

The Sandbox Manual on the Account Information Service

Change log

Date	Version	Description
26.02.2020	8	Document and screen updates
29.05.2020	9	Document edit
18.03.2021	10	In Chapter 11.5, the CLBD balance was replaced by the CLAV balance and the numbering of the other chapters was corrected
29.11.2022	11	Document and screen updates

Contents

1.	Account Information Service (AIS)	4
2.	Account Information Service (AIS) Calling API Sandbox	4
3.	Issuing a Certificate	4
4.	Definition of AIS Calling Testing Accounts	4
5.	Error reporting	5
6.	Definition of the Mock	5
7.	Accounts list AISP calling	6
8.	Account balance check AISP calling	6
9.	Transactions overview AISP calling	8
10.	Procedure for generating AISP key/token	10
10.1	Prerequisites for key/token generation	10
10.2	Access the application menu and select desired application	10
10.3	Selection of application menu functionality	11
10.4	Generate certificate for PISP service	12
11.	AISP Mock Calling Methods	13
A)	Access to the Sandbox through the mock API portal	13
11.1	“List of client’s accounts” AIS mock calling through the API portal	13
	Filling in the required fields of the “Returns list of client’s accounts” operation	14
	“Returns list of client’s accounts” operation error message	15
11.2	“Account balance check” AIS mock calling through API portal	16
	Filling in the required fields of the “Account balance check” operation	17
	“Current account balance” operation error message	18
11.3	“Overview of transactions in the client’s payment account” AIS mock calling through the API portal	19
	Filling in the required fields of the “Account’s transaction history” operation	20
	“Account’s transaction history” operation error message	21
B)	Access to the Sandbox through API Direct Calling	22
11.4	List of client’s accounts – via resource direct calling	22
11.5	Client’s Payment Account Balance	25
11.6	Client’s payment account transactions overview	29
12.	PSD2 glossary – selected terms	53

1. Account Information Service (AIS)

The following information is provided as part of the AIS service:

- List of client's accounts
- Account balance check:
 - o Available balance – CLAV (Closing Available)
 - o Current balance – PRCD (Previously Closed Booked)
- Transaction history for up to 24 months

Services call are authorized by the client and are valid for 90 days.

Komerční banka has based its approach on the unified structure and format of information defined by the Czech Banking Association in the [Czech Open Banking Standard](#).

The information provided through API Open Banking is in both Czech and English.

The allowed character set is based exclusively on the SWIFT character set (i.e., exclusively without diacritics).

Only one query can be sent and processed during a single call.

2. Account Information Service (AIS) Calling API Sandbox

Through the Sandbox, third parties may have a trial (mock) of a service providing information concerning the payment account of a client of Komerční banka, a.s, branches of the foreign bank (hereinafter referred to as Komerční banky).

Any entities, not only the third parties with a PSD2 services licence, may access the API Sandbox. However, they must register at KB's API portal <https://api.kb.cz/portal/?tenant=api.kb.cz>. Failing this, they cannot utilize the Sandbox services. The procedure for registration is described in the document *API Sandbox Registration*.

3. Issuing a Certificate

A certificate is necessary for the production calling and PSD2 Sandbox. After the registration, Komerční banka will provide the third parties with certificates to be used on the Sandbox, **based on their request sent at the electronic address api@kb.cz**. The Sandbox certificates are not intended for production use. The production unit will reject and monitor such calls.

4. Definition of AIS Calling Testing Accounts

The mocked accounts are defined in the tables below for the purpose of calling. Detailed information can be obtained by calling the resource for the list of accounts.

Attributes returned by the answer to the accounts list query:

- IBAN – number of the account used for AIS mock calling in the IBAN format.
- Id – identification of the account number in an encrypted form used for further calling of the services (account balance AIS calling, transactions overview AIS calling).
- Other – identification of the account number in the bban form.
- Currency – currency in which the account is denominated.
- Bank code (bankCode) – code of the bank holding the account.
- Country code (countryCode) – code of the country in which the bank holding the account resides.
- bic - Business Identification Code, a series of numbers that uniquely identifies the bank around the world.
- name18N – name given to the account by the client (e.g. My Best KB Account).
- product18N – name given to the account by KB (e.g. MůjÚčet Plus).

Additional information for testing and AIS calling:

- Authorisation – information whether AIS calling is possible for the given account (or account type).

A list of accounts to which the query (account balance/transaction history) can be made:

BBAN	IBAN	Currency	Permission
900930427310227	CZ8501000900930427310227	CZK	Yes
900930427430237	CZ0301000900930427430237	CZK	Yes

5. Error reporting

Reporting quarantined errors or calling them always takes place via the mailbox api@kb.cz. The e-mail sent must contain the following information, in case the required information is missing, it will not be possible to process the query or error.

PSD2 API: CZ, SK

Environment: Sandbox, Production

Whether it was called from FE Sandbox incl. the type and version of the browser used or, in the case of a BE call, the name and version of the program for the BE call

Request type

Date and time of the call

IP address

The error and its most accurate description, which can be supplemented with the appropriate screenshot

6. Definition of the Mock

Parameters of dynamic calling (in our mock) are defined, which a third party can change within calling. Appropriate responses or, as the case may be, error codes are returned depending on the request parameters used.

The Access and Refresh tokens can be obtained in two ways:

- Through the API portal – after the user is registered in the Sandbox and logged in
 - The access and refresh tokens are generated directly in the API portal
 - It is necessary to choose for which scope the token is required
- By direct API calling – i.e., calling of a given resource. Before direct calling is launched, the user must register at the portal first – see the chapter [User registration / TPP for AISP mock service calling](#).
 - Caution: the procedure of registration in the Sandbox differs in this step from the procedure of registration in the production. In the Sandbox, the user must register into the portal and create the application subsequently – this all through the API portal – while in the production, the application is only registered (created) by calling the registration resource.

7. Accounts list AISP calling

KB makes it possible to call the AIS service in the Sandbox with the below parameters and logics:

AISP Parameter	Input values	Activity/Response
AISP service calling certificate	KB test certificates with a given scope.	If the provided testing certificate is used, the balance is returned (depending on the scope validations).
	Any other certificate.	If any other certificate is used, an error is returned - Authentication with an invalid certificate. 403 FORBIDDEN - Invalid certificate or token.
	Calling without a certificate.	If TPP uses no certificate, an error is returned. 401 UNAUTHORISED - Missing certificate or access token.
x-user-id	KBI ID (KB will provide two)	KB ID – returns list of accounts with AISP consent
		KB ID – returns empty set (no consent is provided for any account)
Page	Positive integer number.	Returns a given page depending on the request.
	Positive integer number.	If the entered number is greater than the number of pages, an error is returned: PAGE_NOT_FOUND.
Size	Positive integer number.	Returns a response depending on the entered parameter.
Sort	Comma-separated list of fields for Sorting, ranked according to their importance.	List sorted according to the entered parameters (only according to IBAN)
		If the value is invalid, an error is returned: FIELD_INVALID
Order	Comma-separated list of manners of sorting (ASC, DESC). The order corresponds to the order of fields in the sort parameter.	List sorted according to the entered parameters (IBAN only)
		If the value is invalid, an error is returned: FIELD_INVALID
x-request-id	Unique identifier for each request specified by TPP. It is recommended to use a UUID format with a length of 36 to 60 characters.	If the maximum length of the field is exceeded, it is returned to TPP error ERROR: {"errors": [{"error": "ERR_CODE_400", "scope": "x-request-id"}]}
User-Involved	The flag false/true identifies if the end user is online and the request was made based on his activity.	If one of the true/false values is not filled in, an error is returned
TPP-Identification	The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'.	If the maximum length is exceeded, we do not return any error.
TPP-name	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'.	If there is an empty field or the maximum length of the field is exceeded (max. length of 100 fields), an error is returned {"errors": [{"error": "FIELD_MISSING", "scope": "Tpp-Name"}]}

8. Account balance check AISP calling

KB makes it possible to call the AIS service in the Sandbox with the below parameters and logics:

AISP Parameter	Input values	Activity/Response
AISP service calling certificate	KB test certificate.	If the provided testing certificate is used, the balance is returned (depending on the validations below and on the scope validations). The validations are the same as in the production.

