Sopra Financial Technology

Sparda Sandbox API

OAuth Description

Version 3.0.3, 23.08.2021

Inhalt

1 Summary

This documentation describes the Sparda XS2A API. It is based on the **NextGenPSD2 Framework Version 1.3.3**, provided by the Berlin Group.

2 OAuth2 protocol

To perform requests to the Sparda XS2A API a valid access token is necessary. You get this token from the Identity-Provider (IDP) – the implementation is based on the OAuth2 protocol of the Berlin Group specification.

2.1 Authorization

You will receive the authorization code within the process of a consent or payment initiation or deletion. The response contains a link, which you have to use for the redirection of the PSU to our Identity Provider (IDP).

2.1.1 Request

Request for the XS2A API (e.g.):

- POST https://api-mock.sparda.de/mock/3.0.0/v1/consents
- POST https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers
- DELETE https://api-mock.sparda.de./mock/3.0.0/v1/payments/sepa-credit-transfers

.

It is important that you include the BIC of the test institute (TEST7999) in the customer header "X-BIC" in order to receive the correct institution URL for the IDP in the response.

If you do not know the customer's IBAN when creating a PIS request, you can also omit it. The customer will then be shown the IBAN of his payment transaction account at the IDP. If the customer has more than one payment transaction account, the customer is shown a list with all IBANs of the payment transaction accounts and can select the correct IBAN there.

Example:

```
curl --location --request POST
'https://api-mock.sparda.de/mock/3.0.0/v1/consents' \

--header 'X-Request-ID: 1ed55ecc-0576-4ffb-96a7-5eaa4d83a26d' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
```

```
--header 'PSU-IP-Address: 192.168.1.2' \
--header 'TPP-Redirect-URI: https://tppapi.sopra-ft.com' \
--header 'TPP-Nok-Redirect-URI: https://tppapi.sopra-ft.com' \
--header 'X-BIC: TEST7999' \

--data-raw '{
    "access": {
      "allPsd2": "allAccounts"
    },
    "recurringIndicator": true,
    "validUntil": "2021-12-31",
    "frequencyPerDay": 4
}'
```

2.1.2 Response

The response from your API call will look like:

Example:

```
{
    "consentStatus": "received",
    "consentId":
"SqmB6FkT54mn9x-PKrr7e1OZb-_hwCc19Gr_FG244HIahqxy60x0btK_DtlXq2xdow38ccJ7
EvvDfKGGAsK8a3eGDd8_WXXE6Y4C13Cr9H8=_=_psGLvQpt9Q",
    "_links": {
        "scaRedirect": {
            "href":
"https://idp-mock.sparda-n.de.schulung.sparda.de/oauth2/authorize?bic=TES
T7999&client_id=PSDDE-BAFIN-TEST&redirect_uri=https://tppapi.sopra-ft.com
&response_type=code&scope=AIS:tx-b913226654d43959ad96172dedd292282039551d
5da292fa256fb1f21ab5bc72&code_challenge_method=S256&code_challenge=lRPHsF
D6rWW1zJlodkYWMRdV0K9uY29EXe_L7ZM_SZc"
        },
        "self": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/consents/SqmB6FkT54mn9x-PKrr7e1
OZb-_hwCc19Gr_FG244HIahqxy60x0btK_DtlXq2xdow38ccJ7EvvDfKGGAsK8a3eGDd8_WXX
E6Y4C13Cr9H8=_=_psGLvQpt9Q"
        },
        "status": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/consents/SqmB6FkT54mn9x-PKrr7e1
OZb-_hwCc19Gr_FG244HIahqxy60x0btK_DtlXq2xdow38ccJ7EvvDfKGGAsK8a3eGDd8_WXX
E6Y4C13Cr9H8=_=_psGLvQpt9Q/status"
        },
        "scaStatus": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/consents/SqmB6FkT54mn9x-PKrr7e1
OZb-_hwCc19Gr_FG244HIahqxy60x0btK_DtlXq2xdow38ccJ7EvvDfKGGAsK8a3eGDd8_WXX
E6Y4C13Cr9H8=_=_psGLvQpt9Q/authorisations/cb6b8558-a5cf-4d3f-
be9e-182927344525"
        }
    }
}
```

In the sandbox, the code_challenge is already automatically preassigned in the redirect link. In production the code_challenge still has to be filled in.

The following code_verifier was used as a basis:

„N6WgAgTXVwLUca7mIPIEDmYjUccOqXSJq9Wf95uI1ZFn253J6orTxdUAOW4RxPEO2Ktwe75nKeQpUxZ0vCdLVr4PIzwn8aVcJEZoOjaq4EH4XcBO6Dx1Nt3CzCjp0gyK".

After opening the IDP address the customer has to enter his credentials and after successful login a strong customer authentication is required.

Following the successful strong customer authentication the customer gets redirected to the specified redirect URI – in addition the authorization code is added to the URI. This could look like this:

https://tppapi.sopra-ft.com/?code=tac-2100dda2383032cbd4ea4236bad3dfd021a112cfb41dc04410ca66703e9cfbd9

2.2 Retrieve a new Access token

To retrieve a new Access token from the IDP you have to send a POST request to it – this time to this address: https://idp-mock.sparda.de/oauth2/token. This address is only reachable with an eIDAS certificate. The request sends the obtained authorization code and gets back an Access token and a Refresh token.

