PSD2 USER DOCUMENTATION

BNP PARIBAS S.A. GERMANY

CONSORSBANK / BNP PARIBAS WEALTH MANAGEMENT – PRIVATE BANKING / DAB BNP PARIBAS

BNP PARIBAS

The bank
for a changing
world

CONTENT

1. CHANGE HISTORY

Version	Remarks	Change Date
1.0	Initial published version	13.03.2019
1.1	Minor edits	14.03.2019
1.2	Minor edits	11.04.2019
1.3	Minor edits	10.05.2019
1.4	Minor edits	11.06.2019
1.5	Minor edits	04.07.2019
1.6	Updated certificate generation and request examples	03.12.2020
1.7	Removed references and links to Consorsbank and BNP Paribas Wealth Management - Private Banking	23.04.2021
1.8	Minor edits	25.05.2021
1.9	New Sandbox implementation	20.07.2021

2 GENERAL INFORMATION / USED ABBREVIATIONS

This guideline documents the BNP Paribas S. A. Germany API interface under the PSD2 directive for its brand DAB BNP Paribas. The manual describes the usage of individual APIs, process flows of individual scopes, and the meaning of the parameters used.

The DAB BNP Paribas PSD2 API and developer info websites offer a set of Application Programming Interfaces (APIs) that gives the ability to use DAB BNP Paribas functionality in third party applications. The developer info websites offer documentations and testing facility for payment initiation, account information and balance check services.

This PSD2 API Implementation is based on Berlin Group Specification 1.3 following the redirect approach.

Developer info for DAB BNP Paribas:

https://b2b.dab-bank.de/xs2a/

Contact DAB BNP Paribas by email: xs2a@dab.com

DAB BNP Paribas profile:

- Bank Identifier Code = DABBDEMM
- National Bank Code = 701 204 00
- xs2a Sandbox URL = https://xs2a-sndbx.dab-bank.de (valid from 14.03.2019)

- xs2a production URL = https://xs2a.dab-bank.de (valid from 14.06.2019, see separate documentation)

Consorsbank is also servicing B2C clients with IBANs of former DAB Bank AG, thus clients with an IBAN with the national bank code 701 204 00 can be either DAB or Consorsbank and BNP Paribas Wealth Management - Private Banking customers. This is the case for both the access to the client web portal and to the XS2A interface.

Therefore, a TPP is required to ask customers with the national bank code 701 204 00 if they are DAB Bank or Consorsbank and BNP Paribas Wealth Management - Private Banking customers, if the TPP wants to use an IBAN to determine the endpoint to be used.

As a service BNP Paribas offers an additional API service for the XS2A interface.

Explanation of frequently repeated terms:

AISP	Account Information Service Provider
ASPSP	Account Servicing Payment Service Provider
PIISP	Payment Instrument Issuer Service Provider
PISP	Payment Initiation Service Provider
PSD2	Payment Services Directive 2
PSP	Payment Service Provider
PSU	Payment Service User
TPP	Third Party Provider
TSP	Trusted Service Provider (for Certificates)

3 TPP AUTHENTICATION / CERTIFICATES

3.1 BASIC INFORMATION

For the authentication of the TPP as a client, the eIDAS-based site authentication certificate will be used. The TPP must also be licensed by the BaFin (German banking regulatory authority) or other national authorities within the EU to provide PSD2 services.

As issuing a real certificate just for testing purposes will create too much effort, the Sandbox acts as a TSP issuing Qualified Website Authentication Certificates (QWAC) valid only for the Sandbox. A QWAC is defined in eIDAS and is implemented as a 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 https://www.etsi.org/deliver/etsi_ts/119400_119499/119495/01.01.02_60/ts_119495v010102p.pdf).

After embedding the QWAC in the actual XS2A request, the role and the signature get validated.

3.2 CREATE A TEST CERTIFICATE

In order to generate a test certificate, our sandbox offers a service call responding with a .pem and .key file. The generated certificate can only be used for the sandbox.

Here an example for it:

```
curl -X POST https://redirect-sndbx.dab-bank.de/api/cert-generator -H 'Content-Type: application/json' -d '{ "roles": [ "PISP", "PIISP", "AISP" ], "authorizationNumber": "87B2AC", "commonName": "XS2A Sandbox", "countryName": "Germany", "domainComponent": "public.corporation.de", "localityName": "Nuremberg", "organizationName": "Fictional Corporation AG", "organizationUnit": "Information Technology", "stateOrProvinceName": "Bayern", "validity": 365 }' | tail -2 | sed -e 's/{/{\n\t/g' | sed -e 's/\:/\:\ /g' | sed -e 's/\,/\,\n\t/g' | sed -e 's/}/\n}/g'
```

The last line helps to format the certificate in an easily usable form. Please also be mindful when copying the example from above, that the dash from the “Content-Type” header is preserved, as an invalid header will not work.

It might be necessary to set the proxy in advance:

```
export https_proxy=http://<proxy address>:<Port>/
```

Here the formatted body from the example from above:

```
{
   "roles":  [
         "PISP",
         "PIISP",
         "AISP"
   ],
   "authorizationNumber": "87B2AC",
   "commonName": "XS2A Sandbox",
   "countryName": "Germany",
  "domainComponent": "public.corporation.de",
  "localityName": "Nuremberg",
  "organizationName": "Fictional Corporation AG",
  "organizationUnit": "Information Technology",
  "stateOrProvinceName": "Bayern",
  "validity": 365
}
```

3.3 USE A TEST CERTIFICATE FROM A TSP

For the time being test certificates are accepted for the sandbox regardless of the issuer. This might change at a later time and will be announced.

4 SANDBOX

4.1 GENERAL SETUP

The DAB BNP Paribas Sandbox aims to make testing for TPPs possible by combining the interface that will be used for the production XS2A interface with dynamic test data as described in detail in chapter TEST DATA.

The new sandbox now includes two new modules to make testing the XS2A requests in a way that is closer to the flow that a customer will follow when creating consents or payments through the XS2A:

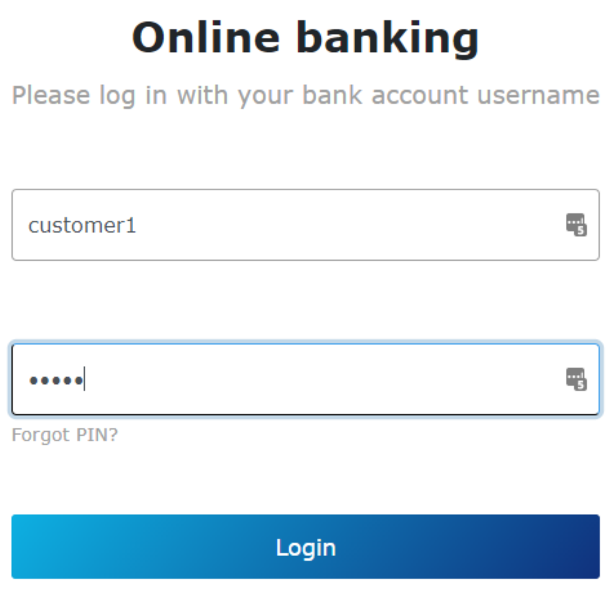
- Banking-UI
  - This is a new UI project, available under the "https://banking-ui-xs2a-sndbx.dab-bank.de " URL
  - Test-accounts can use this page to login into a simplified banking website, to view their accounts, consents and payments or to authorize AIS and PIS requests.
- TPP-UI
  - This is a new UI project, available under the "https://tpp-ui-xs2a-sndbx.dab-bank.de" URL
  - The TPPs will be able to register an account on this page, that can be used to create new and configure existing test-accounts (more details in chapter TEST DATA.

To be able to test the lifecycle of PSD2 use cases, the SCA will be performed by using a browser to navigate to the provided redirect link, which now hosts the Banking-UI module.