	Any other certificate.	If any other certificate is used, an error is returned - Authentication with an invalid certificate. 403 FORBIDDEN - Invalid certificate or token.
	Calling without a SSL certificate.	If TPP uses no certificate, an error is returned. 401 UNAUTHORISED - Missing certificate or access token.
id	The provided accounts ids will be the same as those contained in the accounts list response.	Response depending on the account definition.
	Erroneous account id	An error is returned: ID_NOT_FOUND.
currency	ISO currency code corresponds to the currency of the account	Response depending on the account definition.
	ISO currency code does not correspond to the currency of the account	An error is returned: AC09 - InvalidAccountCurrency
x-request-id	Unique identifier for each request specified by TPP. It is recommended to use a UUID format with a length of 36 to 60 characters.	If the maximum length of the field is exceeded, it is returned to TPP error ERROR: {"errors": [{"error": "ERR_CODE_400","scope": "x-request-id",}]}
User-Involved	The flag false/true identifies if the end user is online and the request was made based on his activity.	If one of the true/false values is not filled in, an error is returned
TPP-Identification	The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'.	If the maximum length is exceeded, we do not return any error.
TPP-name	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'.	If there is an empty field or the maximum length of the field is exceeded (max. length of 100 fields), an error is returned {"errors": [{"error": "FIELD_MISSING", "scope": "Tpp-Name" }]}

9. Transactions overview AISP calling

KB makes it possible to call the AIS service in the Sandbox with the below parameters and logics:

AISP Parameter	Input values	Activity/Response
AISP service calling certificate	KB test certificates.	If the provided testing certificate is used, the balance is returned (depending on the scope validations). The validations are the same as in the production.
	Any other certificate.	If any other certificate is used, an error is returned - Authentication with an invalid certificate. 403 FORBIDDEN - Invalid certificate or token.
	Calling without a SSL certificate.	If TPP uses no certificate, an error is returned. 401 UNAUTHORISED - Missing certificate or access token.
id	The account id from the accounts list response will be used.	Returns the transaction history according to the mock.
	Erroneous account id	An error is returned: ID_NOT_FOUND.
Page	Positive integer number.	Returns the given page depending on the request.
	Positive integer number.	If the entered number is greater than the number of pages, an error is returned: PAGE_NOT_FOUND.
Size	Positive integer number.	Returns a response depending on the entered parameter.
Sort	Comma-separated list of fields for Sorting, ranked according to their importance – we will only support sorting by the transaction date.	List sorted according to the entered parameters (date only). If the value is invalid, an error is returned: FIELD_INVALID.
Order	Comma-separated list of manners of sorting (ASC, DESC). The order corresponds to the order of fields in the sort parameter.	List sorted according to the entered parameters. If the value is invalid, an error is returned: FIELD_INVALID.
Currency	ISO currency code corresponds to the currency of the account	Response depending on the account definition.
	ISO currency code does not correspond to the currency of the account	An error is returned: AC09 – InvalidAccountCurrency.
fromDate	Date and time from which the transaction history should be displayed	- Returns an overview of transactions depending on the required date - Starting/final date is not matching – invalid date or format. AISP - wrong interval of the transaction history: a) The date is older than 24 months b) toDate > today c) toDate < fromDate d) fromDate and toDate have not yet occurred
toDate	Date and time until which the transaction history should be displayed [including]	
x-request-id	Unique identifier for each request specified by TPP. It is recommended to use a UUID format with a length of 36 to 60 characters.	If the maximum length of the field is exceeded, it is returned to TPP error ERROR: {"errors": [{"error": "ERR_CODE_400", "scope": "x-request-id"}]}
User-Involved	The flag false/true identifies if the end user is online and the request was made based on his activity.	If one of the true/false values is not filled in, an error is returned
TPP-Identification	The identification (licence number) of	If the maximum length is exceeded, we do not return any error.

	the original TPP that created the request. Eg. 'CZ013574-15'.	
TPP-name	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'.	If there is an empty field or the maximum length of the field is exceeded (max. length of 100 fields), an error is returned { "errors": [{"error": "FIELD_MISSING", "scope": "Tpp-Name" }] }

10. Procedure for generating AISP key/token

10.1 Prerequisites for key/token generation

The user is registered and logged in to the Sandbox portal.

10.2 Access the application menu and select desired application

The logged-in user will enter the application via the „Applications“ link at the top screen.

API Portál

APIs Applications

PREMYSL_HRIBA@KB.CZ@API.KB.CZ

APIs

Choose category Choose status

What do you want to search? SEARCH

PSD2 services

AISP-SANDBOX

Version: v1

This is KB REST API supposed to be used by AISP (Account Information Service Provider) to retrieve the list of client's accounts, account balance and transaction history

VIEW MORE

CISP-SANDBOX

Version: v1

This is KB REST API supposed to be used by CISP (Card Issuing Service Provider) to check the client's account balance (whether specific amount of money is available on client's account)

VIEW MORE

OAuth2-SANDBOX

Version: v1

SANDBOX KB IDP Authorization API

VIEW MORE

PISP-SANDBOX

Version: v1

KB PISP Component API

VIEW MORE

PREJÍT NA WEB KB.CZ

API@KB.CZ

KB API Portál

APIs Applications

PREMYSL_HRIBA@KB.CZ@API.KB.CZ

+ ADD APPLICATION

Applications

An application is a logical collection of APIs. Applications allow you to use a single access token to invoke a collection of APIs and to subscribe to one API multiple times with different SLA levels. The DefaultApplication is pre-created and allows unlimited access by default.

Filter by ...