2.2.1 Request

The request contains the following parameters:

parameter	mandatory	description	example
grant_type	yes	„authorization_code“ must be specified here	authorization_code
client_id	yes	organizationIdentifier as provided in the eIDAS certificate	PSDDE-BAFIN-TEST
redirect_uri	yes	the exact URI of the TPP where the OAuth2 server redirected the user agent to for this particular transaction	https://tppapp.sopra-ft.com
code	yes	Authorization code from the authorization response	tac-2100dda2383032cbd4ea4236bad3dfd021a112cfb41dc04410ca66703e9cfbd9

code_verifier	yes	PKCE verifier according to cryptographic RFC 7636 (https://tools.ietf.org/html/rfc7636) used to prevent code injection attacks - RegEx for „code_verifier“: [\w\-\._~]{44,127}	N6WgAgTXVwLUca7mIPlEDmYjUccOqXSJq9Wf95uI1ZFn253J6orTxdUAOW4RxPEO2Ktwe75nKeQpUxZ0vCdLVr4PIzwn8aVcJEZoOjaq4EH4XcBO6Dx1Nt3CzCjp0gyK

Example:

```
curl --location --request POST 'https://idp-mock.sparda.de/oauth2/token'
\
--header 'Content-Type: application/x-www-form-urlencoded' \

--data-urlencode 'grant_type=authorization_code' \
--data-urlencode 'code=tac-
2100dda2383032cbd4ea4236bad3dfd021a112cfb41dc04410ca66703e9cfbd9' \
--data-urlencode 'redirect_uri=https://tppapi.sopra-ft.com' \
--data-urlencode 'client_id=PSDDE-BAFIN-TEST' \
--data-urlencode
'code_verifier=N6WgAgTXVwLUca7mIPlEDmYjUccOqXSJq9Wf95uI1ZFn253J6orTxdUAOW
4RxPEO2Ktwe75nKeQpUxZ0vCdLVr4PIzwn8aVcJEZoOjaq4EH4XcBO6Dx1Nt3CzCjp0gyK'
```

2.2.2 Response

The response contains the necessary Access token as well as a Refresh token.

Example:

```
{
    "access_token": "tat-
c18ae95315e414bcd25532cccbd1063f88324e1c2e2dde978e0fb2dff20c32a4",
    "refresh_token": "trt-
ef29cc3836477d3f1b61fd0d7eed49da631746b2d37b2f743c7df4b6a9223bb7",
    "scope": "AIS:tx-
b913226654d43959ad96172dedd292282039551d5da292fa256fb1f21ab5bc72",
    "token_type": "Bearer",
    "expires_in": 300
}
```

2.3 Access token refreshen

An Access token is valid for a maximum of 5 minutes. After that time it can no longer be used to call the XS2A API. The validity of the Access token can be recognized in the payload (field: exp) of the Access token.

With the help of the additionally received Refresh token a new Access token can be generated.

For this purpose you have to send a new request to https://idp-mock.sparda.de/oauth2/token.

2.3.1 Request

The request contains the following parameter:

parameter	mandatory	description	example
grant_type	yes	„refresh_token“ must be specified here	refresh_token
client_id	yes	organizationIdentifier as provided in the eIDAS certificate	PSDDE-BAFIN-TEST
refresh_token	yes	post the Refresh token that was issued together with the expired Access token	trt-ef29cc3836477d3f1b61fd0d7eed49da631746b2d37b2f743c7df4b6a9223bb7

Example:

```
curl --location --request POST 'https://idp-mock.sparda.de/oauth2/token'
\

--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'X-BIC: TEST7999' \

--data-urlencode 'grant_type=refresh_token' \
--data-urlencode 'refresh_token=trt-
ef29cc3836477d3f1b61fd0d7eed49da631746b2d37b2f743c7df4b6a9223bb7' \
--data-urlencode 'client_id=PSDDE-BAFIN-SOPRA-FT'
```

2.3.2 Response

The response contains a new Access token and a new Refresh token.

Example:

```
{
    "access_token": "tat-
6d7a99938cb932bd694207fd19c72521aabcee706a964acef3a31b2370804242",
    "refresh_token": "trt-
fd1ab658a40c68d79a04ca3f3a2740de6127cb8648e2bec18bf245460f33e2cc",
    "scope": "AIS:tx-
b913226654d43959ad96172dedd292282039551d5da292fa256fb1f21ab5bc72",
    "token_type": "Bearer",
    "expires_in": 299
}
```

3 OAuth2 Error Response

In case of error, the error message is appended to the redirect uri instead of the authorization code.

Structure of the error message:

parameter	mandatory	content	description
error	yes	invalid_request unauthorized_client access_denied unsupported_response_type invalid_scope server_error temporarily_unavailable business_error	type of error
error description	no	text	additional information
error_code	no	number	number of error

Example of an error message:

https://tppapp.sopra-ft.com/?error=invalid_request&error_description=Kunden-Authentifizierung+fehlgeschlagen

4 Changelog XS2A API

For version 1.0.0 the following changes were made to the **NextGenPSD2 Framework Version 1.3.3** provided by the Berlin Group

- Remove Endpoints for Signing Baskets
- Remove Endpoints for Common Payments
- Remove Bulk Payments
- Remove Endpoints for POST/PUT authorisation
- Remove CAMT/PAIN/XML Format
- Remove Endpoints for Card Accounts
- Remove all Payment-Products but sepa-credit-transfers
- Adapt general description to fit for Sparda/SFT purposes
- Split POST/GET payment for single payment and periodic payment