4.2 SCA USAGE

Here is a description of the flow, that a test-customer will need to follow, when authorizing the creation of a "Bank-Offered" consent.

Opening the Redirect-URL will lead to the login page of the Banking-UI module, where the PSU has to login using his credentials:



Online banking

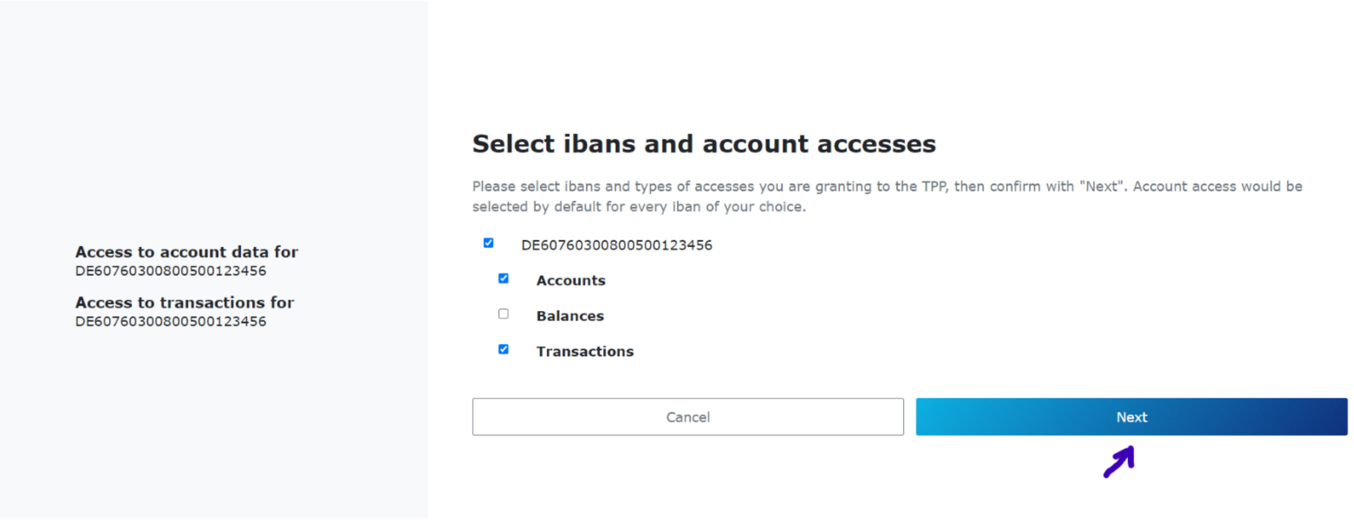
Please log in with your bank account username

customer1

•••••

Forgot PIN?

Just like in the real AIS-flow, after entering his credentials, the customer will see his available accounts (or the consent-details in case of a “dedicated-consent”) where he will have to choose the IBANs and access-types that he wishes to grant to the TPP, after which he will click on the “Next”-button to proceed to the authorization step:



Access to account data for
DE60760300800500123456

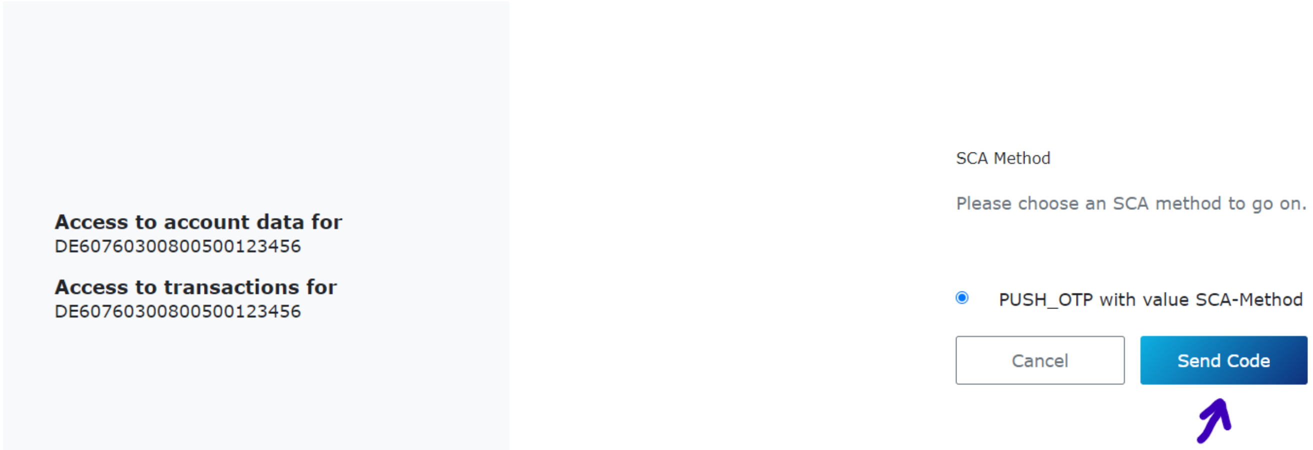
Access to transactions for
DE60760300800500123456

Select ibans and account accesses

Please select ibans and types of accesses you are granting to the TPP, then confirm with "Next". Account access would be selected by default for every iban of your choice.

- ☑ DE60760300800500123456
  - ☑ **Accounts**
  - □ **Balances**
  - ☑ **Transactions**

The first part of the authorization process will be to select the desired SCA method (in this case there is only one available), and request a code by clicking on the “Send Code”-button:



Access to account data for
DE60760300800500123456

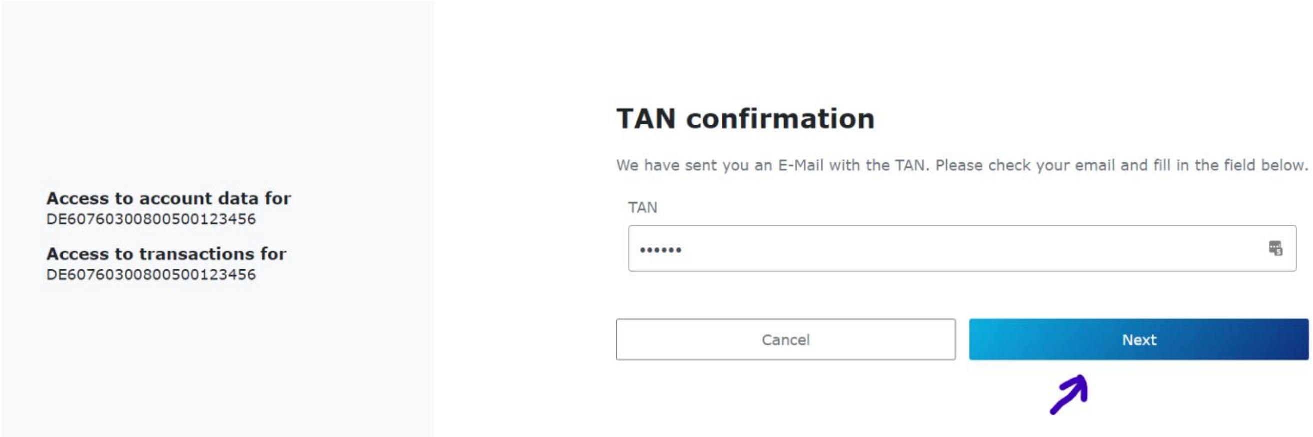
Access to transactions for
DE60760300800500123456

SCA Method

Please choose an SCA method to go on.

