Sparda Sandbox API

Developer Portal

Version 3.0.19.1, 28.07.2025

Inhalt

1 Introduction

The Payment Service Directive 2 (PSD2) instructed banks to provide a fully productive Access-to-Account (XS2A) interface to Third Party Providers (TPPs) until September 2019. XS2A itself consists of banking services to initiate payments (PIS), request account data (AIS) and get the confirmation of the availability of funds (PIIS). In order to guarantee the compliance of this deadline due to adaptions and bugs, PSD2 claimed the banks to provide a functional sandbox offering the XS2A services in a non-productive environment until March 2019.

Central component of the PSD2 Sandbox is the XS2A interface which meets the requirements of the Berlin Group (Version 1.3) and is based on test data. Besides the actual interface, PSD2 instructs banks to offer a technical documentation free of charge containing amongst others, information about supported payment products and payment services.

Before accessing the XS2A services a TPP needs to register at its National Competent Authority (NCA) and requests an eIDAS certificate (a so-called QWAC) at an appropriate Trust Service Provider (TSP). A QWAC is part of eIDAS and might be better known as X.509 certificate. For PSD2-purposes the certificate gets extended by the QcStatement containing appropriate values such as the role(s) of the PSP (see ETSI).

Getting started:

In order to test the services of XS2A you need to execute the following steps:

- For test QWAC Certificates please contact a TSP and embed the .pem and .key files in your requests.
- The most popular use cases are described in a following section.

Overview of supported institutes:

- Sparda-Bank West (BLZ: GENODED1SPW, GENODED1SPE, GENODED1SPK, GENODEF1S08)

2 Configuration

2.1 Active XS2A Configuration (Bank Profile)

- SCA-Approach: Redirect
- Payment-Types
  - Single (sepa-credit-transfers)
  - Future-Dated (sepa-credit-transfers)
  - Periodic (sepa-credit-transfers)
- Confirmation of Funds: Yes
- Unterstützte Consents
  - Dedicated: Yes
  - Bank-Offered: No
  - Global: Yes
  - Available Accounts: No
- Deaktivierte Features:
  - Signing Basket
  - Bulk Payments
  - Delta-Reports
  - Multi-Level SCA

2.2 Technical Configuration of the XS2A API

setting:

ais:

consentTypes:

bankOfferedConsentSupported: false

globalConsentSupported: true

availableAccountsConsentSupported: false

accountAccessFrequencyPerDay: 5

notConfirmedConsentExpirationTimeMs: 270000

maxConsentValidityDays: 90

accountOwnerInformationSupported: false

redirectLinkToOnlineBanking:

aisRedirectUrlToAspsp: '{host-url}/oauth2/authorize?bic={bic}&client_id={client-id}&redirect_uri={redirect-uri}&response_type={response-type}&scope={scope}&code_challenge_method={code-challenge-method}&code_challenge={code-

```
challenge}'
      transactionParameters:
      availableBookingStatuses:
        - BOOKED
        - PENDING
        - INFORMATION
        - BOTH
        - ALL
      transactionsWithoutBalancesSupported: true
      supportedTransactionApplicationTypes:
        - application/json
    deltaReportSettings:
      entryReferenceFromSupported: false
      deltaListSupported: false
    scaRequirementsForOneTimeConsents:
      scaByOneTimeAvailableAccountsConsentRequired: true
      scaByOneTimeGlobalConsentRequired: true
  pis:
    supportedPaymentTypeAndProductMatrix:
      SINGLE:
        - sepa-credit-transfers
      PERIODIC:
        - sepa-credit-transfers
    maxTransactionValidityDays: 0
    notConfirmedPaymentExpirationTimeMs: 270000
    paymentCancellationAuthorisationMandated: true
    redirectLinkToOnlineBanking:
      pisRedirectUrlToAspsp: '{host-url}/oauth2/authorize?bic={bic}&client_id={client-
id}&redirect_uri={redirect-uri}&response_type={response-
type}&scope={scope}&code_challenge_method={code-challenge-method}&code_challenge={code-
challenge}'
      pisPaymentCancellationRedirectUrlToAspsp: >-
        {host-url}/oauth2/authorize?bic={bic}&client_id={client-id}&redirect_uri={redirect-
uri}&response_type={response-type}&scope={scope}&code_challenge_method={code-challenge-
method}&code_challenge={code-challenge}
```

```
    paymentCancellationRedirectUrlExpirationTimeMs: 270000
  countryValidationSupported: DE
  supportedTransactionStatusFormats:
    - application/json
piis:
  piisConsentSupported: ASPSP_CONSENT_SUPPORTED
sb:
  signingBasketSupported: false
common:
  scaApproachesSupported:
    - REDIRECT
  scaRedirectFlow: OAUTH
  oauthConfigurationUrl: '{host-url}/idp/'
  startAuthorisationMode: auto
  tppSignatureRequired: false
  psuInInitialRequestMandated: false
  redirectUrlExpirationTimeMs: 270000
  authorisationExpirationTimeMs: 270000
  forceXs2aBaseLinksUrl: true
  xs2aBaseLinksUrl: 'https://api.sparda.de/xs2a/3.0.0'
  supportedAccountReferenceFields:
    - IBAN
  multicurrencyAccountLevelSupported: SUBACCOUNT
  aisPisSessionsSupported: false
  checkTppRolesFromCertificateSupported: true
  aspspNotificationsSupported:
    - NONE
  authorisationConfirmationRequestMandated: false
  authorisationConfirmationCheckByXs2a: true
  checkUriComplianceToDomainSupported: true
  tppUriComplianceResponse: REJECT
  psuInInitialRequestIgnored: true
```