Name	Tier	Workflow Status	Subscriptions	Actions
DefaultApplication	Unlimited	ACTIVE	1	  
Test_KB	Unlimited	ACTIVE	4	  

Show 10 entries Showing 1 to 2 of 2 entries

1

10.3 Selection of application menu functionality

User selects „SANBOX KEYS“ in application menu

The screenshot displays the KB API Portal interface. At the top, there is a navigation bar with the KB logo, 'API Portál', and links for 'APIs' and 'Applications'. A user profile dropdown shows 'PREMYSL_HRIBA@KB.CZ@API.KB.CZ'. Below the navigation bar, the main content area is titled 'Test_KB'. It features three tabs: 'DETAILS', 'SANDBOX KEYS' (which is highlighted with a red box), and 'SUBSCRIPTIONS'. Under the 'SANDBOX KEYS' tab, the following information is displayed:

Status	APPROVED
Unlimited	Allows unlimited requests
Per Token Quota	This feature allows you to assign an API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.
Description	Not Given

11. AISP Mock Calling Methods

TWO AISP MOCK CALLING METHODS

A) Access to the Sandbox through the mock API portal

[“List of client’s accounts” AIS mock calling through the API portal](#)

[Account balance check” AIS mock calling through API portal](#)

[Overview of transactions in the client’s payment account” AIS mock calling through the API portal](#)

B) Access to the Sandbox through API Direct Calling

[List of client’s accounts – via resource direct calling](#)

[Client’s Payment Account Balance](#)

[Client’s payment account transactions overview](#)

A) Access to the Sandbox through the mock API portal

11.1 “List of client’s accounts” AIS mock calling through the API portal

It means calling the “Returns list of client’s accounts” operation.

The user chooses an operation he/she wishes to test. In this case, it is “Returns list of client’s accounts”. This operation displays the statement of an account of the logged-in client arranged according to the given requirements.

GET

Returns list of client's accounts
/my/accounts

SHOW LESS ^

Implementation Notes
Returns the list of accounts of the logged-in client sorted and paged according to the request parameters

ⓘ Required Scopes

Key	Name
aisp	aisp

Filling in the required fields of the “Returns list of client's accounts” operation

The user wishing to display a list of his/her accounts fills in all fields with values in an appropriate format. If everything is done properly, the sorted list is displayed. If any of mandatory fields is not filled in, the report is not displayed and the blank fields are highlighted in red.

Headers

Header	Description	Type	Other
--------	-------------	------	-------

Parameters

Parameter	Value	Description	Parameter Type	Data Type
size	20	The number of rows per page (in pagination mode)	query	integer
page		The page number to be returned (in pagination mode), 0 is default	query	integer
sort		The list of field to be used to sort by delimiter by comma ordered by significance	query	string
order		Comma-separated list of sorting methods (asc, desc). The order matches the order of the fields in the sorting parameter	query	string
X-Request-Id		Unique identifier for each request specified by TPP. It is recommended to use a UUID format with a length of 36 to 60 characters.	header	string
User-Involved	true	The flag identifies if the end user is online and the request was made based on his activity.	header	boolean
TPP-Name	(required)	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'.	header	string
TPP-Identification		The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'.	header	string

“Returns list of client's accounts” operation error message

If any value has been entered incorrectly, one of the following error messages or an error specified in the mock definition will be displayed after pressing the "TRY IT OUT" button, otherwise the result statement will be displayed.

Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Input parameter is invalid	Model Example Value	
		<pre>{ "errors": [{ "error": "ERR_CODE_400", "scope": "x-request-id", "message": "Value of parameter x-request-id is wrong" }] }</pre>	
401	Missing certificate or access token	Model Example Value	
		<pre>{ "errors": [{ "error": "ERR_CODE_401", "message": "Missing certificate or access token" }] }</pre>	
404	Page does not exist	Model Example Value	
		<pre>{ "errors": [{ "error": "PAGE_NOT_FOUND", "message": "The requested page does not exist" }] }</pre>	
415	Invalid message charset	Model Example Value	
		<pre>{ "errors": [{ "error": "RR10", "message": "InvalidCharacterSet" }] }</pre>	
500	Unexpected error occurred	Model Example Value	
		<pre>{ "errors": [{ "error": "ERR_CODE_500", "message": "Internal Server Error" }] }</pre>	

TRY IT OUT

11.2 “Account balance check” AIS mock calling through API portal

In the Sandbox, it means calling the “Account balance check” operation for the purpose of testing.

The user chooses an operation he/she wishes to test. In this case, it is “Account balance check”. This operation displays the account balance.

GET

Account balance check
/my/accounts/{id}/balance

SHOW LESS ^

Implementation Notes
Returns the current account balance of the client's account

 **Required Scopes**

Key	Name
aisp	aisp

Filling in the required fields of the “Account balance check” operation

The user wishing to display a current account balance in a selected account fills in all fields with values in an appropriate format. If everything is done properly, the account balance is displayed. If any of mandatory fields is not filled in, the report is not displayed and the blank fields are highlighted in red.

Headers

Header	Description	Type	Other
--------	-------------	------	-------

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	Account number in hashed format	path	string
currency		ISO code for the currency the balance should be returned in (if not specified the balance is returned in the account's currency)	query	string
X-Request-Id		Unique identifier for each request specified by TPP. It is recommended to use a UUID format with a length of 36 to 60 characters.	header	string
User-Involved	true	The flag identifies if the end user is online and the request was made based on his activity.	header	boolean
TPP-Name	(required)	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'.	header	string
TPP-Identification		The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'.	header	string

“Current account balance” operation error message

If any value has been entered incorrectly, one of the following error messages or an error specified in the mock definition will be displayed after pressing the “TRY IT OUT” button, otherwise the result statement will be displayed.

Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Input parameter is invalid	Model Example Value	
		<pre>{ "errors": [{ "error": "ERR_CODE_400", "scope": "x-request-id", "message": "Value of parameter x-request-id is wrong" }] }</pre>	
401	Missing certificate or access token	Model Example Value	
		<pre>{ "errors": [{ "error": "ERR_CODE_401", "message": "Missing certificate or access token" }] }</pre>	
404	Account ID was not recognized or is invalid	Model Example Value	
		<pre>{ "errors": [{ "error": "PAGE_NOT_FOUND", "message": "The requested page does not exist" }] }</pre>	
415	Invalid message charset	Model Example Value	
		<pre>{ "errors": [{ "error": "RR10", "message": "InvalidCharacterSet" }] }</pre>	
500	Unexpected error occurred	Model Example Value	
		<pre>{ "errors": [{ "error": "ERR_CODE_500", "message": "Internal Server Error" }] }</pre>	

TRY IT OUT

11.3 “Overview of transactions in the client’s payment account” AIS mock calling through the API portal

In the Sandbox, it means calling the “Account’s transaction history” operation for the purpose of testing.

The user chooses an operation he/she wishes to test. In this case, it is “Account’s transaction history”. This operation displays the transaction history of a specific client’s account.

GET

Account's transaction history
/my/accounts/{id}/transactions

SHOW LESS ^

Implementation Notes
Returns the history of transactions for specified client's account

ⓘ Required Scopes

Key	Name
aisp	aisp

Filling in the required fields of the “Account's transaction history” operation

The user wishing to display a transaction history of a selected account fills in all fields with values in an appropriate format. If everything is done properly, the transaction history of the given account is displayed. If any of mandatory fields is not filled in, the report is not displayed and the blank fields are highlighted in red. If the fromDate and toDate values are not filled in, the history will be displayed will as at the current date, or as at 18 March 2018 in the case of the Sandbox.