● PUSH_OTP with value SCA-Method

Afterwards the customer has to enter his TAN and click on the “Next” -Button:



Access to account data for
DE60760300800500123456

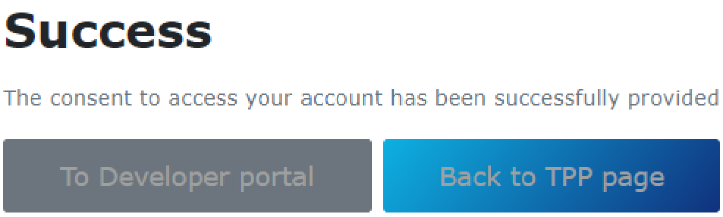
Access to transactions for
DE60760300800500123456

TAN confirmation

We have sent you an E-Mail with the TAN. Please check your email and fill in the field below.

TAN: ••••••

After confirming the TAN, the customer will then be shown the “Success” -Page:



Success

The consent to access your account has been successfully provided

4.3 IBAN SELECTION

Because the new sandbox emulates the actual XS2A implementation, there is a restriction when testing the PIS-flow, where the IBAN for the debtor account has to match a certain format (more details ca be found in chapter TEST DATA. However, the AIS-flow is not impacted by this validation. A difference to the actual XS2A implementation is that all IBANs can be used for the PIIS request in the sandbox (this change was made since it’ s not possible for the TPPs to configure PIIS requests with the current sandbox setup).

5 PAYMENT INITIATION SERVICE (PIS)

5.1 BASIC INFORMATION

This chapter defines a list of services provided for PISP.

Prerequisites:

- The TPP is registered for the PISP role and valid PISP scope
- The TPP has been successfully authenticated

Available Payment Products:

- SEPA Single Payment
- SEPA Future Dated Payment

Exemptions / Specifics

- A change of a SEPA Future Dated Payment needs to be performed by cancelling the existing one and creating a new SEPA Future Dated Payment.
- Payment Cancellation together with PSU until T-1 is only supported for successful authorised Payment Initiations in the case of Future Dated Payment not yet cancelled or executed.
- Payment Cancellations by TPP for Payment Initiations before authorization (therefore in status RCVD) is supported.
- Recurring Payments are not supported.
- Bulk Payments are not supported.
- Signing Basket SCA is not supported.
- If you set the header parameter "Content-Type" with the value "application/json" and your request body of a GET Request is empty, this combination is rated as not valid by our gateway and leads to a 404-Response.
- As mentioned in chapter 4.3, only specific IBANs can be used as the debtor-account (see chapter TEST DATA)

Format:

- JSON

Character Set for SEPA payment content:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9 / - ? : ( ) . , ' + <Space>
```

5.2 ENDPOINT DEFINITION

Method	Endpoint
POST	/v1/{payment-service}/{payment-product}
GET	/v1/{payment-service}/{payment-product}/{paymentId}
DELETE	/v1/{payment-service}/{payment-product}/{paymentId}
GET	/v1/{payment-service}/{payment-product}/{paymentId}/authorisations
GET	/v1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}
POST	/v1/{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations
GET	/v1/{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations
GET	/v1/{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations/{cancellationId}
GET	/v1/{payment-service}/{payment-product}/{paymentId}/status

- Payment-service values: “payments” (single / future dated)
- Payment-product values: “sepa-credit-transfers”
- Redirect SCA Approach: Implicit start of the authorisation process for payment initiation and consent creation; Explicit start of the authorisation process for payment cancellation

5.3 EXAMPLES

The following snippet describes an example for a cURL command to create a SEPA Single Payment for a PSU of the category „PSU-Successful “:

Request:

```
POST https://xs2a-sndbx.dab-bank.de/v1/payments/sepa-credit-transfers

curl -v "https://xs2a-sndbx.dab-bank.de/v1/payments/sepa-credit-transfers" \
  --key private.key --cert certificate.pem --cert-type PEM \
  -H "Content-Type: application/json" \
  -H "accept: application/json" \
  -H "X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721" \
  -H "PSU-IP-Address: 192.168.8.78" \
  -H "tpp-redirect-uri: https://tpp-frontend-redirect.de/“ \
  --cert certificate.pem \
  --key private.key \
  -d '{
    "debtorAccount": {
      "currency": "EUR",
      "iban": "DE37701204003326458001"
    },
    "instructedAmount": {
      "currency": "EUR",
    "amount": "520.00"
  },
  "creditorAccount":{
    "currency": "EUR",
    "iban": "DE15500105172295759744"
  },
  "creditorName": "WBG",
  "remittanceInformationUnstructured": "Ref. Number WBG-1222"
}'
```

Response:

Find below an example response for a successful payment initiation.

```
{
    "transactionStatus": "RCVD",
    "paymentId": "bopPbubjqCvz9qi6JD1k0Oq-Xn_fKPUjzd0QzW2IstQ-6HMwyl6lSS1sGvk3iWNp8N4xvbjPxSBzBKq5VgPnPPSdMWF3876hAweK_n7HJlg=_=_psGLvQpt9Q",
    "transactionFeeIndicator": false,
    "_links": {
        "scaRedirect": {
            "href": "https://redirect-sndbx.dab-bank.de/v1/online-banking/init/pis/cca190c8-ca4c-4fdb-a04c-0934e73e05b7"
        },
        "self": {
            "href": "https://xs2a-sndbx.dab-bank.de/v1/payments/sepa-credit-transfers/bopPbubjqCvz9qi6JD1k0Oq-Xn_fKPUjzd0QzW2IstQ-6HMwyl6lSS1sGvk3iWNp8N4xvbjPxSBzBKq5VgPnPPSdMWF3876hAweK_n7HJlg=_=_psGLvQpt9Q"
        },
        "status": {
            "href": "https://xs2a-sndbx.dab-bank.de/v1/payments/sepa-credit-transfers/bopPbubjqCvz9qi6JD1k0Oq-Xn_fKPUjzd0QzW2IstQ-6HMwyl6lSS1sGvk3iWNp8N4xvbjPxSBzBKq5VgPnPPSdMWF3876hAweK_n7HJlg=_=_psGLvQpt9Q/status"
        },
        "scaStatus": {
            "href": "https://xs2a-sndbx.dab-bank.de/v1/payments/sepa-credit-transfers/bopPbubjqCvz9qi6JD1k0Oq-Xn_fKPUjzd0QzW2IstQ-6HMwyl6lSS1sGvk3iWNp8N4xvbjPxSBzBKq5VgPnPPSdMWF3876hAweK_n7HJlg=_=_psGLvQpt9Q/authorisations/cca190c8-ca4c-4fdb-a04c-0934e73e05b7"
        }
    }
}
```

Status Values:

Used status values for payment status:

- RCVD received
- CANC cancelled
- RJCT rejected
- ACSC AcceptedSettlementCompleted
  (debtor account booked - in case of a SEPA single payment)
- ACTC AcceptedTechnicalValidation
  (in case of a SEPA future dated payment)

More details (e.g., http error codes) you will find in the .yaml file also provided for the download on our developer info site:

https://b2b.dab-bank.de/xs2a/

6 ACCOUNT INFORMATION SERVICE (AIS)

6.1 BASIC INFORMATION

This chapter defines a list of services provided for AISP.

Prerequisites:

- The TPP is registered for the AISP role and valid AISP scope
- The TPP has been successfully authenticated

Available Consent Models:

- Dedicated IBAN / Dedicated List of IBANs
- Bank Offered Consent

Exemptions / Specifics

- Supported transaction type values are BOOKED, PENDING and BOTH
- Supported balance type value is CLOSING BOOKED
- Multicurrency accounts are not supported, all currency accounts are operated at a stand-alone IBAN

Format:

- JSON

6.2 ENDPOINT DEFINITION

Method	Endpoint
POST	/v1/consents
GET	/v1/consents/{consentId}
DELETE	/v1/consents/{consentId}
GET	/v1/consents/{consentId}/authorisations
GET	/v1/consents/{consentId}/authorisations/{authorisationId}
GET	/v1/consents/{consentId}/status
GET	/v1/accounts
GET	/v1/accounts/{account-id}
GET	/v1/accounts/{account-id}/balances
GET	/v1/accounts/{account-id}/transactions
GET	/v1/accounts/{account-id}/transactions/{resource-id}

Redirect SCA Approach: Implicit Start of the Authorisation Process

Account ID: Technical key as returned in attribute resourceId in Request GET /v1/accounts

6.3 EXAMPLES

Request for a dedicated consent:

```
POST https://xs2a-sndbx.dab-bank.de/v1/consents
```

The following snippet describes a sample cURL command to create a **dedicated consent** for psu-id “PSU-Successful”.