3 Use cases with user data

3.1 Simulation of User Authentication (SCA)

SCA is mandated for the initiation and cancellation of payments as well as for the creation of consents. This user authentication is referred to as SCA (Strong Customer Authentication) in the further documentation.

The PSU „PSU Successful“ with IBAN "DE29799905990000603740" and BIC „TEST7999“ is available for the test use cases. For the „Benutzerkennung“ of the PSU in the login-data you have to use „603740“. Password and TAN are not checked, so you can take anything!

3.2 Payment Initiation

In order to execute a payment, replace the Debtor-IBAN in your request with the one of your PSU. To legitimate the payment, use the SCA Redirect Link from the response of the payment initiation.

The following code snippet is an example cURL command which initiates a single payment for PSU "PSU-Successful":

Flow description:

1.) Call of Sandbox XS2A API:

Example-Request:

```
curl --location --request POST
'https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers'
\
--header 'X-Request-ID: 5a151c54-6ff5-47c0-a70d-dea8d7876335' \
--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 'X-BIC: TEST7999' \

--data-raw '{
      "debtorAccount": {
        "currency": "EUR",
        "iban": "DE29799905990000603740"
      },
      "instructedAmount": {
        "currency": "EUR",
        "amount": "612.00"
      },
      "creditorAccount": {
        "currency": "EUR",
```

```
        "iban": "DE67799905990000603767"
      },
      "creditorName": "Max Mustermann",
      "remittanceInformationUnstructured": "Demo-Überweisung in der
Sandbox",
      "requestedExecutionDate": "2021-09-30"
    }'
```

For the customer autorisation the IDP-Link will be returned in the response:

Example-Response:

```
{
    "transactionStatus": "RCVD",
    "paymentId":
"O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmF
YCEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_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=PIS:tx-
e6d1112b002732b2c0c0db0cc80134747f1adb1c22c777405b34fabcc790d90c&code_cha
llenge_method=S256&code_challenge=lRPHsFD6rWW1zJlodkYWMRdV0K9uY29EXe_L7ZM
_SZc"
        },
        "self": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers/
O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmFY
CEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_psGLvQpt9Q"
        },
        "status": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers/
O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmFY
CEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_psGLvQpt9Q/status"
        },
        "scaStatus": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers/
O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmFY
CEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_psGLvQpt9Q/authorisations/487119c6-
6cab-4abe-83d5-3d1f3da0a6dd"
        }
    }
}
```

The code_challenge in the sandbox is pre-filled. The code_verifier for the pre-filled code_challenge is:

„N6WgAgTXVwLUca7mIPlEDmYjUccOqXSJq9Wf95uI1ZFn253J6orTxdUAOW4RxPEO2Ktwe75nKeQ
pUxZ0vCdLVr4Plzwn8aVcJEZoOjaq4EH4XcBO6Dx1Nt3CzCjp0gyK"

2.) Customer Login:

By calling the IDP link from the response, you can authorize the PSU with the login data of the PSU „PSU-Successful“.

