PSD2 API Solution - Documentation of the DKB Specific Methods and Workflows

Addressees:

Business Analysts, Project Managers, Developers, Architects, IT

Autor	Florian Steier
Version	1.5
Datum	24.08.2023

Change Log

Version	Change	Date	Author
1.0	Release: Version 1.0	23.04.2019	A. Kuhn
1.1	Release: Version 1.3.6	30.06.2020	F. Steier
1.2	Update Chapter 3.1	24.02.2021	F. Steier
1.2.1	Quality Assurance	29.11.2021	F. Steier
1.2.2	Update Chapter 6.2	26.01.2021	F. Steier
1.3	Insert Chapter 2.2.1	01.07.2022	F. Steier
1.4	Insert Chapter 6 Contingency Measures	24.01.2023	F. Steier
1.5	Update due to changes for SCA Excemption	24.08.2023	F. Steier

Inhaltsverzeichnis

1. **Introduction** 1
2. **Strong Customer Authentication (SCA)** 1
2.1 PreAuth 1
2.2 SCA TAN 1
2.2.1 Change chipTAN method 2
2.3 SCA Exemptions 5
3. **Account Information Service (AIS)** 6
3.1 Create Consent 6
3.2 Get Account Information 6
3.3 Account Retrievals 4 Times / Day 6
4. **Payment Initiation Service (PIS)** 7
4.1 Payment SCA 7
4.2 Payment Products 7
4.3 Payment Services 8
5. **Fund Confirmation Service (FCS)** 9
5.1 FCS Consent 9
5.2 Fund Confirmation 9
6. **Contingency Measures** 9
6.1 Fallback 9
6.1.1 Successful Response 10
6.1.2 Error Response 10
7. **Miscellaneous** 10
7.1 Service Information 10
7.2 Updates 10

1. Introduction

For the implementation of the **PSD2 XS2A** (Access to Account) interface of the **DKB (**Deutsche Kreditbank) the procedure according to the Berlin Group Standard V1.3.6 was chosen. The PSD2 XS2A API (**PSD2 API Solution**) is provided by **CREALOGIX AG**. The corresponding MVP (Minimum Viable Product) approach is described in detail in the TPP documentation (PSD2 API Solution - Documentation for TPPs) including all standard business transactions and technical information. However, some features of the Berlin Group standard have not been implemented or have been implemented differently. Reasons are the technical design of the interface itself, for which DKB uses the **FinTS** access variant (similar to the **DKB Online Banking**) and DKB-internal decisions. An explanation of the special features can be found in the following chapters. Additionally, this document lists the supported DKB products and methods, which are available for **AISP** (Account Information Service Provider), **PISP** (Payment Initiation Service Provider) and **PIIS** (Payment Instrument Issuing Service Provider) via the XS2A interface.

All PSD2 relevant features and functions will be available at the latest to **14.June 2019**. Thus, the DKB can apply for the exemption from the contingency measures according to **RTS Art. 33(6)**. However, a fallback concept can be found in the document ‘Contingency Measures’.

2. Strong Customer Authentication (SCA)

A SCA is required for the PreAuth of the PSU and for the authorization of AIS consents and PIS payments.

2.1 PreAuth

A TPP-controlled pre-step authentication is used to enable access the system. In the preAuth a SCA is **always** required for the first access and every consecutive 180 days. The preAuth verifies the first factor (of the SCA), i.e. the PSU has to authenticate (login + password) the access to its DKB online banking account before the XS2A interface can be used.

For Payment Initiation Services (PIS), Account Information Services (AIS) and Fund Confirmation Services (FCS) the preAuth has to be the first step. After the successful authentication of a PSU in the preAuth, an access token is generated which can be used at the XS2A interface afterwards.

The required, separate API calls for the PreAuth are described in the TPP documentation (and PreAuth documentation). The OAuth method(s) for the SCA are:

- **Embedded Approach**: The PSU has to enter his login data at the TPP (Third Party Provider). The TPP is responsible to send the credentials (including the TAN) via API to the ASPSP.

