

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 3
- 2. GENERAL INFORMATION / USED ABBREVIATIONS 3
- 3. TPP AUTHENTICATION / CERTIFICATES 5
 - 3.1 Basic Information..... 5
 - 3.2 Create a Test Certificate 5
 - 3.3 Use a Test Certificate from a TSP 6
- 4. SANDBOX 7
 - 4.1 General Setup..... 7
 - 4.2 SCA Usage..... 7
 - 4.3 IBAN selection..... 7
- 5. PAYMENT INITIATION SERVICE (PIS) 8
 - 5.1 Basic Information..... 8
 - 5.2 Endpoint Definition..... 10
 - 5.3 Examples..... 10
- 6. ACCOUNT INFORMATION SERVICE (AIS) 15
 - 6.1 Basic Information..... 15
 - 6.2 Endpoint Definition..... 15
 - 6.3 Examples..... 16
- 7. PAYMENT INSTRUMENT ISSUING SERVICE (PIIS) 20
 - 7.1 Basic Information..... 20
 - 7.2 Endpoint Definition..... 20
 - 7.3 Examples..... 20
- 8. TEST DATA 21
 - 8.1 Test Data..... 21

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

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. For the sandbox, it is not necessary, as the IBANs are static as described in this documentation.

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. 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 static test data as described in detail in chapter TEST DATA.

Accounts and transactions of the sandbox are static and will never change. As in real live sandbox accounts belong to a specific PSU. Initiated consents for payment initiations and account information are dynamic to allow for a realistic work flow.

To be able to test the live cycle of PSD2 use cases, the SCA can be simulated by a request to the SCA redirect URL. The behaviour of the SCA-request is determined by the psu-id and IBAN used, thus making a complete automated test cycle possible.

4.2 SCA USAGE

For test purposes for some psu-ids the SCA redirect request will be confirmed, for others it will be rejected etc. The PSU can be selected by supplying a request parameter that allows to select the desired behaviour. E. g. :

```
GET https://redirect-sndbx.dab-bank.de/v1/online-banking/init/pis/4cd9792e-64d4-42c6-b618-da7bfcfe8135?psu-id=PSU-Rejected
```

```
GET https://redirect-sndbx.dab-bank.de/v1/online-banking/init/pis/7f7dc61e-8a66-445a-aa97-1ff0725e57a9?psu-id=PSU-Successful
```

The response of this call will contain a simple HTML page with information on the changed status and a redirect to the redirect URL initially supplied by the TPP. It is important to use the SCA redirect request, but the HTML-response can be ignored if a TPP prefers to do so.

4.3 IBAN SELECTION

By selecting IBAN for a specific use case, the TPP is able to test different scenarios like successful payments or rejected payments due to an insufficient balance. It is important to use an IBAN that belongs to the psu-id used for the SCA redirect request; otherwise, the case will be rejected.

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 Recurring Payment or not yet executed Future Dated Payment needs to be performed by cancelling the existing one and creating a new SEPA Recurring Payment.
- Payment Cancellation together with PSU until T-1 is only supported for successful authorised Payment Initiations in these cases:
 - o Recurring payment not yet cancelled
 - o Future dated payment not yet cancelled or executed
- Payment Cancellations by TPP for Payment Initiations before authorization (therefore in status RCVD) is 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.
- A difference between the Sandbox and the production XS2A interface: after using the redirect URL to log in, if a user tries to abort the authorization, the status of the transaction will not be automatically updated to CANC/ RJCT, but instead will remain in status ACCP