Headers

Header	Description	Type	Other
--------	-------------	------	-------

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	Account number in hashed format	path	string
fromDate		Starting (first) date of the range of the filtered transactions	query	date
toDate		Ending (last) date of the range of the filtered transactions	query	date
currency		ISO code for the currency the balance should be returned in (if not specified the balance is returned in the account's currency)	query	string
size	20	The number of rows per page (in pagination mode)	query	integer
page		The page number to be returned (in pagination mode), 0 is default	query	integer
sort		The list of field to be used to sort by delimiter by comma ordered by significance	query	string
order		Comma-separated list of sorting methods (asc, desc). The order matches the order of the fields in the sorting parameter	query	string
X-Request-Id		Unique identifier for each request specified by TPP. It is recommended to use a UUID format with a length of 36 to 60 characters.	header	string
User-Involved	true	The flag identifies if the end user is online and the request was made based on his activity.	header	boolean
TPP-Name	(required)	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'.	header	string
TPP-Identification		The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'.	header	string

“Account's transaction history” operation error message

If any value has been entered incorrectly, one of the following error messages or an error specified in the mock definition will be displayed after pressing the "TRY IT OUT" button, otherwise the result statement will be displayed.

Response Messages

HTTP Status Code	Reason	Response Model		Headers
400	Input parameter is invalid	Model	Example Value	
			<pre>{ "errors": [{ "error": "ERR_CODE_400", "scope": "x-request-id", "message": "Value of parameter x-request-id is wrong" }] }</pre>	
401	Missing certificate or access token	Model	Example Value	
			<pre>{ "errors": [{ "error": "ERR_CODE_401", "message": "Missing certificate or access token" }] }</pre>	
404	Account ID was not recognized or is invalid	Model	Example Value	
			<pre>{ "errors": [{ "error": "PAGE_NOT_FOUND", "message": "The requested page does not exist" }] }</pre>	
415	Invalid message charset	Model	Example Value	
			<pre>{ "errors": [{ "error": "RR10", "message": "InvalidCharacterSet" }] }</pre>	
500	Unexpected error occurred	Model	Example Value	
			<pre>{ "errors": [{ "error": "ERR_CODE_500", "message": "Internal Server Error" }] }</pre>	

[TRY IT OUT](#)
[Hide Response](#)

B) Access to the Sandbox through API Direct Calling

11.4 List of client's accounts – via resource direct calling

URI: /my/accounts{?size,page,sort,order}
HTTP Method: GET
Request URL: <https://api.kb.cz/sandbox/aisp/v2/my/accounts>
Authorization: the request **requires** an authorisation by the user/client as part of the API call.
Certification: the request **requires** the use of the third party qualified certificate.
Pagination: yes
Sorting: yes
Filtering: no

Supported encoding: charset=UTF-8

Authorization, Bearer and JWT token are inputted manually!!!

Table 1 Query parameters of the request

PARAMETER	TYPE	MANDA-TORY	PURPOSE
size	Int	No	Pagination. A number of records on the page.
page	Int	No	Pagination. The required page. + Default: 0.
sort	Text	No	A list of fields for sorting separated with a comma and sorted in order of importance.
order	Text	No	A list of types of sorting (ASC, DESC) separated with a comma. The order matches the order of the fields in the 'sort' parameter.

Note: The API supports all optional parameters!!!

Example of the API call curl:

```
curl -X GET --header 'Accept: application/json' --header 'x-request-id: 12345' --header 'Authorization: Bearer INPUT_ACCESS_TOKEN_HERE' 'https://api.kb.cz/sandbox/aisp/v2/my/accounts?size=2&page=1&sort=iban&order=ASC'
```

Table 2 Request header parameters

PARAMETER	TYPE	MANDA-TORY	PURPOSE
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
x-request-id	Text	Yes	A unique identification of the caller's each particular query. The value of this parameter should therefore be generated randomly, and the individual x-request-ids of the same caller within a short time interval should not be identical. This parameter service returns responses to the calling system within response headers.
User-Involved	boolean	Ano	The true/false flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
TPP-Identification	Text	Ne	Identification of the original TPP that created the request . Např.: „CZ013574-15“
TPP-name	Text	Ano	The name of the original TPP that created the request . Např.: „Star Corporation, a.s.“

Example of request headers:

```
{
  "Accept": "application/json",
  "x-request-id": "12345",
}
```

Table 3 Response header parameters

PARAMETER	TYPE	MANDATORY	PURPOSE
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
x- request -id	Text	Yes	Returns the original request id of the API call.

Example of response headers:

```
{
  "date": "Mon, 26 Feb 2018 08:02:40 GMT",
  "cookie": "i18next=cs-CZ",
  "x-request-id": "12345",
  "accept": "application/json",
  "access-control-allow-methods": "GET",
  "x-forwarded-host": "api.kb.cz",
  "host": "api.kb.cz",
  "accept-encoding": "gzip, deflate",
  "cache-control": "no-cache",
  "x-forwarded-server": "api.kb.cz",
  "content-type": "application/json; charset=UTF-8",
  "keep-alive": "timeout=60, max=100",
  "connection": "Keep-Alive",
  "transfer-encoding": "chunked",
  "strict-transport-security": "max-age=16070400; includeSubDomains",
}
```

Komerční banka provides the information about the payment account and transaction history in the following structure:

Table 4 List of Client's Payment Accounts – BASIC ELEMENTS OF THE RESPONSE

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	accounts	[1..1]	±	Collection of client's accounts
++	id	[1..1]	Text	API Identifier of the payment account
++	identification	[1..1]	±	Debtor's account identification
+++	iban	[1..1]	IBAN2007Identifier	IBAN
+++	other	[0..1]	Max35Text	Yesther identifier, e.g. debtor's account number
++	currency	[0..1]	CurrencyCode, ISO 4217	Debtor's account currency
++	servicer	[1..1]	±	
+++	bankCode	[0..1]	Text	
+++	countryCode	[0..1]	CountryCode, ISO 3166	Bank's country code

+++	bic	[0..1]	Max35Text	Bank's BIC
++	nameI18N	[0..1]	Text	Account name
++	productI18N	[0..1]	Text	Product name

JSON – example of a response:

```
{
  "pageNumber": 5,
  "pageCount": 7,
  "pageSize": 100,
  "nextPage": 6,
  "accounts": [
    {
      "id": "C2D2DDBCA5415621A34BB1BB234DC1322EA641A3",
      "identification": {
        "iban": "CZ9501000000001234567899",
        "other": "1234567899"
      },
      "currency": "EUR",
      "servicer": {
        "bankCode": "0100",
        "countryCode": "CZ",
        "bic": "KOMBCZPPXXX"
      },
      "nameI18N": "Muj hlavni osobni ucet",
      "productI18N": "Osobni ucet KB"
    }
  ]
}
```

Table 5 CBA-standard defined error codes for the GET service – List of Client's Payment Accounts

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = the user has not been authenticated
401	UNAUTHORISED	Invalid/ missing certificate = the provider has not been authenticated
403	FORBIDDEN	Invalid certificate / token
404	PAGE_NOT_FOUND	The query concerns a non-existent page
400	PARAMETER_INVALID	The parameter value is not valid
400, 50x		General reason for rejection – will be specified
400	AG01	[TransactionForbidden] – account does not have a consent
400	AG02	[InvalidDebtorAccountNumber] – account validation

JSON – example of an error message body:

```
{
  "errors": [
    {
      "error": "ERR_CODE_500",
      "description": "Internal Server Error"
    }
  ]
}
```


11.5 Client's Payment Account Balance

A balance of the client's particular account according to the account's reference id.

URI: /my/accounts/{id}/balance{?currency}
HTTP Method: GET
Request URL: <https://api.kb.cz/sandbox/aisp/v2/my/accounts/{id}/balance{?currency}>

Authorization: the request **requires** an authorisation by the user/client as part of the API call.
Certification: the request **requires** the use of the third party qualified certificate.

