Sopra Financial Technology

Sparda XS2A API

OAuth2 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

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

- POST https://api.sparda.de/xs2a/3.0.0/v1/consents
- POST https://api.sparda.de/xs2a/3.0.0/v1/payments/sepa-credit-transfers
- DELETE https://api.sparda.de./xs2a/3.0.0/v1/payments/sepa-credit-transfers

It is important that you include the BIC of the customer 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.sparda.de/xs2a/3.0.0/v1/consents' \

--header 'X-Request-ID: 74cef646-39fd-4640-988c-422870ce6b4a' \
--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: GENODEF1S06' \

--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":
"lHxpyGplqaUKd-RdXFuuWsoyl4Gm-WsVt9UMk5a-if1enq-N38DMVOiHJH2Kh1biMaK3A41L
EI6-AQr7NTEPlneGDd8_WXXE6Y4C13Cr9H8=_=_psGLvQpt9Q",
    "_links": {
        "scaRedirect": {
            "href":
"https://idp.sparda-n.de/oauth2/authorize?bic=GENODEF1S06&client_id=PSDDE
-BAFIN-1923678&redirect_uri=https://tppapi.sopra-ft.com&response_type=cod
e&scope=AIS:tx-b6129fab4a076263f31e659ce5f88ffff6b8743b4d2ffd678b97a2eedd
4d8d9d&code_challenge_method=S256&code_challenge={code_challenge}"
        },
        "self": {
            "href":
"https://api.sparda.de/xs2a/3.0.0/v1/consents/lHxpyGplqaUKd-RdXFuuWsoyl4G
m-WsVt9UMk5a-if1enq-N38DMVOiHJH2Kh1biMaK3A41LEI6-AQr7NTEPlneGDd8_WXXE6Y4C
13Cr9H8=_=_psGLvQpt9Q"
        },
        "status": {
            "href":
"https://api.sparda.de/xs2a/3.0.0/v1/consents/lHxpyGplqaUKd-RdXFuuWsoyl4G
m-WsVt9UMk5a-if1enq-N38DMVOiHJH2Kh1biMaK3A41LEI6-AQr7NTEPlneGDd8_WXXE6Y4C
13Cr9H8=_=_psGLvQpt9Q/status"
        },
        "scaStatus": {
            "href":
"https://api.sparda.de/xs2a/3.0.0/v1/consents/lHxpyGplqaUKd-RdXFuuWsoyl4G
m-WsVt9UMk5a-if1enq-N38DMVOiHJH2Kh1biMaK3A41LEI6-AQr7NTEPlneGDd8_WXXE6Y4C
13Cr9H8=_=_psGLvQpt9Q/authorisations/133aade3-1ca4-4839-93f6-
520ef3d52274"
        }
    }
}
```

The code_challenge still needs to be filled in the redirect link. All other data can be transferred as follows.

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-4a8a7d21cad5664cb3158ed87fe3b0f3fda253269f71cb16484cd0660ff871d5

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.sparda.de/oauth2/token (regardless of the bic). 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-1923678
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-4a8a7d21cad5664cb3158ed87fe3b0f3fda253269f71cb16484cd0660ff871d5
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}	thisisfinallyavalidcodeverifieraslongasthedistancetothemoon

Example:

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

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

--data-urlencode 'grant_type=authorization_code' \
--data-urlencode
'code=tac-4a8a7d21cad5664cb3158ed87fe3b0f3fda253269f71cb16484cd0660ff871d
5' \
--data-urlencode 'redirect_uri=https://tppapi.sopra-ft.com' \
--data-urlencode 'client_id=PSDDE-BAFIN-1923678' \
--data-urlencode
'code_verifier=thisisfinallyavalidcodeverifieraslongasthedistancetothemoo
n'
```

2.2.2 Response

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

Example:

```
{
    "access_token": "tat-
568bc4701722b480739a957ffec1f1da19e603dde17911ffc43f552e8ed8f48b",
    "refresh_token": "trt-
945cfc9677c6a5533f7c1ed89c886fa8bc95b103a37397200a17c378b522338a",
    "scope": "AIS:tx-
b6129fab4a076263f31e659ce5f88ffff6b8743b4d2ffd678b97a2eedd4d8d9d",
    "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 the token 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.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-1923678
refresh_token	yes	post the Refresh token that was issued together with the expired Access token	trt-945cfc9677c6a5533f7c1ed89c886fa8bc95b103a37397200a17c378b522338a

Example:

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

--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'X-BIC: GENODEF1S06' \

--data-urlencode 'grant_type=refresh_token' \
--data-urlencode
'refresh_token=trt-945cfc9677c6a5533f7c1ed89c886fa8bc95b103a37397200a17c3
78b522338a' \
--data-urlencode 'client_id=PSDDE-BAFIN-1923678'
```

2.3.2 Response

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

Example:

```
{
    "access_token": "tat-
2c4958882aad89ce289ac0066d1b8fca176e0ead220203df7a7854a5f5d0fba7",
    "refresh_token": "trt-
8f3245b08e011e02bc84febeb3e08a378c2a8a8f67e077bd99ce9daaabebae87",
    "scope": "AIS:tx-
b6129fab4a076263f31e659ce5f88ffff6b8743b4d2ffd678b97a2eedd4d8d9d",
    "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