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": "bopPbubjqCvz9qi6JD1k00q-Xn_fKPUjzd0QzW2IstQ-6HMwyl6lSS1sGvk3iWNp8N4xvbjPxSBzBKq5VgPnPPSdMWF3876hAweK_n7HJlg=_psGLvQpt9Q",
  "transactionFees": null,
  "transactionFeeIndicator": null,
  "scaMethods": null,
  "chosenScaMethod": null,
  "challengeData": null,
  "_links": {
    "scaRedirect": {
      "href": "https://redirect-sndbx.dab-bank.de/v1/online-banking/init/pis/caa190c8-ca4c-4fdb-a04c-0934e73e05b7"
    },
    "self": {
      "href": "https://xs2a-sndbx.dab-bank.de/v1/payments/sepa-credit-transfers/bopPbubjqCvz9qi6JD1k00q-Xn_fKPUjzd0QzW2IstQ-6HMwyl6lSS1sGvk3iWNp8N4xvbjPxSBzBKq5VgPnPPSdMWF3876hAweK_n7HJlg=_psGLvQpt9Q"
    },
    "status": {
      "href": "https://xs2a-sndbx.dab-bank.de/v1/payments/sepa-credit-transfers/bopPbubjqCvz9qi6JD1k00q-Xn_fKPUjzd0QzW2IstQ-6HMwyl6lSS1sGvk3iWNp8N4xvbjPxSBzBKq5VgPnPPSdMWF3876hAweK_n7HJlg=_psGLvQpt9Q/status"
    },
    "scaStatus": {
      "href": "https://xs2a-sndbx.dab-bank.de/v1/payments/sepa-credit-transfers/bopPbubjqCvz9qi6JD1k00q-Xn_fKPUjzd0QzW2IstQ-6HMwyl6lSS1sGvk3iWNp8N4xvbjPxSBzBKq5VgPnPPSdMWF3876hAweK_n7HJlg=_psGLvQpt9Q/authorisations/caa190c8-ca4c-4fdb-a04c-0934e73e05b7"
    }
  },
  "psuMessage": null,
  "tppMessages": null
}
```

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 recurring or 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/>

Useful Psu-ids and IBANs for Payment Initiation:

PSU-ID	IBAN DAB BNP Paribas	SCA-Status	Transaction Status
PSU-Successful	DE74701204003326455007*	finalised	ACTC/ACSC**
PSU-Rejected***	DE21701204003326461005	failed	RJCT
PSU-Blocked	DE61701204003326463001	<i>(no SCA Status available)</i>	<i>(no Transaction Status available)</i>
PSU-InternalLimit	DE05701204003326464009	finalised	RJCT

Useful Psu-ids and IBANs for Payment Cancellation

PSU-ID	IBAN DAB BNP Paribas	SCA-Status	Transaction Status
PSU-Successful	DE74701204003326455007*	finalised	CANC****
PSU-Cancellation-Rejected	DE41701204003326462003	failed	ACTC <i>(no change)</i>

Useful Psu-ids and IBANs for Get Payment

PSU-ID	IBAN DAB BNP Paribas	SCA-Status	Transaction Status
PSU-Successful	DE74701204003326455007*	finalised	ACTC/ACSC

* PSU-Successful has further accounts to test the behaviour, please see chapter 8 Test Data

** 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".

*** Same behaviour is implemented, when PSU does not match IBAN or the PSU-ID is not known.

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

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",  
  "scaMethods": null,  
  "chosenScaMethod": null,  
  "challengeData": null,  
  "_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-
BFxmsQqFYsbtCzCX5o62L2u5cRDRHD0-nvjJJov57baNW3dlHD3QA55-
kgQcc2zVfnrPSdMWF3876hAweK_n7HJlg=_psGLvQpt9Q/authorisations/250b526f-3754-4b63-a8f4-
417ea8c8cbd9"
    }
  },
  "psuMessage": null
}
```

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/>

Useful Psu-ids and IBANs for Consent Creation

PSU-ID	IBAN DAB BNP Paribas	SCA-Status	Consent Status
PSU-Successful	DE74701204003326455007*	finalised	valid
PSU-Rejected	DE21701204003326461005	failed	rejected
PSU-Blocked	DE61701204003326463001	<i>(no SCA Status available)</i>	<i>(no Consent Status available)</i>
PSU-ConsentExpired	DE58701204003326464025	finalised	expired
PSU-ConsentRevokedByPsu	DE36701204003326464033	finalised	revokedByPsu

Useful Psu-ids and IBANs for Consent Deletion

PSU-ID	IBAN DAB BNP Paribas	Consent Status
PSU-Successful	DE74701204003326455007*	terminatedByTpp
PSU-InternalLimit	DE05701204003326464009	terminatedByTpp

Useful Psu-ids and IBANs for Get Account Data

PSU-ID	IBAN DAB BNP Paribas	Consent Status
PSU-Successful	DE74701204003326455007*	valid
PSU-ConsentExpired	DE58701204003326464025	expired
PSU-ConsentRevokedByPsu	DE36701204003326464033	revokedByPsu

* PSU-Successful has further accounts to test the behaviour, please see chapter 8 Test Data.

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
- The PSU has given a consent to the ASPSP

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

8.1 TEST DATA

PSU-ID	IBAN DAB BNP Paribas	Comments
PSU-Successful	DE74701204003326455007	GIRO
	DE94701204003326456005	SAVINGS
	DE17701204003326457003	EMPTY GIRO
	DE37701204003326458001	NEGATIVE BALANCE GIRO
	DE78701204003326459009	LOWER BALANCE GIRO
	DE98701204003326460007	USD GIRO
PSU-Rejected	DE21701204003326461005	
PSU-Cancellation-Rejected	DE41701204003326462003	
PSU-Blocked	DE61701204003326463001	
PSU-InternalLimit	DE05701204003326464009	
PSU-ConsentExpired	DE58701204003326464025	
PSU-ConsentRevokedByPsu	DE36701204003326464033	