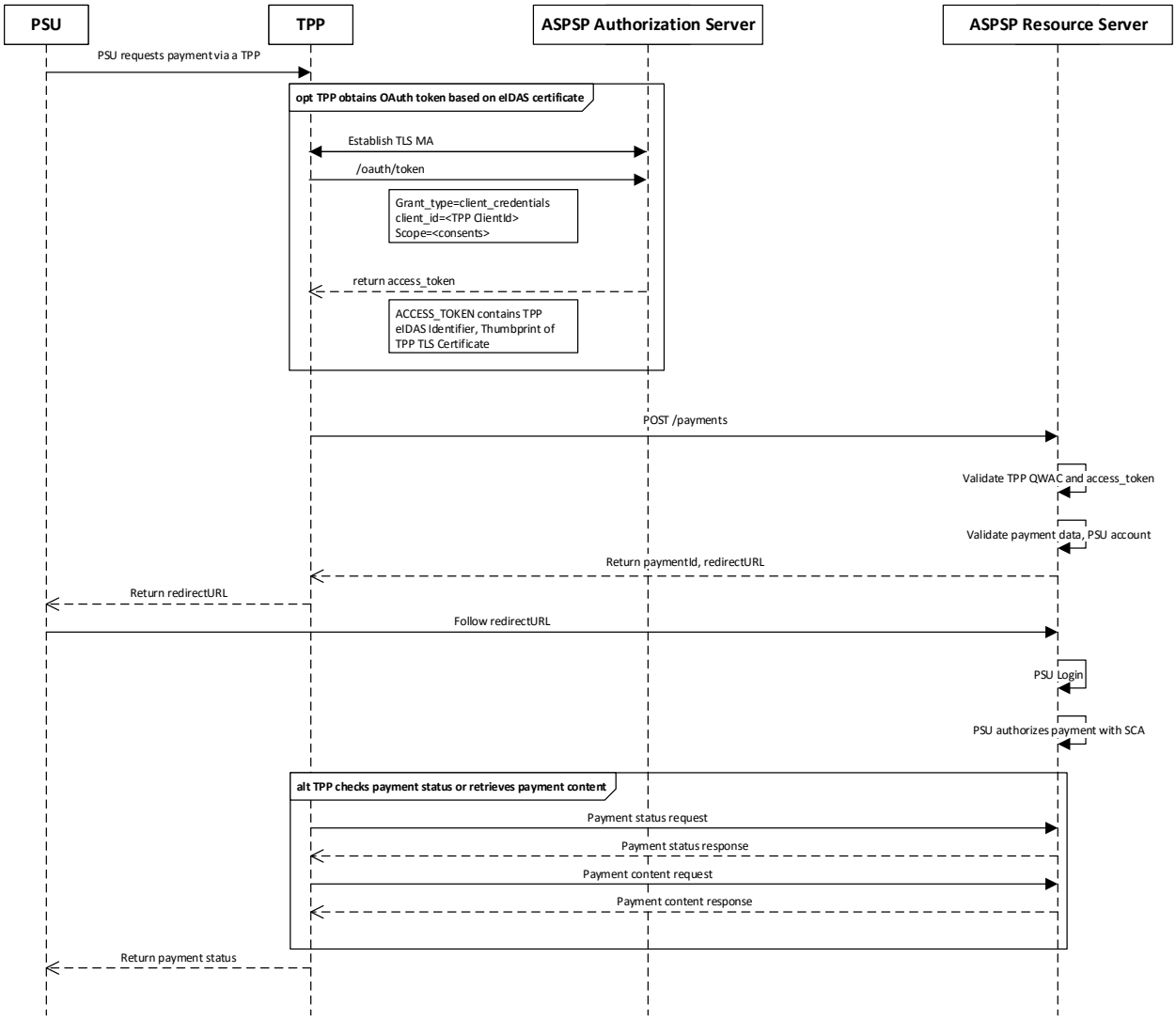
Flows

Establish Consent



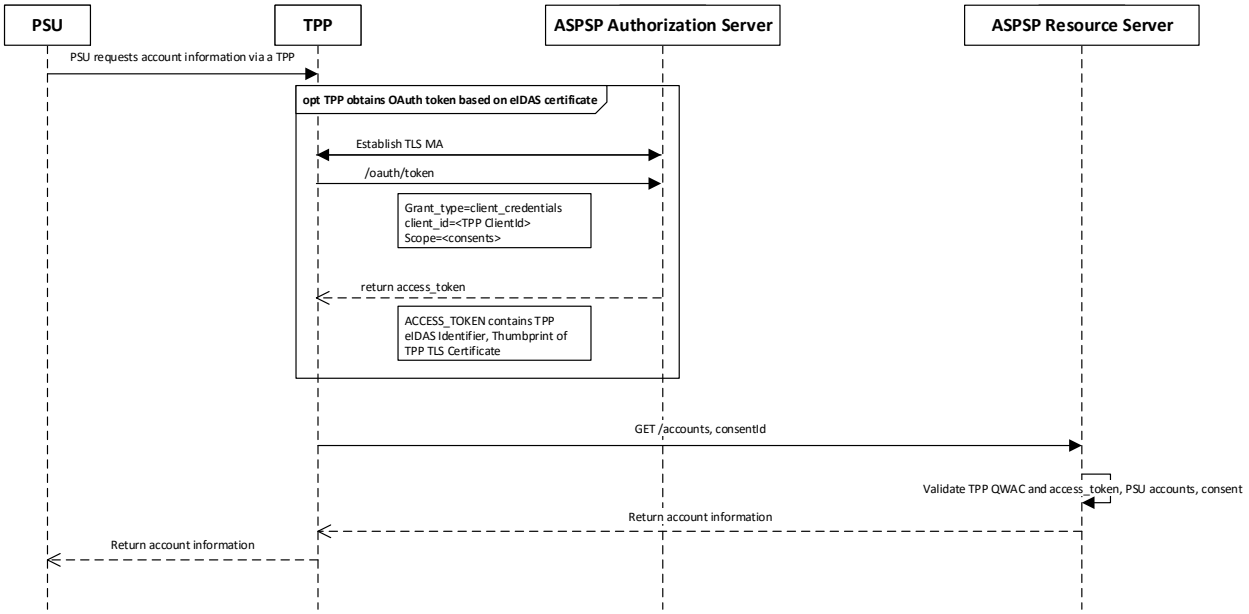
1. PSU initiates an account access via a TPP;
2. The TPP obtains access token from ASPSP Authorization Server following OAuth Client Credentials Flow based on establishing mutual TLS connection;
3. The TPP connects to the ASPSP that services the PSU's payment account and creates a new consent resource with POST request. The ASPSP responds with an identifier for the consent resource and a redirect url;
4. The TPP redirects the PSU to the ASPSP;
5. The ASPSP authenticates the PSU;
6. PSU authorizes consent with SCA;
7. The ASPSP updates the status of the consent resource internally to indicate that the consent has been authorized;
8. The TPP can check the status of the consent and the consent resource (with the ConsentId).