```
curl -X POST https://xs2a-sndbx.dab-bank.de/v1/consents \
  --key private.key --cert certificate.pem --cert-type PEM \
  -H 'accept: application/json' \
  -H 'X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721' \
  -H 'Content-Type: application/json' \
  -H 'psu-ip-address: 192.168.8.78' \
  -H 'tpp-redirect-uri: https://tpp-frontend-redirect.de/' \
  -d '{
  "access": {
    "accounts": [
      {
        "iban": "DE74701204003326455007",
        "currency": "EUR"
     }
    ],
    "balances": [
     {
        "iban": "DE94701204003326456005",
        "currency": "EUR"
     }
    ],
    "transactions": [
      {
        "iban": "DE17701204003326457003",
        "currency": "EUR"
    }
    ]
},
  "recurringIndicator": true,
  "validUntil": "2019-05-31",
  "frequencyPerDay": 4
}'
```

Request for a bank offered consent:

```
POST https://xs2a-sndbx.dab-bank.de/v1/consents
```

The following snippet describes a sample cURL command to create a **bank offered consent** for psu-id “PSU-Successful”.

```
curl -X POST \
  https://xs2a-sndbx.dab-bank.de/v1/consents \
  --key private.key --cert certificate.pem --cert-type PEM \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'psu-ip-address: 192.168.8.78' \
  -H 'tpp-redirect-uri: https://tpp-frontend-redirect.de' \
  -H 'x-request-id: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721' \
  -d '{
    "access": {
        "accounts": [],
        "balances": [],
        "transactions": []
    },
    "recurringIndicator": true,
    "validUntil": "2019-05-31",
    "frequencyPerDay": 4
}'
```

Response:

```
{
    "consentStatus": "received",
    "consentId": "IrBy0Htn3DlznS23-BFxmsQqFYsbtcOzCX5o62L2u5cRDRHD0-
nvjJJoV57baNW3dlHD3QA55-kgQcc2zVfnrPSdMWF3876hAweK_n7HJlg=_=_psGLvQpt9Q",
    "_links": {
        "scaRedirect": {
            "href": "https://redirect-sndbx.dab-bank.de/v1/online-
banking/init/ais/250b526f-3754-4b63-a8f4-417ea8c8cbd9"
        },
        "self": {
            "href": "https://xs2a-sndbx.dab-bank.de/v1/consents/IrBy0Htn3DlznS23-
BFxmsQqFYsbtcOzCX5o62L2u5cRDRHD0-nvjJJoV57baNW3dlHD3QA55-
kgQcc2zVfnrPSdMWF3876hAweK_n7HJlg=_=_psGLvQpt9Q"
        },
        "status": {
            "href": "https://xs2a-sndbx.dab-bank.de/v1/consents/IrBy0Htn3DlznS23-
BFxmsQqFYsbtcOzCX5o62L2u5cRDRHD0-nvjJJoV57baNW3dlHD3QA55-
kgQcc2zVfnrPSdMWF3876hAweK_n7HJlg=_=_psGLvQpt9Q/status"
        },
        "scaStatus": {
            "href": "https://xs2a-sndbx.dab-bank.de/v1/consents/IrBy0Htn3DlznS23-
BFxmsQqFYsbtcOzCX5o62L2u5cRDRHD0-nvjJJoV57baNW3dlHD3QA55-
kgQcc2zVfnrPSdMWF3876hAweK_n7HJlg=_=_psGLvQpt9Q/authorisations/250b526f-3754-4b63-a8f4-
417ea8c8cbd9"
          }
     }
}
```

Status Values:

Used status values for Account Information consent:

- received
- rejected
- valid
- revokedByPsu
- terminatedByTpp
- expired

More details (e.g. http error codes) you will find in the .yaml file also provided for the download on our developer info sites:

https://b2b.dab-bank.de/xs2a/

7 PAYMENT INSTRUMENT ISSUING SERVICE (PIIS)

7.1 BASIC INFORMATION

This chapter defines a list of services provided for PIISP.

Prerequisites:

- The TPP is registered for the PIISP role and valid PIIS scope
- The TPP has been successfully authenticated
- All test-accounts can be used.

Format:

- JSON

7.2 ENDPOINT DEFINITION

Method	Endpoint
GET	/v1/funds-confirmations

7.3 EXAMPLES

Request:

```
curl -v -X POST "https://xs2a-sndbx.dab-bank.de/v1/funds-confirmations" \
  --key private.key --cert certificate.pem --cert-type PEM \
  -H "Content-Type: application/json" \
  -H "accept: application/json" \
  -H "X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721" \
  -d '{
  "cardNumber": "CARD-NUMBER",
  "payee": "Testuser",
  "account": {
    "currency": "EUR",
    "iban": "DE74701204003326455007"
  },
  "instructedAmount": {
    "currency": "EUR",
    "amount": "15"
  }
}'
```

Response:

```
{
  "fundsAvailable": true
}
```

Status Values:

Response values: true, false

More details (e.g., http error codes) you will find in the .yaml file also provided for the download on our developer info sites:

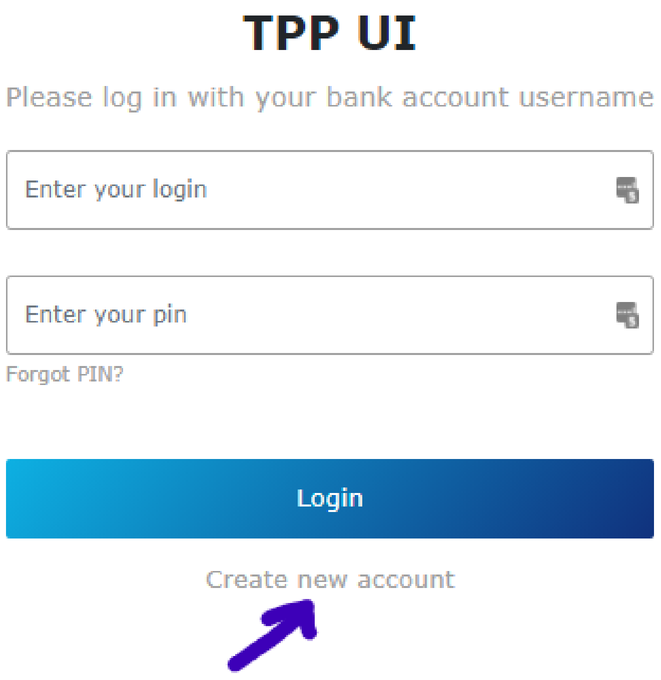
https://b2b.dab-bank.de/xs2a/

8 TEST DATA

Each TPP can register an account in the TPP-UI module, from where they can configure the test-data that they require for their test cases.

8.1 REGISTERING A NEW TPP ACCOUNT

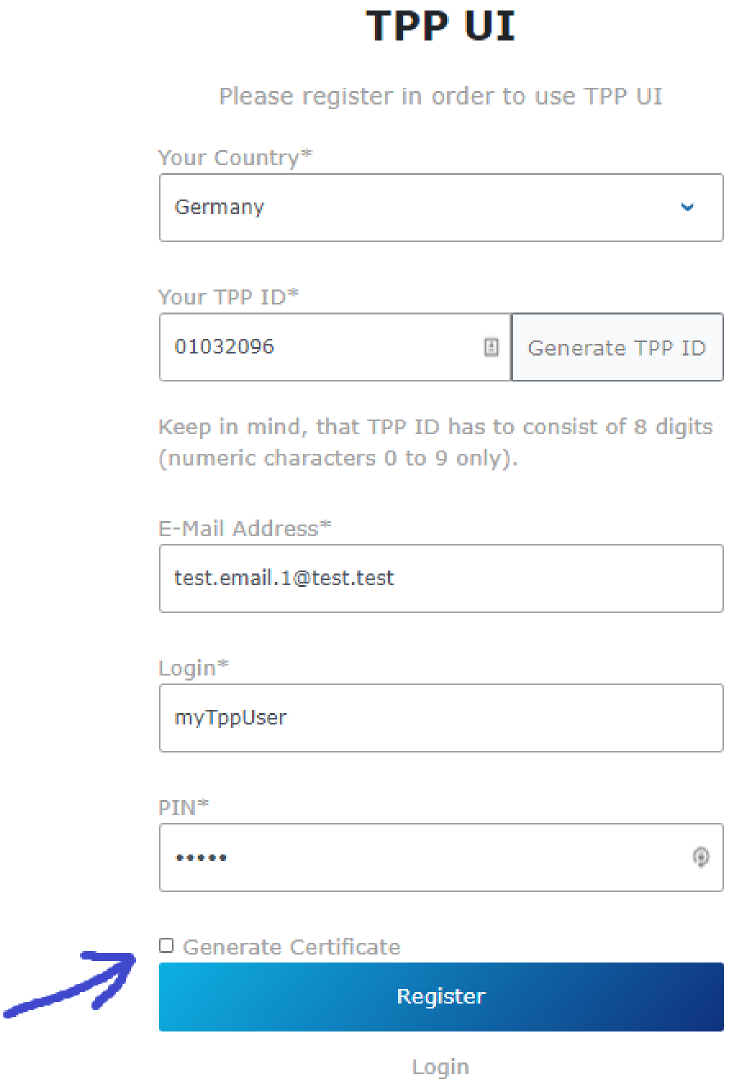
To register an account, you have to navigate to the "https://tpp-ui-xs2a-sndbx.dab-bank.de" URL, where you can click on the "Register"-button:



TPP UI

Please log in with your bank account username

After entering the details for the TPP, like seen in the next screenshot, you can also optionally select the "Generate Certificate"-checkbox, if you want to generate a new test-certificate:



TPP UI

Please register in order to use TPP UI

Your Country*: Germany

Your TPP ID*: 01032096

Keep in mind, that TPP ID has to consist of 8 digits (numeric characters 0 to 9 only).

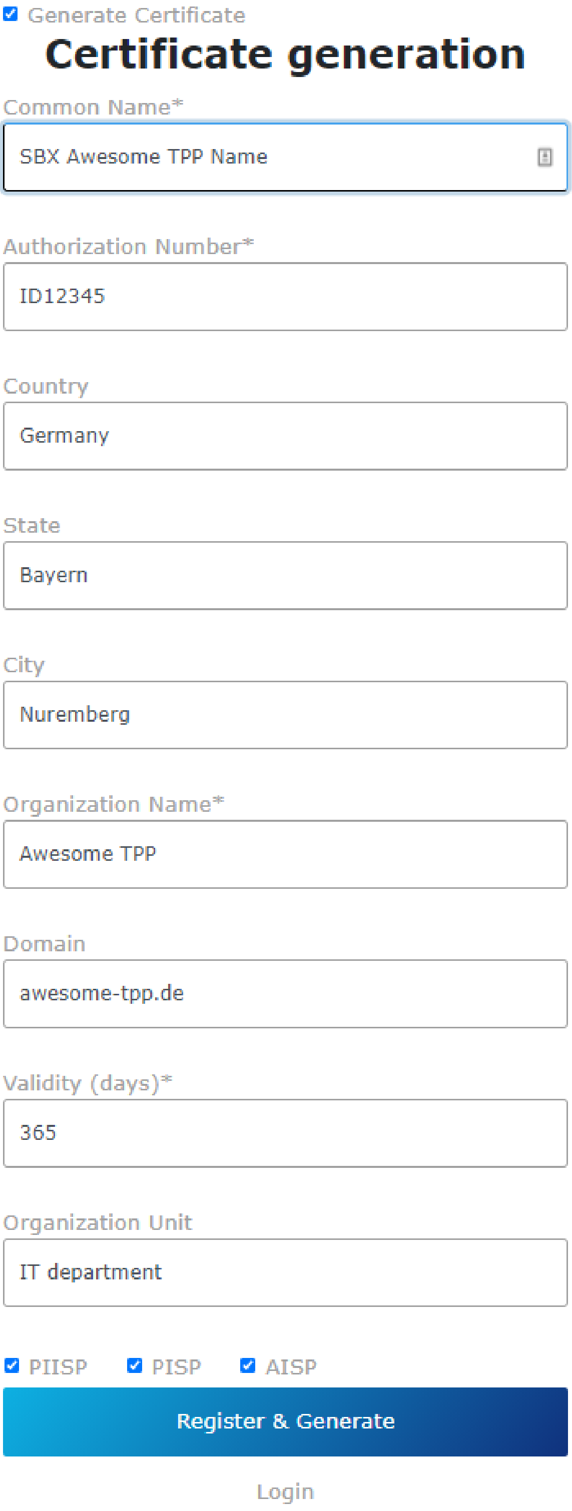
E-Mail Address*: test.email.1@test.test

Login*: myTppUser

PIN*: •••••

□ Generate Certificate

If you selected the checkbox, the registration form will be extended with the fields required to generate a new certificate, and after filling in all the data the "Register & Generate"-button can be clicked to finish the registration process an proceed to the certificate-download if the option was selected:



☑ Generate Certificate

Certificate generation

Common Name*: SBX Awesome TPP Name

Authorization Number*: ID12345

Country: Germany

State: Bayern

City: Nuremberg

Organization Name*: Awesome TPP

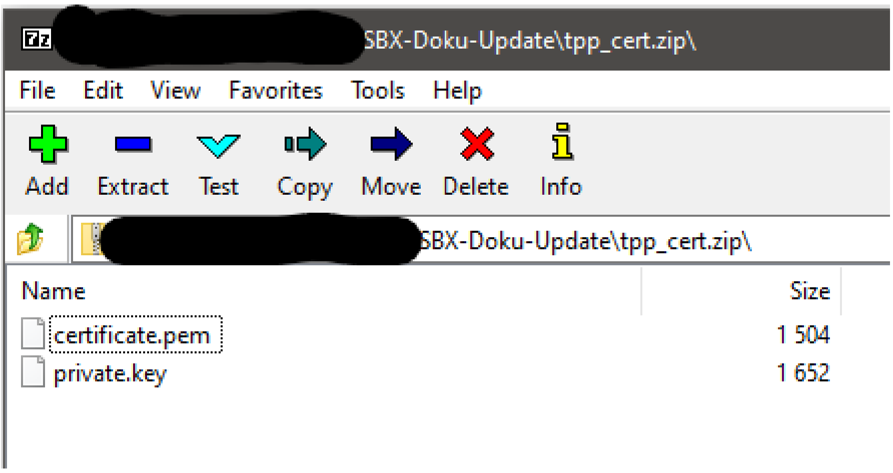
Domain: awesome-tpp.de

Validity (days)*: 365

Organization Unit: IT department

