Technical Manual for Implementing the “Account Direct Access” API

23 March 2022

Version 1.7

CHANGE LOG

Version	Date of effectiveness	Change description
1.8	9.6.2022	Bug fixing, added recommedation in ch. 9
1.7	23.3.2022	New endpoint /account Depreccated endpoint /account-ids
1.6	6.1.2022	Divide registration part.
1.5	2.12.2021	Link to html prototype, Ch. 6 Unique id transaction.
1.4	15.10.2021	Extend SW statements validation to 12 months
1.3	15.6.2021	Add more detail to chapter 9
1.2	22.4.2021	Add parameter state to step 2
1.1	20.1.2020	Add notification chapter 9.
1.0	2.5.2020	Public document

TABLE OF CONTENTS

Change log 1
Background.. 2
Registration within the API portal 3
Functionality description 4
Registrations and security 5
5. Replace IBAN with an account identification number 5
6. Transaction history downloads 6
7. Download account balance.. 7
8. Download account statement.. 8
9. Notification of changes on the account.. 9

BACKGROUND

- In connection with the relationship with KB, the Partner shall comply with the rules described in the Code of Prudent Conduct.
- Before reading this Manual, we advise you to read the general description of the service. It also specifies the requirements and actions that must be met or taken prior to the technical connection to the Account Direct Access API.

REGISTRATION WITHIN THE API PORTAL

In order to actually connect to the service, it is necessary to register within the production API portal, register an application (third party to consume the issued API) and generate an API key. For more details, see the Production API Portal manual.

It is then necessary to subscribe the application to API:

- Account Direct Access
- OAUTH2
- Client registration

The swagger for the aforementioned service contains a detailed description of all attributes and should be sufficient to understand the functionality. In general, the swagger documentation is written in English.

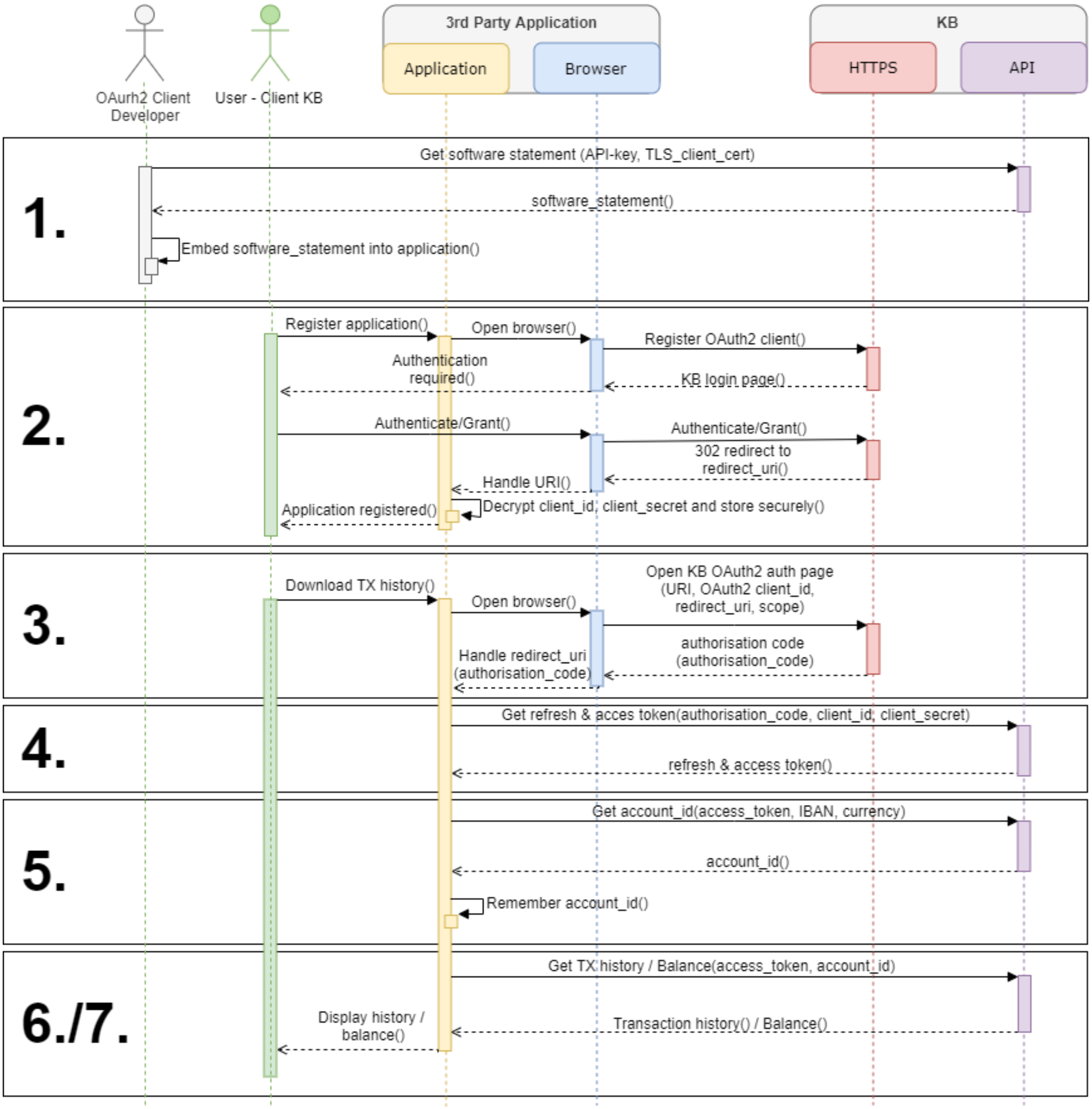
BANK'S RECOMMENDATIONS:

- Keep the API key secure;
- Access your API key remotely from your application;
- Generate a new API key for each new version of your application.

FUNCTIONALITY DESCRIPTION

Sample application screenflow from KB client side: https://api.kb.cz/adaa-flow/

[Sample API requests]()


OAurh2 Client Developer
User - Client KB
3rd Party Application
Application
Browser
KB
HTTPS
API
1.
Get software statement (API-key, TLS_client_cert)
software_statement()
Embed software_statement into application()
2.
Register application()
Open browser()
Register OAuth2 client()
Authentication required()
KB login page()
Authenticate/Grant()
Authenticate/Grant()
302 redirect to redirect_uri()
Handle URI()
Application registered()
Decrypt client_id, client_secret and store securely()
3.
Download TX history()
Open browser()
Open KB OAuth2 auth page (URI, OAuth2 client_id, redirect_uri, scope)
Handle redirect_uri (authorisation_code)
authorisation code (authorisation_code)
4.
Get refresh & acces token(authorisation_code, client_id, client_secret)
refresh & access token()
5.
Get account_id(access_token, IBAN, currency)
account_id()
Remember account_id()
6./7.
Get TX history / Balance(access_token, account_id)
Display history / balance()
Transaction history() / Balance()

Registrations AND SECURITY

Part 1-4 from process are described in the Technical Manual for Registration and Security

API Business Suite

5. REPLACE IBAN WITH AN ACCOUNT IDENTIFICATION NUMBER

Objective	**Obtain accountID for the given IBAN**
Type	REST API
Called API	[GET] https://api.kb.cz/open/api/adaa/v1/accounts
Requirements	-
Request data	access_token
Response data	Iban, currency, account_id

For the purpose of enhanced security, it is not possible to send IBAN in the header in its open form; it is necessary to apply for the so-called identification number (i.e. encrypted IBAN). Transaction history or account information downloads are called using the identification number - accountId. For detailed API description, see the ADAA swagger documentation.