Example-Request:

```
curl --location --request GET
'https://idp-mock.sparda-n.de.schulung.sparda.de/oauth2/authorize?bic=TES
T7999&client_id=PSDDE-BAFIN-TEST&redirect_uri=https://tpp.api.sopra-ft.co
m&response_type=code&scope=PIS:tx-e6d1112b002732b2c0c0db0cc80134747f1adb1
c22c777405b34fabcc790d90c&code_challenge_method=S256&code_challenge=lRPHs
FD6rWW1zJlodkYWMRdV0K9uY29EXe_L7ZM_SZc'
```

3.) Display of the payment data at the IDP and confirmation with the TAN of the PSU Successful:

4.) If the entry is successful, the customer is automatically redirected to the APP of the TPP; the authorization code is attached to the URL:

https://tppapi.sopra-ft.com/?code=tac-4d9c367630ff8418bc991a03f44ed2597dcaa8a12f7f1d1c1adddd017bf36268

5.) Calling the token endpoint at the IDP with the authorization code:

Example-Request:

```
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-4d9c367630ff8418bc991a03f44ed2597dcaa8a12f7f1d1c1adddd017bf3626
8' \
--data-urlencode 'redirect_uri=https://tppapi.sopra-ft.com' \
--data-urlencode 'client_id=PSDDE-BAFIN-TEST' \
--data-urlencode
'code_verifier=N6WgAgTXVwLUca7mIPlEDmYjUccOqXSJq9Wf95uI1ZFn253J6orTxdUAOW
4RxPEO2Ktwe75nKeQpUxZ0vCdLVr4PIzwn8aVcJEZoOjaq4EH4XcBO6Dx1Nt3CzCjp0gyK'
```

6.) In the response you receive the Access token and the Refresh token:

Example-Response:

```
{
    "access_token": "tat-
68c780b1b33b4fa91718ee5d8c7e1a039dea8069232efe4f087ebe96232a20ea",
    "refresh_token": "trt-
9b52b0e507c9e05216bac0a17d25d008cdabef8ce04831d375b62d465e122208",
    "scope": "PIS:tx-
e6d1112b002732b2c0c0db0cc80134747f1adb1c22c777405b34fabcc790d90c",
    "token_type": "Bearer",
    "expires_in": 300
}
```

PSU-Description	IBAN	SCA Status	Transaction Status
PSU-Successful	DE29799905990000603740	finalised	ACTC/ACSC*

(*) The Status is depending on the payment-type. A single payment will get "executed" by our mocked backend. A future-dated payment will get executed when the "requestedExecutionDate" is reached. A similar behaviour is implemented for periodic payments which is depending on the "endDate".

3.3 Payment initiation without IBAN

It is also possible to initiate a payment without entering an IBAN. Follow the process description under 3.2 for payment initiation. After successful login the payment account list of the user is displayed. If the PSU has more than one payment account, he must select the preferred one for the payment initiation. Subsequently, the SCA and further steps are performed as described in the previous paragraph.

Beispiel-Request:

```
curl --location --request POST
'https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers'
\
--header 'X-Request-ID: 5a151c54-6ff5-47c0-a70d-dea8d7876335' \
--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 'X-BIC: TEST7999' \

--data-raw '{
      "debtorAccount": {},
      "instructedAmount": {
        "currency": "EUR",
```

```
        "amount": "612.00"
      },
      "creditorAccount": {
        "currency": "EUR",
        "iban": "DE76799905990000603767"
      },
      "creditorName": "Max Mustermann",
      "remittanceInformationUnstructured": "Demo-Überweisung in der
Sandbox",
      "requestedExecutionDate": "2021-09-30"
    }'
```

3.4 Payment Cancellation (Future dated payment or periodic payment)

In order to cancel a payment, insert your Payment-Id in the Delete Payment Endpoint. Payment Cancellation requires an *explicit start of the authorisation process*, which means you need to follow the startAuthorisation link in the response. From then on you can legitimate the payment cancellation by using the SCA Redirect described in the previous section.

Flow description:

1.) Call of XS2A Sandbox API:

Example-Request:

```
curl --location --request DELETE
'https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers/
O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmFY
CEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_psGLvQpt9Q' \

--header 'X-Request-ID: 3bceb0ab-2670-4460-9703-c836293cf46f' \
```

```
--header 'TPP-Redirect-URI: https://tppapi.sopra-ft.com' \
--header 'X-BIC: TEST7999'
```

For the customer authorization the IDP link will be returned in the response:

Example-Response:


```
{
    "transactionStatus": "ACSP",
    "_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=PIS:tx-d634a632ca211edab4669093035f28c293dff626
b83a144868699d044a13c35a&code_challenge_method=S256&code_challenge=lRPHsF
D6rWW1zJlodkYWMRdV0K9uY29EXe_L7ZM_SZc"
        },
        "self": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers/
O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmFY
CEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_psGLvQpt9Q"
        },
        "status": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers/
O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmFY
CEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_psGLvQpt9Q/status"
        },
        "scaStatus": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers/
O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmFY
CEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_psGLvQpt9Q/cancellation-authorisati
ons/23dbcf5c-c9b2-4a47-8d49-3bd8845d5a17"
        }
    }
}
```


2.) Customer Login:

By calling the IDP link from the response, you can authorize the PSU with the login data of the PSU „PSU-Successful.