☑ PIISP ☑ PISP ☑ AISP

If you selected the option to generate a certificate you will get to download a ZIP file that contains the “certificate.pem” file and the “private.key” file as seen in the screenshot (please note that both the certificate and private key are generated as a single line, so further formatting might be required, depending on the usage):

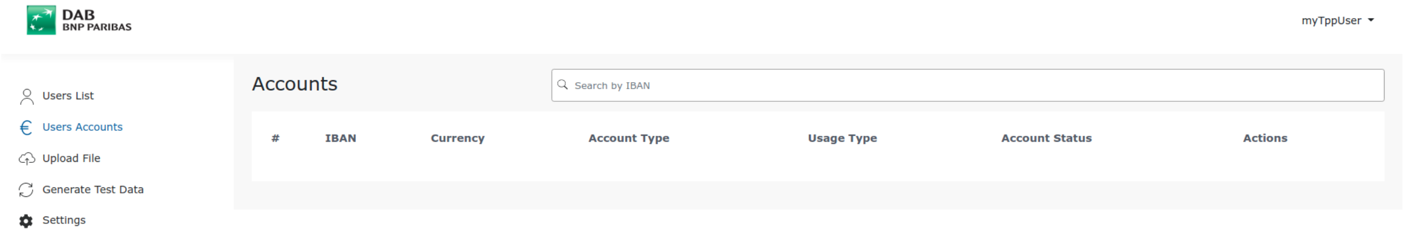


[redacted]SBX-Doku-Update\tpp_cert.zip\

[redacted]SBX-Doku-Update\tpp_cert.zip\

Name	Size
certificate.pem	1 504
private.key	1 652

After logging into the TPP-UI, you will see the following dashboard, from where you can create and manage the test data for your test-cases:

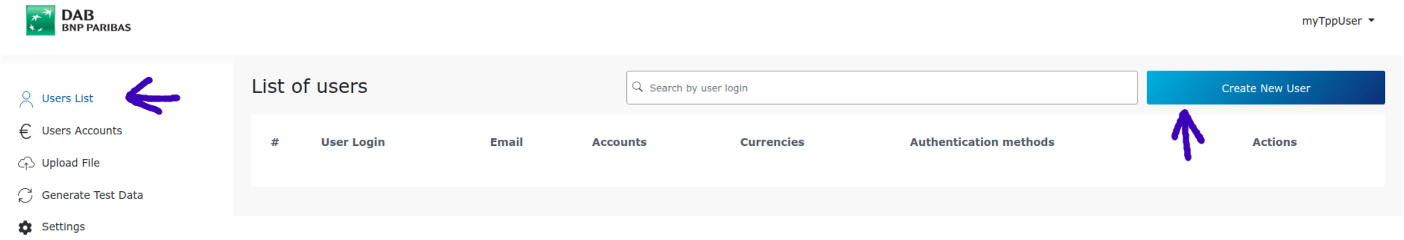


Accounts

#	IBAN	Currency	Account Type	Usage Type	Account Status	Actions

8.2 CREATING A NEW TEST USER (PSU)

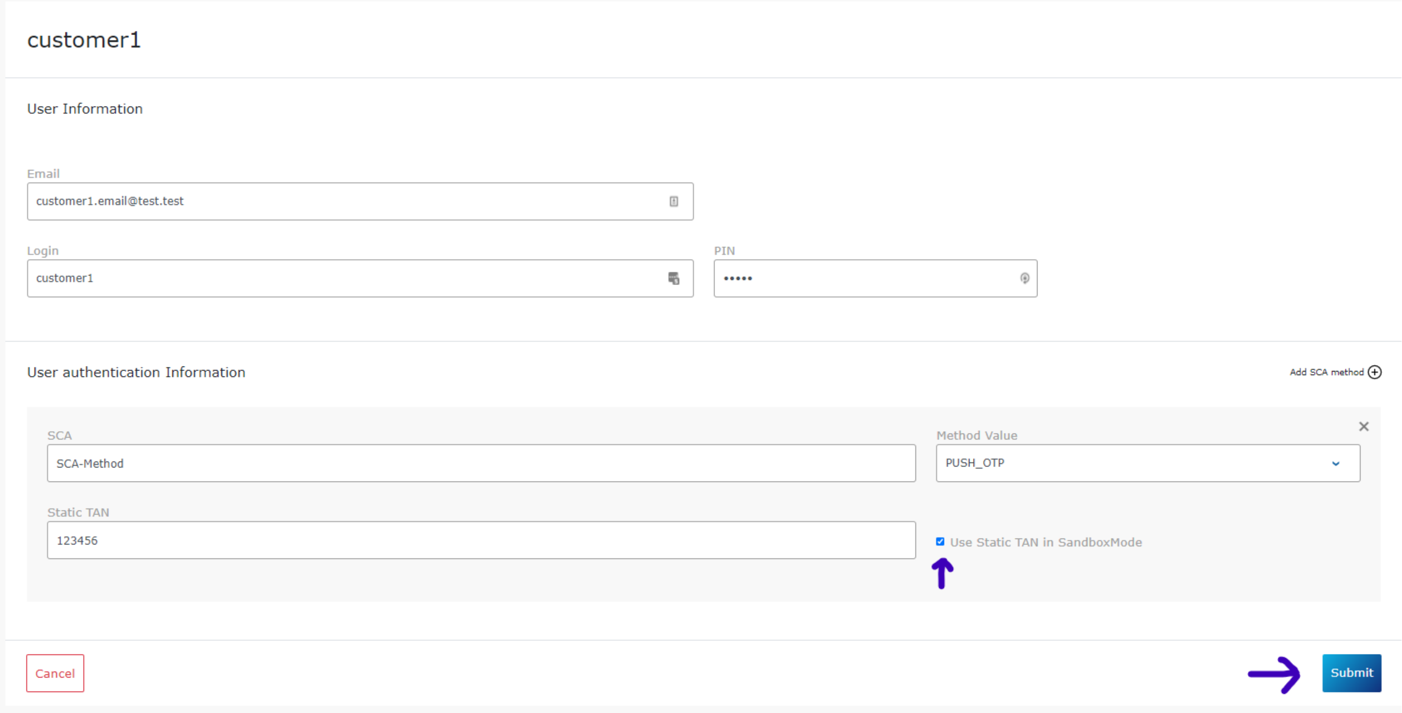
To create a new PSU account, you need to navigate to the "Users List" and click on the "Crete New User"-button in the top-right of the page:



List of users

#	User Login	Email	Accounts	Currencies	Authentication methods	Actions

Afterwards you will be redirected to the User Information page, where the email, login, pin and SCA for the new user have to be filled in. The sandbox currently has no email functionality implemented, so no emails can be sent from within the sandbox. The login and PIN are used to login into the Banking-UI module with the user to authorize PIS or AIS requests. Please also note, that the sandbox is currently only supporting the "Use Static TAN in SandboxMode" option for SCA, regardless of the chosen "Method Value". After filling in all the data and pressing the "Submit"-button, the user will have been created:



customer1

User Information

Email: customer1.email@test.test

Login: customer1

PIN: •••••

User authentication Information

SCA: SCA-Method