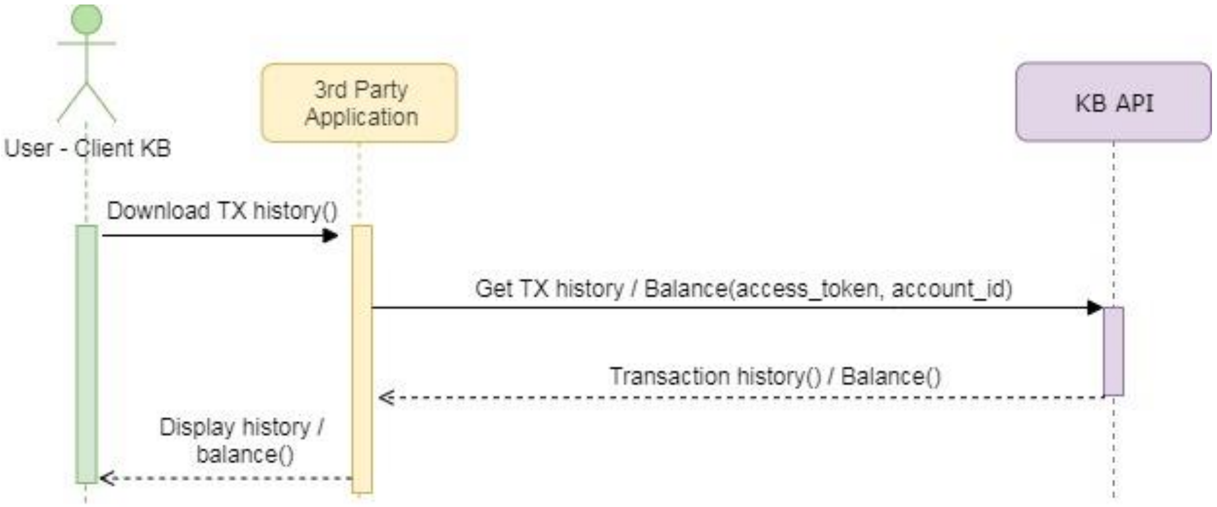
Bank's recommendations:

- Keep the account_id in the application for later use. It is not necessary to call the endpoint again.

6. TRANSACTION HISTORY DOWNLOADS

Objective	Download transaction history
Type	REST API
Called API	[GET] https://api.kb.cz/open/api/adaa/v1/accounts/accountId/transactions
Requirements	Valid access_token, account_id
Request data	account_id, access_token
Response data	Transaction history

The endpoint is used to download transaction history for the account in question. For detailed API description, see the ADAA swagger documentation.


User - Client KB
3rd Party Application
KB API
Download TX history()
Get TX history / Balance(access_token, account_id)
Transaction history() / Balance()
Display history / balance()

Bank’s recommendations:

- We recommend using the notification system, see chapter - Notification of changes on the account.
- **Without notifications, you can only download transaction history once per 61 minutes. Other queries may be rejected**
- The attribute references.accountServicer can be used as a unique identifier. Typical usage e.g. card blocking and then booking
- Within paginh it is necessary to call next page within 10 seconds, otherwise the call is rejected.
- KB user must have access to accounts through one of the KB channels
  .

7. DOWNLOAD ACCOUNT BALANCE

Objective	Download account balance
Type	REST API
Called API	[GET] https://api.kb.cz/open/api/adaa/v1/accounts/balances
Requirements	Valid access_token, account_id
Request data	account_id, access_token
Response data	Account balance

The endpoint is used to download account balance for the account in question. For detailed API description, see the swagger documentation.

8. DOWNLOAD ACCOUNT STATEMENT

Cíl	Obtain statement metadata (include statementId)
Typ	REST API
Volané API	[GET] https://api.kb.cz/open/api/adaa/v1/statements
Požadavky	Valid access_token, account_id
Request data	account_id, access_token, dateFrom
Response data	Statement metadata (include statemen_iId – id of specific PDF file)

The endpoint is used to download information about account statements (metadata). Metadata contains important parameter – statement_id – id of specific PDF file (for downloading specific PDF file; see endpoint below). For detailed API description, see the ADAA swagger documentation.

Cíl	Download account statement in PDF format
Typ	REST API
Volané API	[GET] https://api.kb.cz/open/api/adaa/v1/statements
Požadavky	Valid access_token, account_id, statement_id
Request data	account_id, access_token, statement_id
Response data	Statement in PDF format

The endpoint is used to download specific statement / file in PDF format. For detailed API description, see the ADAA swagger documentation.

9. NOTIFICATION OF CHANGES ON THE ACCOUNT

To receive notifications of account changes, you must complete the following steps:

1. register to receive notifications
   - part of the registration is the transfer of the URL address of the above API (for sending notifications)

2. implement on its side the API to which notifications will be sent
   - KB will call this API if a change is detected on the account
   - KB will only provide information on the number of changes, for specific data, it is necessary to download the transaction history

You can have unlimited registrations for notifications.