Pagination: no
Sorting: no
Filtering: no

Supported encoding: charset=UTF-8

Authorization, Bearer and JWT token are inputted manually!!!

Table 6 Query parameters of the request

PARAMETER	TYPE	MANDA-TORY	PURPOSE
id	Text	Yes	Client's account system identifier in the hashed format.
currency	Text	No	The required currency of the account in case of multicurrency accounts. The ISO code of the currency in which the balance will be returned (if no currency is specified, the balance is returned in the main currency of the account).

Note: The API supports all optional parameters!!!

Example of the API call curl:

```
curl -X GET --header 'Accept: application/json' --header 'x-request-id: 12345' --header 'Authorization: Bearer INPUT_ACCESS_TOKEN_HERE ' 'https://api.kb.cz/sandbox/aisp/v2/my/accounts/C2D2DDBCA5415621A34BB1BB234DC1322EA641A3/balance'
```

Table 7 Request header parameters

PARAMETER	TYPE	MANDA-TORY	PURPOSE
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
x-request-id	Text	Yes	A unique identification of the caller's each particular query. The value of this parameter should therefore be generated randomly, and the individual x-request-ids of the same caller within a short time interval should not be identical. This parameter service returns responses to the calling system within response headers.
User-Involved	boolean	No	The true/false flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
TPP-Identification	Text	No	Identification of the original TPP that created the request . Např.: „CZ013574-15“
TPP-name	Text	No	The name of the original TPP that created the request . Např.: „Star Corporation, a.s.“

Example of request headers:

```
{  
  "Accept": "application/json",  
  "x-request-id": "12345",  
}
```

Table 8 Account Balance RESPONSE REPORT ELEMENTS

LEVEL	MESSAGE ELEMENT	OCCUR- RENCE	FORMAT TYPE	PRESENTATION
+	balances	[1..1]	±	A collection of the client's payment account balance
++	type	[1..1]	±	A designation of the balance type to which the information about the balance pertains
+++	codeOrProprietary	[1..1]	±	
++++	code	[1..1]	Balance type	Balance type codes
++	creditLine	[0..1]	±	An amount of the agreed authorised overdraft/overdraft facility
+++	included	[0..1]	Boolean	
+++	amount	[0..1]	±	An amount of the agreed authorised overdraft
++++	value	[0..1]	Double	An amount of the agreed authorised overdraft
++++	currency	[0..1]	Text	The currency corresponds to the currency of the account to which the generated statement pertains
++	amount	[1..1]	±	The value/amount of the account balance, depending on the balance type. The currency corresponds to the currency of the account to which the generated statement pertains.
+++	value	[1..1]	Double	Account balance amount
+++	currency	[1..1]	Text	The currency corresponds to the currency of the account to which the generated statement pertains
++	creditDebitIndicator	[1..1]	Text	An indicator showing whether the balance of the account to which the generated statement pertains is greater or less than zero
++	date	[1..1]	±	The date (and time) of the balance of the account to which the generated statement pertains. The format corresponds to the ISO 8601.
+++	dateTime	[1..1]	Text	The date (or date and time) of the balance as per the ISO 8601

JSON – example of a response:

```
{
  "balances": [
    {
      "type": {
        "codeOrProprietary": {
          "code": "PRCD"
        }
      },
      "creditLine": {
        "included": true,
        "amount": {
          "value": 10000,
          "currency": "EUR"
        }
      },
      "amount": {
        "value": 15241.3,
        "currency": "EUR"
      },
      "creditDebitIndicator": "CRDT",
      "date": {
        "date": "2017-04-25T05:00:00.000Z"
      }
    }
  ]
}
```

Table 9 Return types of balances in the “codeOrProprietary” parameter

CODE	DESCRIPTION
PRCD	PreviouslyClosedBooked – initial balance
CLAV	ClosingAvailable - available balance

Table 10 Return types of balances in the “creditDebitIndicator” parameter

CODE	DESCRIPTION
DBIT	Balance is less than zero
CRDT	Balance is greater than or equal to zero

Table 11 Response header parameters

PARAMETER	TYPE	MANDA-TORY	PURPOSE
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
x- request -id	Text	Yes	Returns the original request id of the API call.

Example of response headers:

```
{
  "date": "Mon, 26 Feb 2018 08:02:40 GMT",
  "cookie": "i18next=cs-CZ",
  "x-request-id": "12345",
  "accept": "application/json",
  "access-control-allow-methods": "GET",
  "x-forwarded-host": "api.kb.cz",
  "host": "api.kb.cz",
  "accept-encoding": "gzip, deflate",
  "cache-control": "no-cache",
  "x-forwarded-server": "api.kb.cz",
  "content-type": "application/json; charset=UTF-8",
  "keep-alive": "timeout=60, max=100",
  "connection": "Keep-Alive",
  "transfer-encoding": "chunked",
  "strict-transport-security": "max-age=16070400; includeSubDomains",
}
```

Table 12 CBA-standard defined error codes for the GET service – Client's payment account balance

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Missing access token = the user has not been authenticated
401	UNAUTHORISED	Missing certificate = the provider has not been authenticated
403	FORBIDDEN	Invalid certificate or access token
404	ID_NOT_FOUND	Invalid/unknown account ID
400	AC09	[InvalidAccountCurrency] – the declared account currency does not correspond to the currency in which the client's account held with the bank under the given number is denominated.

400	AG01	[TransactionForbidden] – the query concerning an unauthorised account (the client has not granted his/her consent for PSD2).
400	AC02	[InvalidDebtorAccountNumber] – validation concerning the account status
400, 50x		General reason for rejection – will be specified

JSON – example of an error message body:

```
{
  "errors": [
    {
      "error": "ERR_CODE_500",
      "description": "Internal Server Error"
    }
  ]
}
```

11.6 Client's payment account transactions overview

A paginated list of transactions in the client's selected account.

URI: /my/accounts/{id}/transactions{?fromDate,toDate,currency,size,page,sort,order}

HTTP Method: GET

Request URL: <https://api.kb.cz/sandbox/aisp/v2/my/accounts/{id}/transactions{?fromDate,toDate,currency,size,page,sort,order}>

Authorization: the request **requires** an authorisation by the user/client as part of the API call.

Certification: the request **requires** the use of the third party qualified certificate.

Pagination: yes

Sorting: yes

Filtering: no

Supported encoding: charset=UTF-8

Authorization, Bearer and JWT token are inputted manually!!!

Table 13 Query parameters of the request

PARAMETER	TYPE	MANDATORY	PURPOSE
id	Text	Yes	Client's account system identifier.
currency	Text	No	The required currency of the account in case of multicurrency accounts.
fromDate	Text	No	The initial date and time of the required transaction history. The format corresponds to the ISO 8601.
toDate	Text	No	The final date and time of the transaction history [including]. The format corresponds to the ISO 8601.
currency	Text	No	The required currency of the account in case of multicurrency accounts.
size	Int	No	Pagination. A number of records on the page.
page	Int	No	Pagination. The required page. + Default: 0
sort	Text	No	A list of fields for sorting separated with a comma and sorted in order of importance. The records in the output can only be sorted by the transaction execution date.
order	Text	No	A list of types of sorting (ASC, DESC) separated with a comma. The order matches the order of the fields in the 'sort' parameter.