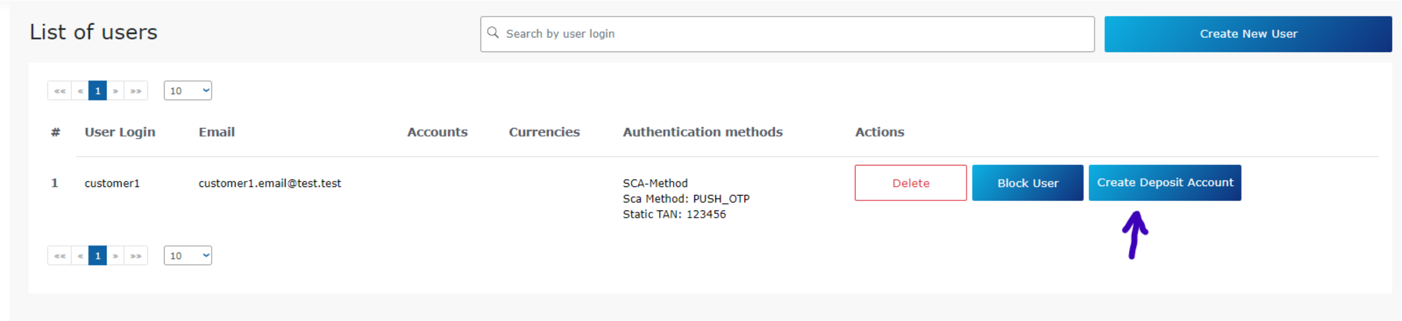
Method Value: PUSH_OTP

Static TAN: 123456

☑ Use Static TAN in SandboxMode

8.3 CREATING A NEW ACCOUNT FOR A PSU

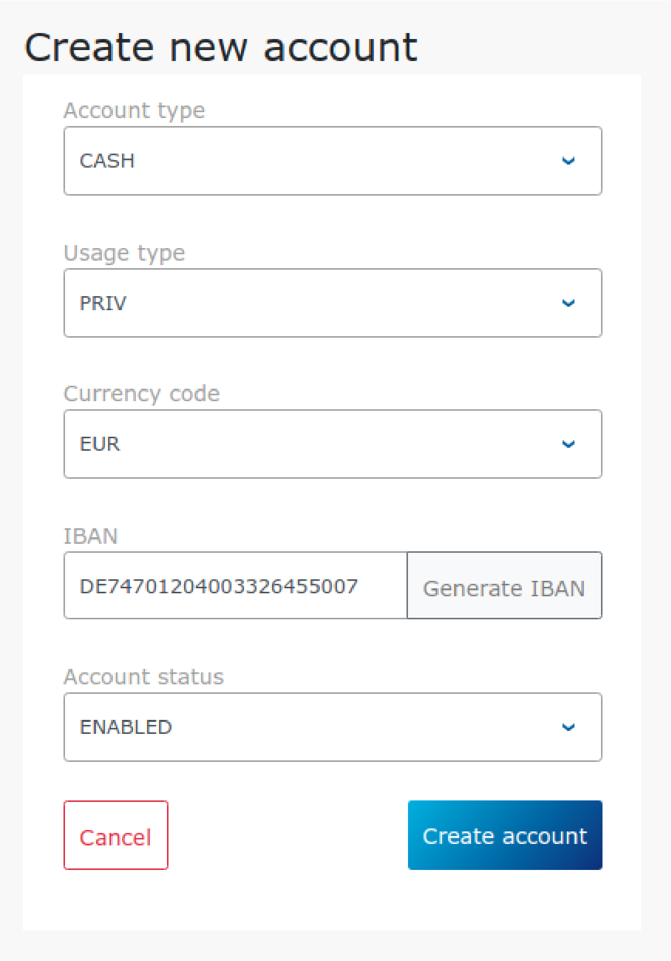
New accounts can be created for each user, from the "Users List" page, by clicking the "Create Deposit Account"-button:



List of users

#	User Login	Email	Accounts	Currencies	Authentication methods	Actions
1	customer1	customer1.email@test.test			SCA-Method Sca Method: PUSH_OTP Static TAN: 123456	Delete Block User Create Deposit Account

To create the account the following form has to be filled in, with the mention that the "Generate IBAN" functionality will not generate an IBAN that can be used in a PIS request as the debtor. To create an account that can be used as a debtor for a PIS request, the IBAN has to contain the following bank-code: 701 204 00 (in chapter 8.6 is a list of example IBANs that can be used):



Create new account

Account type: CASH

Usage type: PRIV

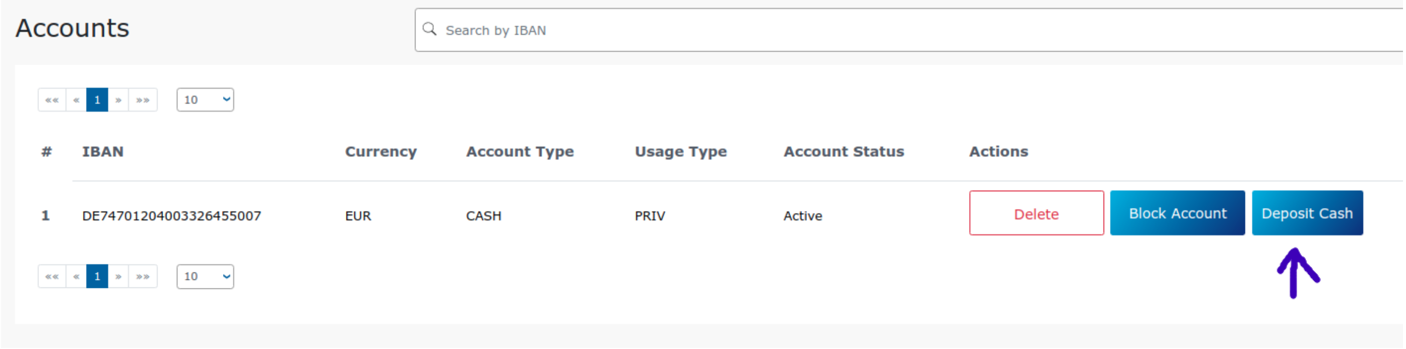
Currency code: EUR

IBAN: DE74701204003326455007

Account status: ENABLED

8.4 ADDING CASH TO AN EXISTING ACCOUNT

Adding cash to an account can be done by going to the “Users Accounts” page from the navigation menu on the left, and clicking on the “Deposit Cash”-button:

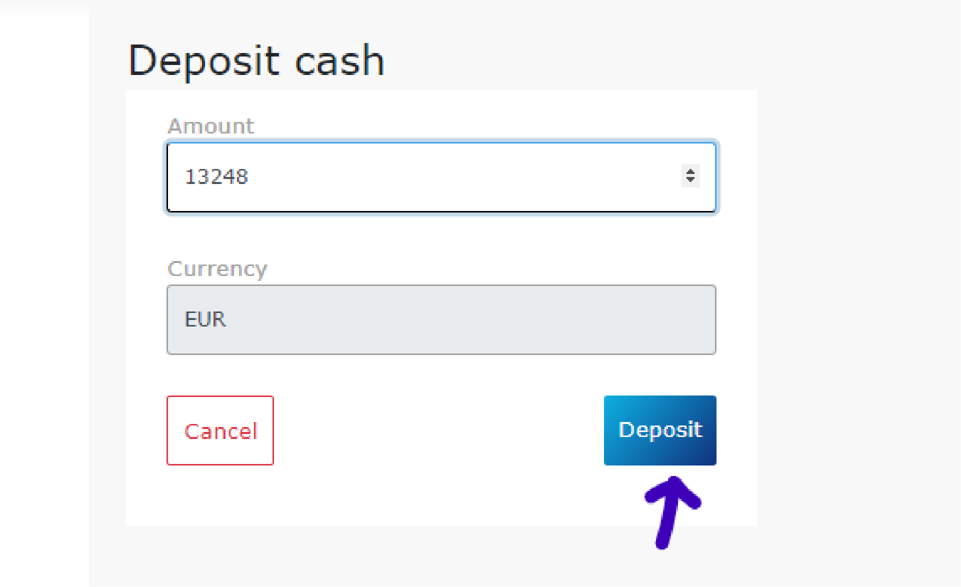


Accounts

#	IBAN	Currency	Account Type	Usage Type	Account Status	Actions
1	DE74701204003326455007	EUR	CASH	PRIV	Active	Delete Block Account Deposit Cash

Afterwards you have to fill in the desired amount and click on the “Deposit”-button as seen in the following screenshot.

This action will then be visible in the transaction history of that account, with the “Cash deposit through Bank ATM” description:



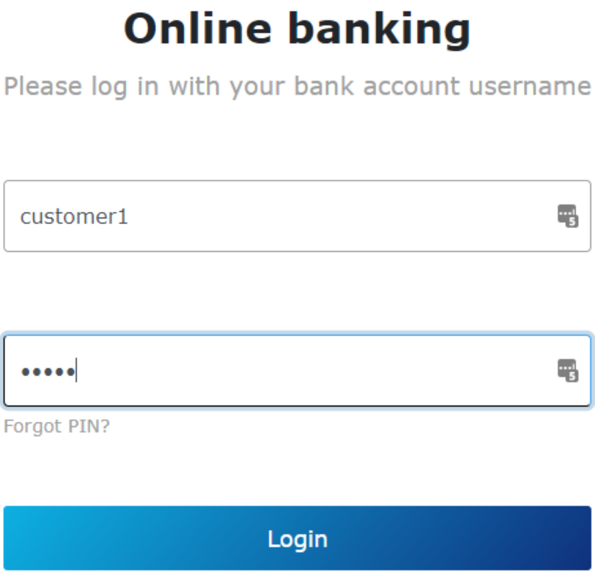
Deposit cash

Amount: 13248

Currency: EUR

8.5 CHECKING TRANSACTION HISTORY FOR AN ACCOUNT