Register your endpoint to receive notifications

Objective	**Registration for receiving notifications**
Type	REST API service
Called API	[POST], https://api.kb.cz/open/api/adaa/v1/accounts/accountId/transactions/event-subscriptions
Requirements	Valid access_token, account_id,
Request data	account_id, access_token, eventApiUrl (API address for receiving notifications, see above), eventApiKey (API key for calling API for receiving notifications)
Response data	Registration information: • subscriptionId - assigned unique registration id • eventApiUrl - API address for receiving notifications • eventApiVersion - before creating the registration, a check call of the API for subscription of notifications (is-alive method) is performed and if it is successful, the version of the given API is saved • status - the result of the call - according to the returned answer, the relevant status is set: ○ ACTIVE - API (for subscription of notifications) performed successfully (returned 204) ○ SUSPENDED - API call (for notifications) failed - API is unavailable (returned 500) - in this case the call is made automatically later (in several iterations within a total of 24 hours after API failure; then the expiration of this time is considered unavailable and registration set to STOPPED status) ○ STOPPED - API call (for receiving notifications) failed (401, 403, 404 response returned)

	• statusNarrative - the reason for the status, for example (Connection Timeout: https://company.org/event-api/v1/subscriptions/71f415f4-412d-4c55-af05-15d1e0389f8f/events)

Registration is a necessary part of receiving notifications. As part of the registration, an API URL is passed to receive notifications. The output of the registration is a unique id (subscriptionId), which identifies to which account the notifications relate.

If it is a requirement to register multiple accounts for subscription notifications, it is necessary (for each account) to re-register (new subscriptionId).

Status of Registration:

The registration (notification subscription) statuses are listed above (ACTIVE, SUSPENDED, STOPPED). The statuses are set according to the detected availability / behavior of the API for receiving notifications. From the SUSPENDED state, it is possible to automatically switch to ACTIVE, provided that the call is successful (detected in several iterations, see above). The STOPPED state is final - registration in this state can no longer be renewed and the only option is a new registration.

Objective	**Check of the registration status for receiving notifications**
Type	REST API service
Called API	[GET], https://api.kb.cz/open/api/adaa/v1/accounts/accountId/transactions event-subscriptions
Requirements	Valid access_token, account_id, subscriptionId
Request data	Account_id, access_token, subscriptionId
Response data	Registration information (same scope as for the POST method, see above).

The endpoint is used to determine the status of registration / subscription of notifications. For example, if there has been a long-term outage of the api for receiving notifications (and as a result, on the registration side, the status changes, eg to STOPPED). Sending a notification will not work and the way to find out the situation is to call this method (GET).

Objective	**Cancellation of registration - cancellation of subscription of notifications**
Type	REST API service
Called API	[DELETE], https://api.kb.cz/open/api/adaa/v1/accounts/accountId/transactions event-subscriptions
Requirements	Valid access_token, account_id, subscriptionId
Request data	Account_id, access_token, subscriptionId
Response data	Confirmation of cancellation of registration.

The endpoint is used to delete the registration if the subscription of notifications (for a defined account) is no longer required

Your API to receive notifications

Objective	Notification Subscription API
Type	REST API service
Called API	[POST], https://company.org/event-api/v1
Requirements	The structure of the API corresponds to the example of github adaa-event-api.yaml
Request data	subscriptionId (id assigned within the registration for subscription of notifications), eventCount (number of changes)
Response data	Information on receipt of notification.

The endpoint is used to receive a notification when there is a change in the account (new event in the transaction history). The API issues a notification to the recipient (third party). The main benefit of this solution is that there is no need to regularly download the transaction history to identify the change. The download can be performed only at the moment of obtaining the notification from KB.

You can see example https://github.com/komercka/adaa-eventapi-spring-boot-example

Bank's recommendations:

- Please check your endpoints. If it does not respond within 5 seconds, it will be set to "Suspended" as if it did not respond..
- If you want to limit network traffic, incoming requests from the bank come from IP addresses:
  - 194.50.202.179
  - 194.50.226.179