2.2 SCA TAN

During the SCA, the PSU has to authorize an action with its 2nd factor. (The 1st factor are the PSUs DKB credentials.) Therefore, the DKB offers following TAN methods:

2nd Factor	PSU Requirements	Description
TAN2Go	Mobile Device , DKB TAN2Go App	The PSU gets a one-time password (OTP/TAN) pushed into the DKB TAN2Go App. The PSU needs to provide the TAN in the TPP page and the TPP send it via the API to the ASPSP (embedded).
chipTAN (manual)	chipTAN Generator DKB-Banking-Card / Girocard	The PSU receives a one-time password (OTP/TAN) on the ‘TAN-Generator’. The PSU needs to provide the TAN in the TPP page and the TPP sends it via the API to the ASPSP (embedded).

The DKB customer (PSU) can choose one or several of these TAN methods at DKB. The available 2nd factor methods for a PSU are reported to the TPP via the XS2A API. The TAN must be transmitted to the API within 5 minutes.

2.2.1 Change chipTAN method

The XS2A API provides a mechanism to change the chipTAN method for authorising consents, transactions older 90 days and payments. The authorisation flow will follow the same steps already known from push-TAN/TAN2go authorisations.

2.2.1.1 Flow for creating a consent

1. create consent - /v1/consents
2. start authorisation - /v1/consents/{{consentId}}/authorisations
3. select method - /v1/consents/{{consentId}}/authorisations/{{authorisationId}}
4. send TAN - /v1/consents/{{consentId}}/authorisations/{{authorisationId}}

The step **create authorisation** will provide the list of available chipTAN methods (manual, flicker and QR). The current implementation skips the following step (**select method)** because only the method manual is supported. In the pushTAN flow, this step is already necessary to select the TAN device of the PSU. The next release provides the same flow for chipTAN (analog to the current pushTAN flow).

2.2.1.2 The response of create authorisation - /v1/consents/{{consentId}}/authorisations:

```
{
    "scaStatus": "started",
    "authorisationId": "{{authorisationId}}",
    "scaMethods": [
        {
            "authenticationType": "CHIP_OTP",
            "authenticationVersion": "1",
            "authenticationMethodId": "eyJhbGciOiJIUzI1NiJ9.eyJhdXRoZW50aWNhdGlvblR5cGUiOiJDSEl-
QIiwiYXV0aGVudGljYXRpb25WZXJzaW9uIjoiMSIsImF1dGhlbnRpY2F0aW9uTWV0aG9kSWQiOiJNQU5VQUwiLCJuYW1lI-
joiTUFOVUFMIiwiZXhwbGFuYXRpb24iOm51bGwsImlkIjoiTUFOVUFMIiwic3ViamVjdCI6IlRBTk1lZGlhQWxpYXMiLCJpYXQi-
OjE2NTcwMTIxNzk2MTl9.RtQf-TBNdN5tB_Q2JfgcaXL8V3uktuYLi8lXbcn3eAk",
            "name": "MANUAL"
        },
        {
            "authenticationType": "CHIP_OTP",
            "authenticationVersion": "1",
            "authenticationMethodId": "eyJhbGciOiJIUzI1NiJ9.eyJhdXRoZW50aWNhdGlvblR5cGUiOiJDSEl-
QIiwiYXV0aGVudGljYXRpb25WZXJzaW9uIjoiMSIsImF1dGhlbnRpY2F0aW9uTWV0aG9kSWQiOiJRUiIsIm5hbWUiOiJRUiI-
sImV4cGxhbmF0aW9uIjpudWxsLCJpZCI6IlFSIiwic3ViamVjdCI6IlRBTk1lZGlhQWxpYXMiLCJpYXQi-
OjE2NTcwMTIxNzk2MjF9.UeKX0rqD7Czzm7zEQsuZv5g-WDEfl2zEkR5Yw56Tgfc",
            "name": "QR"
        },
        {
            "authenticationType": "CHIP_OTP",
            "authenticationVersion": "1",
            "authenticationMethodId": "eyJhbGciOiJIUzI1NiJ9.eyJhdXRoZW50aWNhdGlvblR5cGUiOiJDSEl-
QIiwiYXV0aGVudGljYXRpb25WZXJzaW9uIjoiMSIsImF1dGhlbnRpY2F0aW9uTWV0aG9kSWQiOiJGTElDS0VSIiwi-
bmFtZSI6IkZMSUNLRVIiLCJleHBsYW5hdGlvbiI6bnVsbCwiaWQiOiJGTElDS0VSIiwic3ViamVjdCI6Il-
RBTk1lZGlhQWxpYXMiLCJpYXQiOjE2NTcwMTIxNzk2MjF9._eDG0W_MDd3U6-Zqp4AUnZtIaJxwmG6cBtefXV05XA0",
            "name": "FLICKER"
        }
    ],
    "_links": {
        "scaStatus": {
            "href": "/v1/consents/{{consentId}}/authorisations/{{authorisationId}}"
        }
    }
}
```