To check the transaction history (from the perspective of the PSU), you can go to the Banking-UI website (https://banking-ui-xs2a-sndbx.dab-bank.de), and login using the credential of a user:



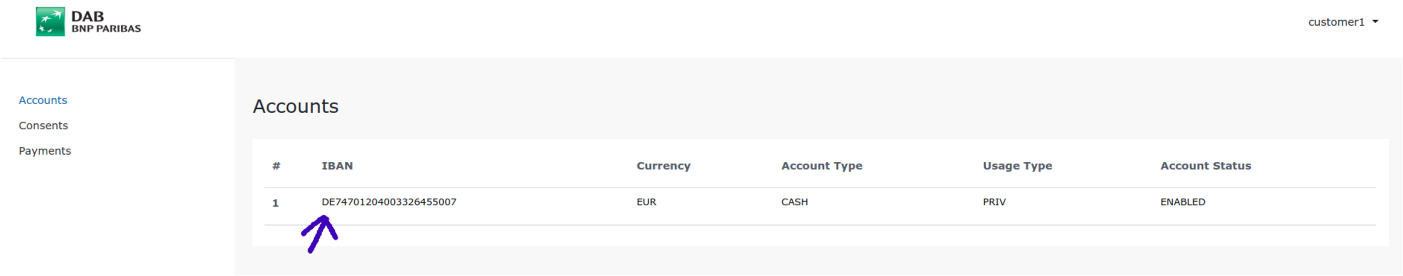
Online banking

Please log in with your bank account username

customer1

•••••

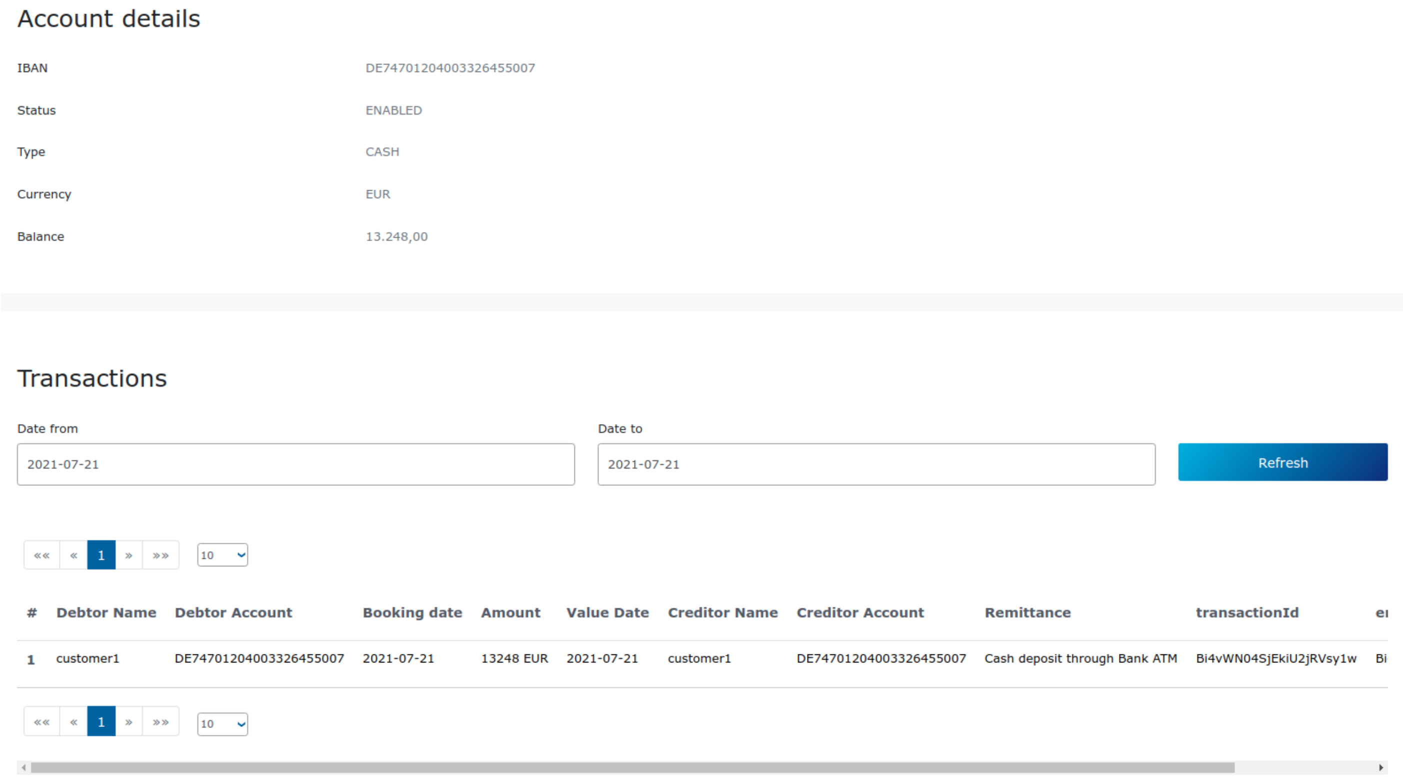
After logging into the account, the transaction history can be reached by clicking on the desired account:



Accounts

#	IBAN	Currency	Account Type	Usage Type	Account Status
1	DE74701204003326455007	EUR	CASH	PRIV	ENABLED

After clicking on the account, the “Account details” will be displayed, and underneath will be the “Transactions” section, where the transactions are shown (please note that there is a date-filter on the transaction list):



Account details

IBAN	DE74701204003326455007
Status	ENABLED
Type	CASH
Currency	EUR
Balance	13.248,00

Transactions

Date from: 2021-07-21

Date to: 2021-07-21

#	Debtor Name	Debtor Account	Booking date	Amount	Value Date	Creditor Name	Creditor Account	Remittance	transactionId	er
1	customer1	DE74701204003326455007	2021-07-21	13248 EUR	2021-07-21	customer1	DE74701204003326455007	Cash deposit through Bank ATM	Bi4vWN04SjEkiU2jRVsy1w	Bi

8.6 IBAN EXAMPLES FOR DEBTOR ACCOUNTS

As mentioned previously, there is a validation for the PIS requests, which checks that the IBAN for the selected debtor-account is valid and that it contains one of the two bank-codes: 760 300 80 / 701 204 00. Below is a list of example IBANs that match this rule:

1. DE74701204003326455007
2. DE94701204003326456005
3. DE17701204003326457003
4. DE37701204003326458001
5. DE78701204003326459009
6. DE98701204003326460007
7. DE21701204003326461005
8. DE41701204003326462003
9. DE61701204003326463001

10. DE05701204003326464009
11. DE58701204003326464025
12. DE36701204003326464033