

Pre-authentication – Open API file

Addressees:

Business Analysts, Project Managers, Developers, Architects, IT

Authors:

David Schneider, Lars Kieffer, Hristo Teliatinov, Marcel Signer

Version, date

Version 1.0, April 12th, 2019

Copyright

© CREALOGIX AG

This document and its content are the property of CREALOGIX AG and may not be copied, reproduced, passed on, or used for any order execution without the written consent of the owner.

Table of Contents

1	Introduction	3
2	Client secret	4
	2.1 Generate Client secret.....	4
3	Using the API.....	5

List of figures

Figure 1 - Create secret.....	4
Figure 2 - First call	5
Figure 3 - BasicAuth	5
Figure 4 - Flowchart.....	6

Document history

Version	Description (remarks)	Date	Author(s)
0.9	First draft	April 9 th , 2019	David Schneider, Lars Kieffer, Hristo Teliatinov, Marcel Signer, Martin Bierkoch
1.0	Version 1.0	April 12 th , 2019	David Schneider, Lars Kieffer, Hristo Teliatinov, Marcel Signer

1 Introduction

This document describes the Pre-authentication Open API file and extends the TPP Documentation (PSD2 API Solution). The Pre-authentication Open API file is not part to Berlin Group.

The document assumes that you have basic knowledge about Payment Services Directive 2 (PSD2) regulation of the European Union, its terminology and use cases. Please refer to the Glossary in the TPP documentation which explains the most important PSD2 terms.

Before you read this document, you should have already read chapter 3 – Pre-authentication in TPP documentation as you will find basic information about the pre-authentication and workflows there.

More information on this API you will find in the Open API file description itself.

2 Client secret

Before a TPP can consume the pre-authentication API, a client secret must be generated in the TPP MM module.

For technical reasons, it is necessary for a TPP to generate a "pre-auth secret" for the API in addition to the WSO2 token. This secret is then used by the OAuth server to secure the corresponding OAuth tokens and must usually be included in the APP generated by the TPP. The creation of the secret is triggered in the TPP management.

2.1 Generate Client secret

In the two UIs (see TPP documentation)

- Register TPP (by TPP)
- Register TPP (by ASPSP)

a "Change Secret" button at the bottom next to the certificate upload button will be implemented. This button will then lead to the window outlined below. A TPP can generate the secret here. It is generated once after clicking the Create button and displayed only once in the field below.

 Add secret

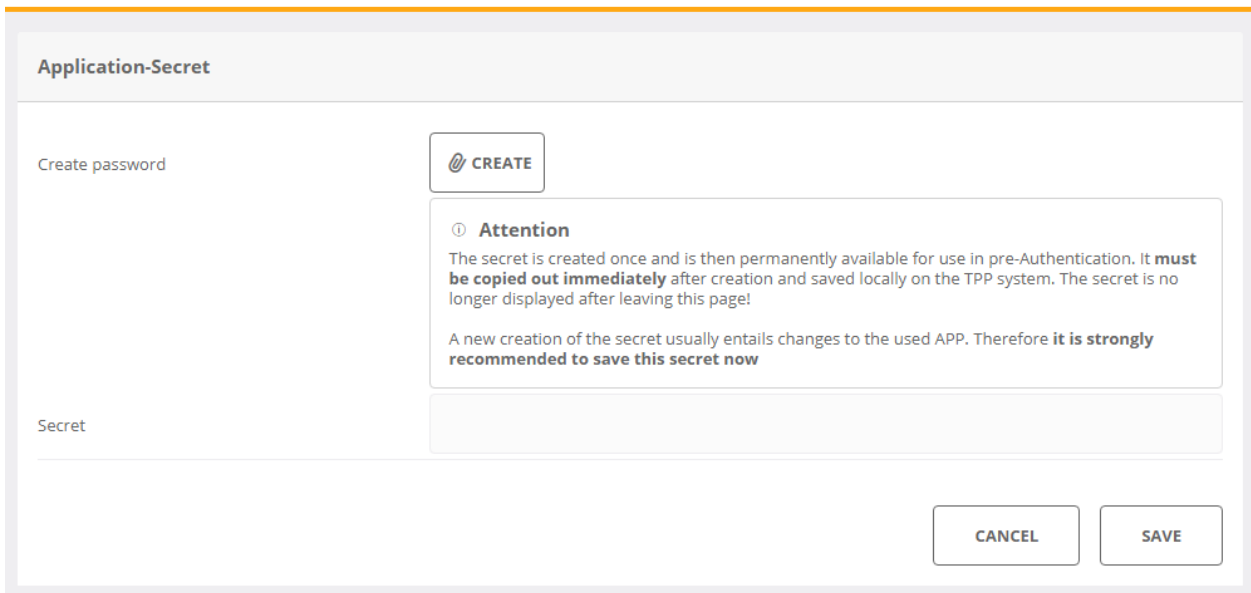



Figure 1 - Create secret

 **The TPP must now copy this secret immediately and store it in a secure place** or integrate it into its APP.

With a click on "Save", the secret is transferred to the OAuth server and serves as a permanent backup of the tokens created there.

 **For security reason this secret will never be displayed again!** A loss can only be repaired by a new creation via this page. However, this leads to a new secret and accordingly to a change in the APP used by the TPP.

3 Using the API

To consume the API the TPP must add the client secret in the first call *POST psd2-auth/v1/auth/token*. This will happen in the basicAuthentication. To get there click on the lock icon on the right side.