Note: The API supports all optional parameters!!!

Example of the API call curl:

```
curl -X GET --header 'Accept: application/json' --header 'x-request-id: 12345' --header 'Authorization: Bearer INPUT_ACCESS_TOKEN_HERE ' 'https://api.kb.cz/sandbox/aisp/v2/my/accounts/CZ950100000001234567899/transactions'
```

Table 14 Request header parameters

PARAMETER	TYPE	MANDATORY	PURPOSE
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
x-request-id	Text	Yes	A unique identification of the caller's each particular query. The value of this parameter should therefore be generated randomly, and the individual x-request-ids of the same caller within a short time interval should not be identical. This parameter service returns responses to the calling system within response headers.
User-Involved	boolean	Ano	The true/false flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
TPP-Identification	Text	Ne	Identification of the original TPP that created the request . Např.: „CZ013574-15“
TPP-name	Text	Ano	The name of the original TPP that created the request . Např.: „Star Corporation, a.s.“

Example of request headers:

```
{
  "Accept": "application/json",
  "x-request-id": "12345",
}
```

Table 15 Response header parameters

PARAMETER	TYPE	MANDATORY	PURPOSE
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
x-request-id	Text	Yes	Returns the original request id of the API call.

Example of response headers:

```
{
  "date": "Fri, 02 Mar 2018 15:08:00 GMT",
  "cookie": "i18next=cs-CZ",
  "x-request-id": "12345",
  "accept": "application/json",
  "x-forwarded-host": "api.kb.cz",
  "host": "api.kb.cz",
}
```

```

"accept-encoding": "gzip, deflate",
"x-forwarded-server": "api.kb.cz",
"content-type": "application/json; charset=UTF-8",
"x-cnection": "close",
"transfer-encoding": "chunked",
"strict-transport-security": "max-age=16070400; includeSubDomains",
}

```

Table 16 Client's payment account transaction history – RESPONSE REPORT ELEMENTS

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
+	entryReference	Max35Text	Payment ID number assigned by the bank.
+	amount	Amount	The payment currency expressed in the currency of the account to which the generated statement relates.
++	value	Amount	
++	currency	CurrencyCode	
+	creditDebitIndicator	CreditDebitCode	Indicator showing whether the payment is debited or credited to the account. A debited / credited payment is indicated by one of the below codes. DBIT: debited payment CRDT: all other cases.
+	reversalIndicator	TrueFalseIndicator	Indicator showing whether or not it is a reversal (cancellation). It is expressed using one of the below codes: true = reversal false = no reversal
+	status	Code	Status of the item in the account (credited or debited payment) from the bank's point of view. The statement only shows booked items (BOOK constant) or blocked items (PDNG constant).
+	bookingDate	±	Date of the payment processing/entering in the accounts (booking) by the bank in the ISODate or ISODateTime format, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.STZD.

++	date	ISODate ISODateTime	Date of the payment processing/entering in the accounts (booking) by the bank in the ISODate format, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.sTZD, depending on the type of transaction and on how the bank presents the date and time of the payment processing/entering in the accounts (booking). It is represented as ISODateTime, especially in case of card or cash transactions. where: YYYY = four-digit year MM = two-digit month (01=January, etc.) DD = two-digit day of month (01 through 31) hh = two digits of hour (00 through 23) (am/pm NOT allowed) mm = two digits of minute (00 through 59) ss = two digits of second (00 through 59) TZD = time zone designator (Z or +hh:mm or -hh:mm)
+	valueDate	±	Maturity/value date of the payment in the ISODate or ISODateTime format, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.sTZD.

++	date	ISODate/ISODateTime	Maturity/value date of the payment in the ISODate format, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.sTZD, depending on the type of transaction and on how the bank presents the date and time of the payment date /value. It is represented as ISODateTime, especially in case of card or. cash transactions. where: YYYY = four-digit year MM = two-digit month (01=January, etc.) DD = two-digit day of month (01 through 31) hh = two digits of hour (00 through 23) (am/pm NOT allowed) mm = two digits of minute (00 through 59) ss = two digits of second (00 through 59) TZD = time zone designator (Z or +hh:mm or -hh:mm) where: YYYY = four-digit year MM = two-digit month (01=January, etc.) DD = two-digit day of month (01 through 31).
+	bankTransactionCode	±	Bank transaction code as per the Czech Banking Association code list, assigned to a specific payment. Each bank uses its own code list to identify a payment, which is based on the level 1 to 3 of the transaction code list according to the ČBA Standard for camt.053.
++	proprietary	±	Bank transaction code as per the Czech Banking Association code list, assigned to a specific payment. Each bank uses its own code list to identify a payment, which is based on the level 1 to 3 of the transaction code list according to the ČBA Standard for camt.053.

+++	code	Max35Text	Bank transaction code as per the Czech Banking Association code list, assigned to a specific payment. Each bank uses its own code list to identify a payment, which is based on the level 1 to 3 of the transaction code list according to the ČBA Standard for camt.053.
+++	issuer	Max35Text	Identification of the bank transaction code list issuer, which takes its value from the Czech Banking Association list.
+	entryDetails	±	Turnover details. This level occurs only once in each given item
++	transactionDetails	±	Payment details. This level occurs only once in each given item.
+++	references	±	Set of references that unequivocally identify the payment.
++++	messageIdentification	Max35Text	Received payment identification entered by the client when initiating the payment, or a sequence of the given payment in the payment history statement.
++++	accountServicerReference	Max35Text	Bank reference assigned to the given payment, e.g. when initiated via direct banking services.
++++	paymentInformationIdentification	Max35Text	Other bank reference assigned to the payment allotted by the bank; in case of card payments, a payment card sequence number can be filled in, or alternatively the specific symbol.
++++	instructionIdentification	Max35Text	Payment identification entered by a third party, or alternatively the specific symbol.
++++	endToEndIdentification	Max35Text	Unique identification entered by the client who initiates the payment; it allows for an unequivocal identification of the payment and is forwarded, unaltered, throughout the payment chain, as the case may be, the variable symbol may be entered here.

++++	mandateIdentification	Max35Text	In case of SEPA Direct Debits (SDD), a Unique Mandate Reference is specified for a given SDD as a mandatory field [1..1].
++++	chequeNumber	Max35Text	In case of cheque transactions or card transactions, a cheque number or a card number with asterisks can be inputted here; respectively. The card number is always in the xxxxxxxxxxxx1234 format, where the standard only allows for digits from 0 to 9 (8-28 characters) i.e. without the xxxx substitution.
++++	clearingSystemReference	Max35Text	Code list value defined by the bank to identify the payment type, or a used payment type name. A card association can be identified here in case of card payments.
+++	amountDetails	+	Details concerning the payment amount, especially in case of a converse payment or Cashback.
++++	instructedAmount	+	Amount and currency of the payment expressed in the currency required by the client for the transfer. E.g., in the case of intrabank payments, it is the currency in which the debtor's account is denominated and, at the same time, the payment amount, if the client instructed that the payment should be made in the debtor's account currency.
+++++	amount	Amount	Amount and currency of the payment expressed in the currency in which the client's account is denominated, after the conversion of the amount required by the client for the transfer.
+++++	value	Amount	
+++++	currency	CurrencyCode	
++++	transactionAmount	+	Amount and currency of the payment for cumulated payments and Cashback.
+++++	amount	Amount	Amount and currency of the payment for cumulated payments and Cashback, where the total payment amount including Cashback is filled in.
+++++	value	Amount	
+++++	currency	CurrencyCode	