Example-Request:

```
curl --location --request GET
'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=PIS:tx-79da42d0d3ecd20bcae93bfc60cd16436f203d38
f61b8dc3ad7501df7e8fe73b&code_challenge_method=S256&code_challenge=lRPHsF
D6rWW1zJlodkYWMRdV0K9uY29EXe_L7ZM_SZc'
```

Anmeldung

603740

••••••

Anmelden

3.) Display of the payment data at the IDP and confirmation with the TAN of the PSU Successful:

mobileTAN Freigabe

Überweisung löschen durch Sopra Financial Technology GmbH

Die Überweisung wird gelöscht.

Verwendungszweck	Demo-Überweisung in der Sandbox
Empfängerkonto	DE76799905990000603767
Betrag	612.00 EUR
Ausführungsdatum	2021-09-30
Emfängername	Max Mustermann
Auftraggeberkonto	DE29799905990000603740

mobileTAN

Freigeben

4.) If the entry is successful, the customer is automatically redirected to the APP of the TPP; the authorization code is attached to the URL:

```
https://tppapi.sopra-ft.com/?code=tac-4ec9bb45d4dfae7cb685f8e5b584078b616417872130597f8b1abc9bbc290470
```

5.) Calling the token endpoint at the IDP with the authorization code:

Example-Request:

```
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-4ec9bb45d4dfae7cb685f8e5b584078b616417872130597f8b1abc9bbc29047
0' \
--data-urlencode 'redirect_uri=https://tppapi.sopra-ft.com' \
--data-urlencode 'client_id=PSDDE-BAFIN-TEST' \
--data-urlencode
'code_verifier=N6WgAgTXVwLUca7mIPlEDmYjUccOqXSJq9Wf95uI1ZFn253J6orTxdUAOW
4RxPEO2Ktwe75nKeQpUxZ0vCdLVr4PIzwn8aVcJEZoOjaq4EH4XcBO6Dx1Nt3CzCjp0gyK'
```

6.) In the response you receive the Access token and the Refresh token:

Example-Response:

```
{
    "access_token": "tat-
c9822797b090cf3258943bb6f20e1c47eda6ed2169b18d535e7607f0fe0f61b5",
    "refresh_token": "trt-
1ba4717db30ad5ff93d5f783f7faf87cefc04ca40ca1f8572c266ab1013b774b",
    "scope": "PIS:tx-
d634a632ca211edab4669093035f28c293dff626b83a144868699d044a13c35a",
    "token_type": "Bearer",
    "expires_in": 300
}
```

PSU-Description	IBAN	SCA Status	Transaction Status
PSU-Successful	DE29799905990000603740	finalised	CANC*

(*) It is only possible to cancel payments which are not yet executed. Since our mocked backend "executes" single payments directly, only future-dated payments and periodic payments can be deleted.

3.5 Get Payment

In order to get the Payment Data, insert your Payment-Id in the GET Payment Data Endpoint.

Example-Request:

```
curl --location --request GET
'https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers/
O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmFY
CEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_psGLvQpt9Q' \

--header 'X-Request-ID: 02ee5aec-88bd-46d7-a040-b26867d767ab' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'PSU-IP-Address: 192.168.1.2' \
--header 'Authorization: Bearer tat-
68c780b1b33b4fa91718ee5d8c7e1a039dea8069232efe4f087ebe96232a20ea'
```

In order to get the Payment Status, insert your Payment-Id in the GET Payment Status Data Endpoint.

