Sopra Financial Technology

Sparda Sandbox-API

OAuth Beschreibung

Version 3.0.3, 23.08.2021

Inhalt

1 Zusammenfassung

Dieses Dokument beschreibt die Sparda XS2A-API. Es basiert auf dem **NextGenPSD2 Framework Version 1.3.3**, welche von der Berlin Group bereitgestellt wird.

2 OAuth2-Protokoll (Autorisierung + Token)

Um Konto- oder Payment-Daten abfragen zu können, ist ein gültiger Access-Token notwendig. Die Realisierung wird grundlegend auf Basis des OAuth2-Protokolls der Berlin-Group-Spezifikation durchgeführt.

2.1 Autorisierung

Sie erhalten den Autorisierungscode, wenn Sie über die XS2A-API einen Consent oder ein Payment anlegen bzw. löschen. In der Response des jeweiligen Aufrufs erhalten Sie einen Link, mittels dem Sie den Kunden an den Identity Provider (IDP) weiterleiten.

2.1.1 Request

Aufruf der XS2A –API über z.B.:

- 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

Wichtig ist, dass Sie bei dem Request in dem Customer Header „X-BIC“ die BIC des Testinstitut (TEST7999) mitgeben, um in der Response die richtige Instituts-URL für den IDP zu erhalten.

Sollte Ihnen bei der Erstellung eines PIS-Request die IBAN des Kunden nicht bekannt sein, können Sie diese auch weglassen. Dem Kunden wird dann am IDP die IBAN seines Zahlungsverkehrskontos angezeigt. Falls der Kunde mehr als ein Zahlungsverkehrskonto haben sollte, bekommt der Kunde eine Liste mit allen IBANs der Zahlungsverkehrskonten angezeigt und kann dort die richtige IBAN auswählen.

Beispiel:

```
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

Die Response aus dem API-Call wird folgendermaßen aussehen:

Beispiel:

```
{
    "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 der Sandbox ist im Redirect-Link die code_challenge automatisch schon vorbelegt. In Produktion muss die code_challenge noch befüllt werden.

Folgender code_verifier wurde zugrunde gelegt:

„N6WgAgTXVwLUca7mIPlEDmYjUccOqXSJq9Wf95uI1ZFn253J6orTxdUAOW4RxPEO2Ktwe75nKeQpUxZ0vCdLVr4PIzwn8aVcJEZoOjaq4EH4XcBO6Dx1Nt3CzCjp0gyK".

Nach dem Aufruf des IDP-Links gibt der Kunde im Login-Formular der entsprechenden Sparda-Bank seine Online-Banking-Credentials ein. Anschließend wird er bei korrekter Eingabe zur starken Kundenauthentifizierung weitergeleitet.

Ist diese starke Kundenauthentifizierung erfolgreich, wird der Kunde mittels der Redirect-URI in Ihre Anwendung zurückgeleitet – gleichzeitig wird der Autorisierungscode an die URI angehängt. Dies kann wie folgt aussehen:

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

2.2 Neuen Access-Token abrufen

Um den besagten Access-Token vom IDP zu erhalten muss ein POST-Request an den IDP gesendet werden – dieses Mal an die Adresse https://idp-mock.sparda.de/oauth2/token. Diese Adresse ist nur mit einem eIDAS-Zertifikat erreichbar. Der Request tauscht den erhaltenen Autorisierungscode gegen einen Access-Token und einen Refresh-Token aus.

2.2.1 Request

Der Request enthält folgende Parameter:

Parameter	Pflicht	Beschreibung	Beispiel
grant_type	ja	„authorization_code“ muss hier angegeben werden	authorization_code
client_id	ja	Eindeutiger TPP-Identifier – entspricht dem *organizationIdentifier* aus dem eIDAS-Zertifikat	PSDDE-BAFIN-TEST
redirect_uri	ja	die exakte TPP-URI an die der User Agent nach der Autorisierung weitergeleitet wurde – URI muss also mit der redirect_uri aus der Autorisierung übereinstimmen	https://tppapp.sopra-ft.com
code	ja	Autorisierungscode aus der Autorisierung	tac-2100dda2383032cbd4ea4236bad3dfd021a112cfb41dc04410ca66703e9cfbd9
code_verifier	ja	vom TPP-Client generierter „code_verifier“ (PKCE für OAuth 2.0) – muss mit dem	N6WgAgTXVwLUca7mIPlEDmYjUccOqXSJq9Wf95

		„code_challenge“ aus Autorisierung zusammenpassen (PKCE-Standard) - RegEx für „code_verifier“: [\w\-\._~]{44,127}	uI1ZFn253J6orTxdUAOW4RxPEO2Ktwe75nKeQpUxZ0vCdLVr4PIzwn8aVcJEZoOjaq4EH4XcBO6Dx1Nt3CzCjp0gyK

Beispiel:

```
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

Die Response enthält sowohl den notwendigen Access-Token als auch einen Refresh-Token.

Beispiel:

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

2.3 Access-Token refreshen

Ein Access-Token ist maximal 5 Minuten gültig. Nach dieser Zeit kann mit diesem kein Aufruf mehr gegen die XS2A-API durchgeführt werden. Die Gültigkeit des Access-Tokens ist anhand des Feldes „exp“ im Payload des Access-Tokens erkennbar.

Mit Hilfe des zusätzlich erhaltenen Refresh-Token kann jedoch ein neuer Access-Token generiert werden.

Hierfür ist erneut ein Request gegen https://idp-mock.sparda.de/oauth2/token durchzuführen.

2.3.1 Request

Der Request enthält folgende Parameter:

Parameter	Pflicht	Beschreibung	Beispiel
grant_type	ja	„refresh_token“ muss hier angegeben werden	refresh_token
client_id	ja	Eindeutiger TPP-Identifier – entspricht dem *organizationIdentifier* aus dem eIDAS-Zertifikat	PSDDE-BAFIN-TEST
refresh_token	ja	es muss der Refresh-Token übergeben werden, der gleichzeitig mit dem abgelaufenen Access-Token ausgegeben wurde	trt-ef29cc3836477d3f1b61fd0d7eed49da631746b2d37b2f743c7df4b6a9223bb7

Beispiel:

```
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

Die Response enthält einen neuen Access-Token und einen neuen Refresh-Token.

Beispiel:

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

3 OAuth2 Fehlermeldung

Im Fehlerfall wird anstatt des Autorisierungscode die Fehlermeldung an die Redirect URI angehängt.

Aufbau der Fehlermeldung:

Parameter	Pflicht	Inhalt	Beschreibung
error	ja	Invalid_request unauthorized_client access_denied unsupported_response_type invalid_scope server_error temporarily_unavailable business_error	Fehlerkategorie
error description	nein	Text	Text der Fehlermeldung
error_code	nein	Nummer	Nummer der Fehlermeldung

Beispiel einer Fehlermeldung:

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

4 Changelog XS2A-API

Für Version 1.0.0 wurden die folgenden Änderungen am **NextGenPSD2 Framework Version 1.3.3** (bereitgestellt durch die Berlin Group) durchgeführt

- Endpunkte für Signing Baskets entfernt
- Endpunkte für Common Payments entfernt
- Bulk Payments entfernt
- Endpunkte für POST/PUT authorisation entfernt
- CAMT/PAIN/XML Format entfernt
- Endpunkte für Card Accounts entfernt
- alle Payment-Produkte bis auf sepa-credit-transfers entfernt
- allgemeine Beschreibung an die Sparda/SFT Bestimmungen angepasst
- POST/GET-Aufrufe für single payment und periodic payment aufgeteilt