++++	counterValueAmount	+	Amount and currency of the payment expressed in the currency in which the client's account is denominated, after the conversion of the amount required by the client for the transfer.
+++++	amount	Amount	Ultimate amount and currency of the payment required by the client for the transfer.
++++++	value	Amount	
++++++	currency	CurrencyCode	
+++++	currencyExchange	+	Information about the used currencies and exchange rates.
++++++	sourceCurrency	CurrencyCode	Client's account currency (source/original currency / debtor's account currency for intrabank conversion payments).
++++++	targetCurrency	CurrencyCode	Currency of the payment (target currency / creditor's account currency for intrabank conversion payments).
++++++	exchangeRate	BaseOneRate	Exchange rate used for entering the payment into the accounts. One rate is filled-in only, even in the case of a cross conversion.
++++	proprietaryAmount	+	Amount of cash withdrawn via Cashback.
+++++	type	Max35Text	'CASHBACK' constant is to be filled in.
+++++	amount	Amount	Cashback amount and currency – only the portion of the payment withdrawn in cash via Cashback.
++++++	value	Amount	
++++++	currency	CurrencyCode	
+++	charges	+	Information about charges.
++++++	bearer	Code	Identification of the bearer of the charges (OUR, SHA, BEN).

+++	relatedParties	+	Information about the debtor, debtor's account, ultimate debtor, and the creditor, creditor's account, ultimate creditor, as specified in the payment instruction.
++++	debtor	+	Information about the debtor. Depending on the direction of the payment, filled in for the counterparty.
+++++	name	Max140Text	Debtor's name.
+++++	postalAddress	+	Debtor's postal address.
++++++	streetName	Max70Text	Street name used in the debtor's postal address.
++++++	buildingNumber	Max16Text	Building number used in the debtor's postal address.
++++++	postCode	Max16Text	Postal code used in the debtor's postal address.
++++++	townName	Max35Text	Town name used in the debtor's postal address.
++++++	country	CountryCode	Country name used in the debtor's postal address.
++++++	addressLine	Max70Text	Debtor's postal address in an unstructured format.
+++++	identification	+	Identification of the debtor.
++++++	organisationIdentification	+	Unique identification of the debtor as an organization/legal person.
++++++	bicOrBei	BICIdentifier	Identification of the debtor as an organization/legal person in the form of the BIC or BEI code.
++++++	other	+	Other identification of the debtor as an organization/legal person.
++++++	identification	Max35Text	Other identification of the debtor as an organization/legal person in an unstructured format.

+++++++	schemeName	+	Type of the code identifying the debtor as an organization/legal person.
+++++++	code	Code	Type of the code identifying the debtor as an organization/legal person in the form of a code as per the ISO code list.
+++++++	proprietary	Max35Text	Type of the code identifying the debtor as an organization/legal person in a free text format.
+++++++	issuer	Max35Text	Issuer of the code identifying the debtor as an organization/legal person.
+++++	privateIdentification	+	Unique identification of the debtor as a natural person.
+++++++	other	+	Other identification of the debtor as a natural person.
+++++++	identification	Max35Text	Other identification of the debtor as a natural person in an unstructured format.
+++++++	schemeName	+	Type of the code identifying the debtor as a natural person.
+++++++	code	Code	Type of the code identifying the debtor as a natural person in the form of a code as per the ISO code list.
+++++++	proprietary	Max35Text	Type of the code identifying the debtor as a natural person in a free text format.
+++++++	issuer	Max35Text	Issuer of the code identifying the debtor as a natural person.
++++	debtorAccount	+	Information about the debtor's account. Depending on the direction of the payment, filled in for the counterparty.
+++++	identification	+	Debtor's account type identification.

+++++	iban	IBAN2007Identifier	Debtor's account number in the IBAN international format.
+++++	other	+	Debtor's account number expressed in other/local format.
+++++	identification	Max34Text	Value of debtor's account number expressed in other/local format.
+++++	currency	CurrencyCode	Debtor's account currency.
+++++	name	Max70Text	Debtor's account name.
++++	ultimateDebtor	+	Information about the ultimate debtor. Depending on the direction of the payment, filled in for the counterparty.
+++++	name	Max140Text	Ultimate debtor's name.
+++++	postalAddress	+	Ultimate debtor's postal address.
+++++	streetName	Max70Text	Street name used in the ultimate debtor's postal address.
+++++	buildingNumber	Max16Text	Building number used in the ultimate debtor's postal address.
+++++	postCode	Max16Text	Postal code used in the ultimate debtor's postal address.
+++++	townName	Max35Text	Town name used in the ultimate debtor's postal address.
+++++	country	CountryCode	Country name used in the ultimate debtor's postal address.
+++++	addressLine	Max70Text	Ultimate debtor's postal address in an unstructured format.
+++++	identification	+	Identification of the ultimate debtor.
+++++	organisationIdentification	+	Unique identification of the ultimate debtor as an organization/legal person.

++++++	bicOrBei	BICIdentifier	Identification of the ultimate debtor as an organization/legal person in the form of the BIC or BEI code.
++++++	other	+	Other identification of the ultimate debtor as an organization/legal person.
++++++	identification	Max35Text	Other identification of the ultimate debtor as an organization/legal person in an unstructured format.
++++++	schemeName	+	Type of the code identifying the ultimate debtor as an organization/legal person.
++++++	code	Code	Type of the code identifying the ultimate debtor as an organization/legal person in the form of a code as per the ISO code list.
++++++	proprietary	Max35Text	Type of the code identifying the ultimate debtor as an organization/legal person in a free text format.
++++++	issuer	Max35Text	Issuer of the code identifying the ultimate debtor as an organization/legal person.
+++++	privateIdentification	+	Unique identification of the ultimate debtor as a natural person.
+++++	other	+	Other identification of the ultimate debtor as a natural person.
++++++	identification	Max35Text	Other identification of the ultimate debtor as a natural person in an unstructured format.
++++++	schemeName	+	Type of the code identifying the ultimate debtor as a natural person.
++++++	code	Code	Type of the code identifying the ultimate debtor as a natural person in the form of a code as per the ISO code list.

+++++++	proprietary	Max35Text	Type of the code identifying the ultimate debtor as a natural person in a free text format.
+++++++	issuer	Max35Text	Issuer of the code identifying the ultimate debtor as a natural person.
++++	creditor	+	Information about the creditor. Depending on the direction of the payment, filled in for the counterparty.
+++++	name	Max140Text	Creditor's name.
+++++	postalAddress	+	Creditor's postal address.
+++++	streetName	Max70Text	Street name used in the creditor's postal address.
+++++	buildingNumber	Max16Text	Building number used in the creditor's postal address.
+++++	postCode	Max16Text	Postal code used in the creditor's postal address.
+++++	townName	Max35Text	Town name used in the creditor's postal address.
+++++	country	CountryCode	Country name used in the creditor's postal address.
+++++	addressLine	Max70Text	Creditor's postal address in an unstructured format.
+++++	identification	+	Identification of the creditor.
+++++	organisationIdentification	+	Unique identification of the creditor as an organization/legal person.
+++++	bicOrBei	BICIdentifier	Identification of the creditor as an organization/legal person in the form of the BIC or BEI code.
+++++	other	+	Other identification of the creditor as an organization/legal person.
+++++	identification	Max35Text	Other identification of the creditor as an organization/legal person in an unstructured format.