Example-Request:

```
curl --location --request GET
'https://api-mock.sparda.de/mock/3.0.0/v1/payments/sepa-credit-transfers/
O-hbq9zocm9Eu2MX0MUk9q_U3MWZ0wW_Z9EoENjV0cIDJ1HsOUMgfx3GSrJeW4bEC6eRMBmFY
CEFmx6GgWLqs8E8ZcgkupoAptjViDqg52c=_=_psGLvQpt9Q/status' \
--header 'X-Request-ID: 37c3fd04-c38c-4030-8dcd-ee72c4ae69c2' \
--header 'Authorization: Bearer
tat-c9822797b090cf3258943bb6f20e1c47eda6ed2169b18d535e7607f0fe0f61b5'
```

PSU-Description	IBAN	SCA Status	Transaction Status
PSU-Successful	DE29799905990000603740	finalised	ACTC/ACSC*

3.6 Dedicated/Global Consent Creation

To be able to query accounts, it is necessary to create a so-called Dedicated/Global Consent in advance. This is a kind of declaration of consent with a defined validity, which allows you to query accounts for a defined period of time, depending on the type of consent.

Flow description:

1.) Call of Sandbox XS2A API:

Example-Request:

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

--header 'X-Request-ID: 0b5236ef-d0bd-42c9-85ad-2ddd7e66f55b' \
--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
}'
```

For the customer authorization the IDP-Link will be returned in the response:

Example-Response:


```
{
    "consentStatus": "received",
    "consentId":
"SpU4KWjuP-d0LQYhfkOvWbZh0pR9KgbPbVruVLG3avPM6NEvGGs9k8ZrrGTt6SuoEggjY9Wf
5NJUUpjYiwd6Z3eGDd8_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-7bc6419c1fc14dea02945fdeea4c621a0d98371f
fe393f8039ea89ea1faeae9a&code_challenge_method=S256&code_challenge=lRPHsF
D6rWW1zJlodkYWMRdV0K9uY29EXe_L7ZM_SZc"
        },
        "self": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/consents/SpU4KWjuP-d0LQYhfkOvWb
Zh0pR9KgbPbVruVLG3avPM6NEvGGs9k8ZrrGTt6SuoEggjY9Wf5NJUUpjYiwd6Z3eGDd8_WXX
E6Y4C13Cr9H8=_=_psGLvQpt9Q"
        },
        "status": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/consents/SpU4KWjuP-d0LQYhfkOvWb
Zh0pR9KgbPbVruVLG3avPM6NEvGGs9k8ZrrGTt6SuoEggjY9Wf5NJUUpjYiwd6Z3eGDd8_WXX
E6Y4C13Cr9H8=_=_psGLvQpt9Q/status"
```

```
        },
        "scaStatus": {
            "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/consents/SpU4KWjuP-d0LQYhfkOvWb
Zh0pR9KgbPbVruVLG3avPM6NEvGGs9k8ZrrGTt6SuoEggjY9Wf5NJUUpjYiwd6Z3eGDd8_WXX
E6Y4C13Cr9H8=_=_psGLvQpt9Q/authorisations/7460edd1-d638-4072-82c9-302148d
e9799"
        }
    }
}
```


2.) Customer Login:

By calling the IDP link from the response, you can authorize the PSU with the login data of the PSU „PSU-Successful.

Example-Request:


```
curl --location --request GET
'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-7bc6419c1fc14dea02945fdeea4c621a0d98371f
fe393f8039ea89ea1faeae9a&code_challenge_method=S256&code_challenge=lRPHsF
D6rWW1zJlodkYWMRdV0K9uY29EXe_L7ZM_SZc'
```


3.) Display of the consent data at the IDP and confirmation with the TAN of the PSU Successful:

4.) If the entry is successful, the customer is automatically redirected to the APP of the TPP; the authorization code is attached to the URL:

```
https://tppapi.sopra-ft.com/?code=tac-56a36d3ac30879507786424004 9a150b6c905f584a8f8c2b1b25c3aefe841b24
```

5.) Calling the token endpoint at the IDP with the authorization code:

Example-Request:

```
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-56a36d3ac30879507786424004 9a150b6c905f584a8f8c2b1b25c3aefe841b2
4' \
--data-urlencode 'redirect_uri=https://XS2ATEST.produktion.api.sopra-
ft.com' \
--data-urlencode 'client_id=PSDDE-BAFIN-TEST' \
--data-urlencode
'code_verifier=N6WgAgTXVwLUca7mIPlEDmYjUccOqXSJq9Wf95uI1ZFn253J6orTxdUAOW
4RxPEO2Ktwe75nKeQpUxZ0vCdLVr4PIzwn8aVcJEZoOjaq4EH4XcBO6Dx1Nt3CzCjp0gyK'
```

6.) In the response you receive the Access token and the Refresh token:

Example-Response:

```
{
    "access_token": "tat-
09032fbfa220b40e6ac666afdcee1ac563c1294de7b74bfbd52e9d45d66f1208",
    "refresh_token": "trt-
a79f30547fa1762c821cbeb7eb0699ab4e9447f8b5108948f87300b8519c059f",
    "scope": "AIS:tx-
7bc6419c1fc14dea02945fdeea4c621a0d98371ffe393f8039ea89ea1faeae9a",
    "token_type": "Bearer",
    "expires_in": 300
}
```

In order to create a dedicated consent, replace the IBAN in your request with the one of PSU Successful. To legitimate the consent creation, use the SCA Redirect described in the previous section.

The following code snippet is an example cURL command which creates Dedicated Consent for „PSU Successful“:

Example-Request:

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

--header 'X-Request-ID: 729d3568-75a4-4a08-b73c-2911a9718552' \
--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": {
        "accounts": [
            {
                "iban": "DE29799905990000603740",
                "currency": "EUR"
            }
        ],
        "balances": [    {
                "iban": "DE29799905990000603740",
                "currency": "EUR"
            }],
        "transactions": [    {
                "iban": "DE29799905990000603740",
                "currency": "EUR"
            }]
    },
    "recurringIndicator": false,
    "validUntil": "2021-12-31",
    "frequencyPerDay": 4
}'
```

PSU-Description	IBAN	SCA Status	Transaction Status
PSU-Successful	DE29799905990000603740	finalised	ACTC/ACSC*

3.7 Consent Deletion

In order to delete a consent, insert your consent-Id in the Delete Consent Endpoint. For the authorization of the deletion, follow the flow description in 3.6 Dedicated/Global Consent Creation.

Example-Request:

```
curl --location --request DELETE
'https://api-mock.sparda.de/mock/3.0.0/v1/consents/SpU4KWjuP-d0LQYhfkOvWb
Zh0pR9KgbPbVruVLG3avPM6NEvGGs9k8ZrrGTt6SuoEggjY9Wf5NJUUpjYiwd6Z3eGDd8_WXX
E6Y4C13Cr9H8=_=_psGLvQpt9Q' \

--header 'X-Request-ID: 4f6208b3-92ff-48d8-8e6a-a58e5ce66089' \
--header 'PSU-IP-Address: 192.168.1.2' \
--header 'Authorization: Bearer tat-
09032fbfa220b40e6ac666afdcee1ac563c1294de7b74bfbd52e9d45d66f1208'
```

PSU-Description	IBAN	Consent Status
PSU-Successful	DE29799905990000603740	terminatedByTpp

3.8 Get Account Data

In order to get the Account Data, replace the IBAN in your request with the one of PSU Successful.

The following code snippet is an example cURL command which gets all the account data from PSU "PSU-Successful":

Example-Request:

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

--header 'x-request-id: cd6ce532-6829-4db4-a289-fa3b274e1ed0' \
--header 'consent-id:
0uiT_zyWfXTZRaKcrNl7xgOKXWRcYwPQFboiahZ3C1NfkTvWUgzN_Pg1WdAe4vpvvkTCsNVwv
gRgBXBSSZ-H53eGDd8_WXXE6Y4C13Cr9H8=_=_psGLvQpt9Q' \
```

```
--header 'Content-Type: application/json' \
--header 'PSU-IP-Address: 192.168.1.2' \
--header 'Authorization: Bearer tat-
8a57ecbbd5fe21d552e5ceb5d372d9aae6898ad30f6d4e1350c708a279945f6c'
```

The following code snippet is an example response for a successful get account data:

Example-Response:

```
{
    "accounts": [
        {
            "resourceId": "9070c27c2d953c8237c2c803f77ac54026236d0a",
            "iban": "DE29799905990000603740",
            "currency": "EUR",
            "name": "",
            "product": "GIRO",
            "cashAccountType": "CACC",
            "status": "enabled",
            "bic": "TEST7999",
            "usage": "PRIV",
            "_links": {
                "balances": {
                    "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/accounts/9070c27c2d953c8237c2c8
03f77ac54026236d0a/balances"
                },
                "transactions": {
                    "href":
"https://api-mock.sparda.de/mock/3.0.0/v1/accounts/9070c27c2d953c8237c2c8
03f77ac54026236d0a/transactions"
                }
            },
            "ownerName": "Max Mustermann"
        }
    ]
}
```

PSU-Description	IBAN	Consent Status
PSU-Successful	DE29799905990000603740	valid