Payment Initiation



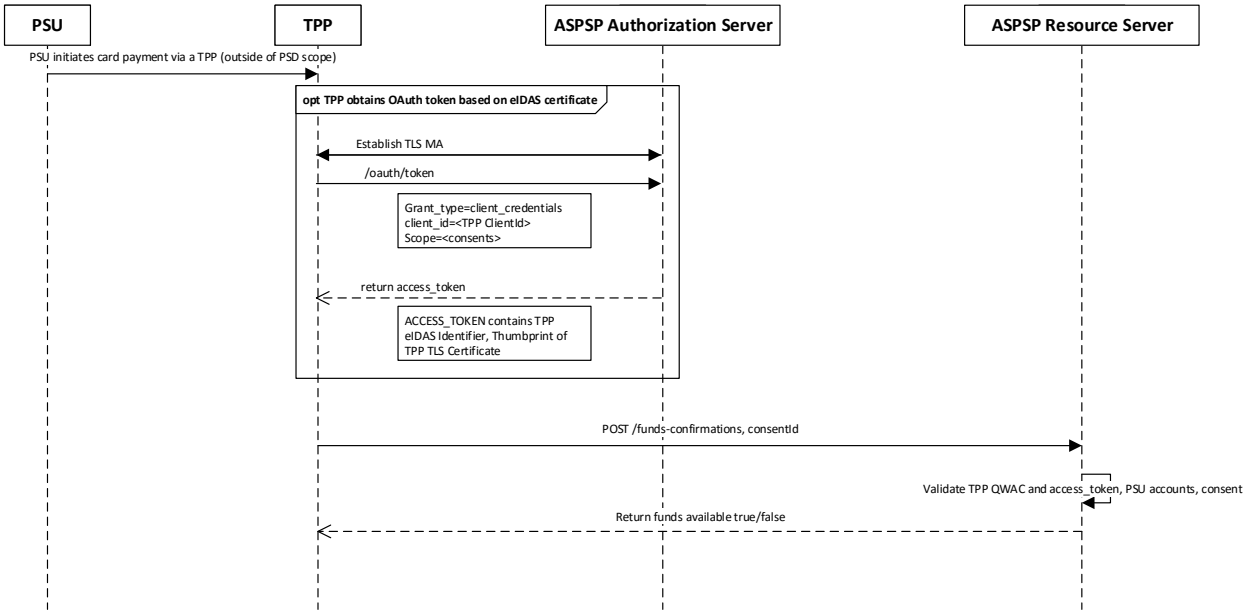
1. PSU requests payment via a TPP;
2. The TPP obtains access token from ASPSP Authorization Server following OAuth Client Credentials Flow based on establishing mutual TLS connection;
3. The TPP connects to the ASPSP that services the PSU's payment account and creates a new payment initiation resource with POST request. The ASPSP responds with an identifier for the payment initiation resource and a redirect url;
4. The TPP redirects the PSU to the ASPSP;
5. The ASPSP authenticates the PSU;
6. PSU authorizes payment initiation with SCA;
7. The ASPSP updates the status of the payment resource internally;
8. The TPP can check the status of the initiated payment and the payment resource (with the PaymentId);
9. The TPP updates payment status to the PSU.

Account Information



1. PSU requests account information via a TPP;
2. The TPP obtains access token from ASPSP Authorization Server following OAuth Client Credentials Flow based on establishing mutual TLS connection;
3. The TPP connects to the ASPSP that services the PSU's payment account and requests account information;
4. The ASPSP checks account and consent (with the ConsentId);
5. The ASPSP returns account information to the TPP;
6. The TPP updates account information to the PSU.

Funds confirmation



1. PSU initiates card payment via a TPP (outside of PSD scope);
2. The TPP obtains access token from ASPSP Authorization Server following OAuth Client Credentials Flow based on establishing mutual TLS connection;
3. The TPP connects to the ASPSP that services the PSU's payment account and requests confirmation on funds availability of an account;
4. The ASPSP checks account and consent (with the ConsentId);
5. The ASPSP returns a simple true/false response to the TPP.