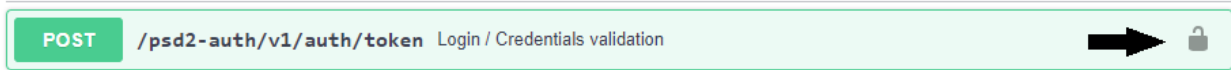


Figure 2 - First call

The username will be the TPP-ID (Authorization number) and the Password will be the client-secret of the TPP Management.

Available authorizations

basicAuth (http, Basic) ←

Basic Authentication. Must contain the clientId and the clientSecret of the TPP

Username:

Password:

Authorize **Close**

Figure 3 - BasicAuth

In the calls course of the pre-Auth API, the TPP then receives a session cookie. The session cookie is equal to the "login-info", which is displayed in the flows of the TPP documentation.

The generated psd2-access-token will be valid for a short time (depending on the backend, up to 15 minutes) like in the online banking. This token will automatically extend its validity if the TPP continues to consume the API. A TPP should be able to process several calls at once if the idle time stays below the token-lifetime.

Using the delete call of the API, a TPP can disconnect the connection if wished. Otherwise it is terminated automatically via the timeout.

The other methods were already explained in the TPP documentation flows. Additionally, you'll find a flowchart here for better explanation.

For technical reasons, the API calls return the same result record. Depending on the situation, however, only the currently relevant parameters are filled with content.

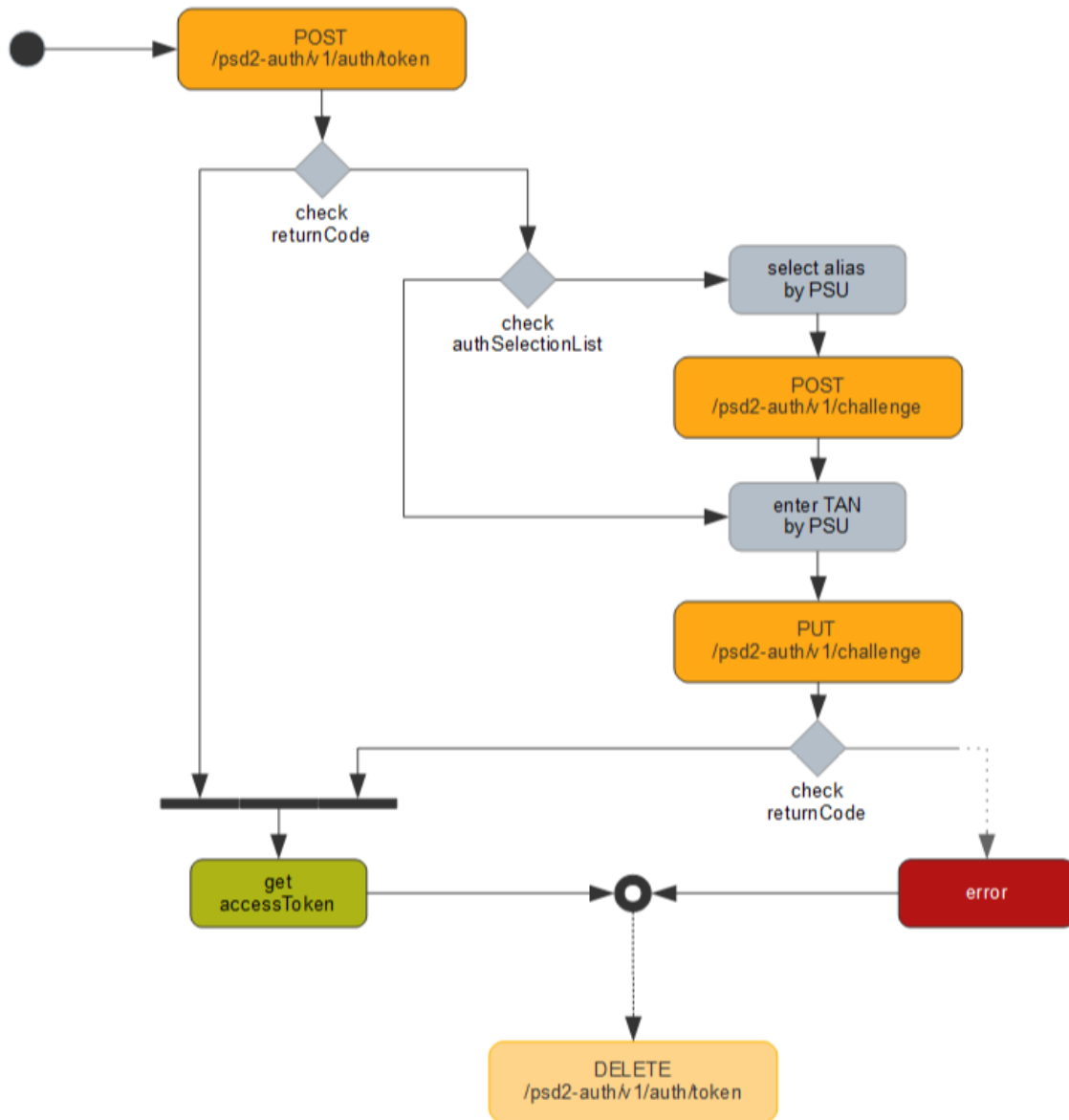


Figure 4 - Flowchart