2.2.1.3 Example of select method request - QR:

```
{
  "authenticationMethodId": "eyJhbGciOiJIUzI1NiJ9.eyJhdXRoZW50aWNhdGlvblR5cGUiOiJDSElQIiwiYXV0aGV-
udGljYXRpb25WZXJzaW9uIjoiMSIsImF1dGhlbnRpY2F0aW9uTWV0aG9kSWQiOiJRUiIsIm5hbWUiOiJRUiI-
sImV4cGxhbmF0aW9uIjpudWxsLCJpZCI6IlFSIiwic3ViamVjdCI6IlRBTk1lZGlhQWxpYXMiLCJpYXQi-
OjE2NTcwMTIxNzk2MjF9.UeKX0rqD7Czzm7zEQsuZv5g-WDEfl2zEkR5Yw56Tgfc"
}
```

2.2.1.4 Example of select method response - QR:

```
{
    "chosenScaMethod": {
        "authenticationType": "CHIP_OTP",
        "authenticationVersion": "1",
        "name": "QR"
    },
    "challengeData": {
        "image": "{{base64 encoded binary data}}",
        "data": [
            "Kartennummer ******1558: Sie möchten einen \"Online-Abschluss\" durchführen: Bitte bestätigen Sie den \"Startcode 123456789 \" mit der Taste \"OK\"."
        ]
    },
    "_links": {
        "scaStatus": {
            "href": "/v1/consents/{{consentId}}/authorisations/{{authorisationId}}"
        }
    },
    "scaStatus": "scaMethodSelected"
}
```

For chipTAN method QR the PNG File is delivered (the QR code) as a base64 encoded string in the field image.

2.2.1.5 Example of select method response - flicker:

```
{
    "chosenScaMethod": {
        "authenticationType": "CHIP_OTP",
        "authenticationVersion": "1",
        "name": "FLICKER"
    },
    "challengeData": {
        "data": [
            "Kartennummer ******1558: Sie möchten einen \"Online-Abschluss\" durchführen: Bitte bestätigen Sie den \"Startcode 123456789\" mit der Taste \"OK\"."
        ],
        "additionalInformation": "1008123456789"
    },
    "_links": {
        "scaStatus": {
            "href": "/v1/consents/{{consendId}}/authorisations/{{authorisationId}}"
        }
    },
    "scaStatus": "scaMethodSelected"
}
```

The field additionalInformation contains the startcode for the flicker code initialization and it is also included in array data.

2.3 SCA Exemptions

The derogation provided in Article 10 of the RTS has been implemented. Thus, a PSU has update its consent every 180 days using a SCA. During the 180 days period the PSU can access payment transactions executed in the last 90 days through one or more designated payment accounts without 2nd factor.

Whitelisting, low-value transaction and transaction risk analysis (Article 13, 15 and 16 of the RTS) are not supported in the first approach of the XS2A interface.

3. Account Information Service (AIS)

3.1 Create Consent

Account information can only be requested after a consent has been created. A TAN (2.factor) is required for the creation of a consent. Here again the embedded approach is used.

After the PSU has created a consent for an account, the TPP receives a consentId. **In general,** the consentId is valid for 180 days, i.e. it can be used to receive account information (of the last 90 days) for the related account in this and the following sessions in the 180 day period. Account information older 90 days can be requested in the very first Session. After that account information older 90 day are not covered by the consent anymore and an additional SCA (TAN) is required.

A consent can become invalid within the 180 days period, if:

- The PSU, TPP or ASPSP (DKB) revokes the consent.
- The consent was created for a specific period of time (validUntil).
- The consent becomes invalid after 180 days without renewal.

3.2 Get Account Information

After the creation of the consent, the consentId is used as header to access the associated account and read account list, read account details, read account balances and read transactions.

3.3 Account Retrievals 4 Times / Day

In accordance with BerlinGroup specifications, the interface has an integrated counting function that limits access to account data (here: transactions and balances) without customer involvement to a maximum of four queries per 24 hours.

4. Payment Initiation Service (PIS)

4.1 Payment SCA

To confirm payments, a SCA takes place for each transaction. A, **dynamic linked** TAN is required for the authorization of a payment. Here again the embedded approach is used. As with all SCA procedures used by the DKB, the TAN must have been transmitted to the API within 5 minutes.

After the authorization of a payment, the API responds if the payment was **accepted or declined**, similar to the DKB Online Banking. It is not possible to send an automated confirmation that the payment was booked successfully because of the DKB **batch-booking** approach.

4.2 Payment Products

Description	Characteristic	Decision
Supported Single Payment products	sepa-credit-transfers	YES
	instant-sepa-credit-transfers	NO
	target-2-payments	NO
	cross-border-credit-transfers	YES
Supported Bulk Payment products	sepa-credit-transfers	YES
	instant-sepa-credit-transfers	NO
	target-2-payments	NO
	cross-border-credit-transfers	NO
Supported Periodic Payment products	sepa-credit-transfers	YES
	instant-sepa-credit-transfers	NO
	target-2-payments	NO
	cross-border-credit-transfers	YES

4.3 Payment Services

Description	Code	Characteristic	Decision
Standing order (Dauerauftrag)	HKCDB	Standing order status	YES
	HKCDE	Standing order create	YES
	HKCDL	Standing order delete	YES
	HKCDN	Standing order change	NO
	HKCDU	Standing order suspend	NO
Scheduled transfer (Terminüberweisung)	HKCSB	Scheduled transfer status	YES
	HKCSE	Scheduled transfer create	YES
	HKCSL	Scheduled transfer delete	YES
	HKCSA	Scheduled transfer change	NO

Moreover, it is not possible for the PSU to identify the TPP as the creator or initiator of the orders by means of a bank statement. The TPP is requested to provide the customer with the relevant information.

5. Fund Confirmation Service (FCS)

5.1 FCS Consent

FCS consents will be created by ASPSP (DKB) as suggested from Berlin Group. The TPP has to ask the PSU (DKB customer) to establish a FCS consent at the DKB. The PSU has to inform DKB with TPP registration number (which is part of the QWAC), the card number of the issued card (if available) and the account that should be linked to the FCS consent. After the creation of the consent, the TPP can check if enough funds are available on the related account.

Note: DKB would like to establish the "Extended value add services" of the Berlin Group in a further release. Then it should be possible to create a FCS consent via the TPP webpage, in a similar way like an AIS consent.

5.2 Fund Confirmation

Before the fund confirmation check a PreAuth is required. The ASPSP (DKB) checks if the certificate is valid and if a consent for the requested account exists.

The amount provided in this query or used for the evaluation corresponds to the regular account balance. Reserved amounts and the credit line are **not** included in the check.

6. Contingency Measures

6.1 Fallback

A TLS-connection between TPP and ASPSP has to be established always including TPP authentication by using the TPPs QWAC certificate. The same mechanism as describes in PSD2 API Solution - Documentation for TPPs in chapter 3.1 is applied.

This connection must be used to call the following endpoint using an empty request body:

Headers

- X-Request-ID {generated UUID to identify the request}

URL

- https://certcheck.dkb.de/api/v1/validation

HTTP-Method

- POST

Request Body

```
{
 }
```

More details can be obtained from the OpenApi file which can be found in the API Store (XS2A-NextGenPSD2BerlinGroup - 1.3.6 → Documentation → Other)

6.1.1 Successful Response

A successful certificate validation is returned with the HTTP-Response Code 200. The scopes field describes the PSD2 scopes contained in the TPPs QWAC certificate. Using the URL contained in the loginURL field the TPP can redirect to the banking solution.

```
{
            "scopes": [
                    "AIS",
                    "PIIS",
                    "PIS"
                ],
                "loginUrl": https://www.dkb.de/banking
            }
```

6.1.2 Error Response

In case of an invalid certificate a possible response is shown below.

```
{
 "errorCode": "CERTIFICATE_INVALID",
"errorMessage": "The certificate 4d60bcc12d1f70957ebde440c412bd7a is
invalid."
}
```

7. Miscellaneous

7.1 Service Information

Technical support for TPPs is available through our service provider Crealogix.

Contact: kundenportal@crealogix.com

Important: Please include the following information in your request (if available):

- Requested endpoint
- Request ID
- Timestamp
- TPP-ID

7.2 Updates

Future Updates will be published in advance on https://www.dkb.de/info/psd2-api/.