+++++++	schemeName	+	Type of the code identifying the creditor as an organization/legal person.
+++++++	code	Code	Type of the code identifying the creditor as an organization/legal person in the form of a code as per the ISO code list.
+++++++	proprietary	Max35Text	Type of the code identifying the creditor as an organization/legal person in a free text format.
+++++++	issuer	Max35Text	Issuer of the code identifying the creditor as an organization/legal person.
++++++	privateIdentification	+	Unique identification of the creditor as a natural person.
++++++	other	+	Other identification of the creditor as a natural person.
+++++++	identification	Max35Text	Other identification of the creditor as a natural person in an unstructured format.
+++++++	schemeName	+	Type of the code identifying the creditor as a natural person.
+++++++	code	Code	Type of the code identifying the creditor as a natural person in the form of a code as per the ISO code list.
+++++++	proprietary	Max35Text	Type of the code identifying the creditor as a natural person in a free text format.

++++++	issuer	Max35Text	Issuer of the code identifying the creditor as a natural person.
++++	creditorAccount	+	Information about the creditor's account. Depending on the direction of the payment, filled in for the counterparty.
++++	identification	+	Creditor's account type identification.
+++++	iban	IBAN2007Identifier	Creditor's account number in the IBAN international format.
+++++	other	+	Creditor's account number expressed in other/local format.
+++++	identification	Max34Text	Value of creditor's account number expressed in other/local format.
++++	currency	CurrencyCode	Creditor's account currency.
++++	name	Max70Text	Creditor's account name.
++++	ultimateCreditor	+	Information about the ultimate creditor. Depending on the direction of the payment, filled in for the counterparty.
++++	name	Max140Text	Ultimate creditor's name.
++++	postalAddress	+	Ultimate creditor's postal address.
+++++	streetName	Max70Text	Street name used in the ultimate creditor's postal address.
+++++	buildingNumber	Max16Text	Building number used in the ultimate creditor's postal address.
+++++	postCode	Max16Text	Postal code used in the ultimate creditor's postal address.
+++++	townName	Max35Text	Town name used in the ultimate creditor's postal address.
+++++	country	CountryCode	Country name used in the ultimate creditor's postal address.

++++++	addressLine	Max70Text	Ultimate creditor's postal address in an unstructured format.
+++++	identification	+	Identification of the ultimate creditor.
++++++	organisationIdentification	+	Unique identification of the ultimate creditor as an organization/legal person.
+++++++	bicOrBei	BICIdentifier	Identification of the ultimate creditor as an organization/legal person in the form of the BIC or BEI code.
+++++++	other	+	Other identification of the ultimate creditor as an organization/legal person.
+++++++	identification	Max35Text	Other identification of the ultimate creditor as an organization/legal person in an unstructured format.
+++++++	schemeName	+	Type of the code identifying the ultimate creditor as an organization/legal person.
+++++++	code	Code	Type of the code identifying the ultimate creditor as an organization/legal person in the form of a code as per the ISO code list.
+++++++	proprietary	Max35Text	Type of the code identifying the ultimate creditor as an organization/legal person in a free text format.

+++++++	issuer	Max35Text	Issuer of the code identifying the ultimate creditor as an organization/legal person.
+++++	privateIdentification	+	Unique identification of the ultimate creditor as a natural person.
+++++++	other	+	Other identification of the ultimate creditor as a natural person.
+++++++	identification	Max35Text	Other identification of the ultimate creditor as a natural person in an unstructured format.
+++++++	schemeName	+	Type of the code identifying the ultimate creditor as a natural person.
+++++++	code	Code	Type of the code identifying the ultimate creditor as a natural person in the form of a code as per the ISO code list.
+++++++	proprietary	Max35Text	Type of the code identifying the ultimate creditor as a natural person in a free text format.
+++++++	issuer	Max35Text	Issuer of the code identifying the ultimate creditor as a natural person.
++++	proprietary	+	More detailed specification on the ATM at which the card transaction took place.
+++++	type	Max35Text	Specification of a bank's own / other bank's ATM.
+++++	party	+	Specification of the ATM's name / owner.
++++++	name	Max140Text	Specification of the ATM's name / location.

+++	relatedAgents	+	Information about the payment-related debtor's bank and creditor's bank.
++++	debtorAgent	+	Information about the debtor's bank. Depending on the direction of the payment, filled in for the counterparty.
+++++	financialInstitutionIdentification	+	Debtor's bank code expressed in the BIC / SWIFT international code format Filled-in value (constant): KOMBCZPPXXX.
++++++	bic	BICIdentifier	Debtor's bank BIC / SWIFT code. Filled-in value (constant): KOMBCZPPXXX.
++++++	clearingSystemMemberIdentification	+	Debtor's bank code in a local format, either as a code or a text description.
++++++	clearingSystemIdentification	+	Identification of the debtor's bank in the local payment system in which the debtor's bank operates.
+++++++	code	Code	Identification of the debtor's bank in the local payment system in which the debtor's bank operates, in the form of a payment system code.
+++++++	proprietary	Max35Text	Identification of the debtor's bank in the local payment system in which the debtor's bank operates, in an unstructured format of a descriptive text.
+++++++	memberIdentification	Max35Text	Debtor's bank code in a local format.

+++++	name	Max140Text	Debtor's bank name as given in the SWIFT Directory. Filled-in value (constant): KOMERCNI BANKA A.S.
+++++	postalAddress	+	Debtor's bank postal address.
+++++	streetName	Max70Text	Street name used in the debtor's postal address.
+++++	buildingNumber	Max16Text	Building number used in the debtor's postal address.
+++++	postCode	Max16Text	Postal code used in the debtor's postal address.
+++++	townName	Max35Text	Town name used in the debtor's postal address.
+++++	country	CountryCode	Country name used in the debtor's postal address.
+++++	addressLine	Max70Text	Debtor's postal address in an unstructured format.
+++++	other	+	Other identification of the debtor's bank.
+++++	identification	Max35Text	Other identification of the debtor's bank, in particular in the form of the so-called local bank code.
++++	creditorAgent	+	Information about the creditor's bank. Depending on the direction of the payment, filled in for the counterparty.
++++	financialInstitutionIdentification	+	Creditor's bank code expressed in the BIC / SWIFT international code format. Filled-in value (constant): KOMBCZPPXXX.

+++++	bic	BICIdentifier	Creditor's bank BIC / SWIFT code. Filled-in value (constant): KOMBCZPPXXX.
+++++	clearingSystemMemberIdentification	+	Creditor's bank code in a local format – either as a code or a text description.
+++++	clearingSystemIdentification	+	Identification of the creditor's bank in the local payment system in which the creditor's bank operates.
+++++	code	Code	Identification of the creditor's bank in the local payment system in which the creditor's bank operates, in the form of a payment system code.
+++++	proprietary	Max35Text	Identification of the creditor's bank in the local payment system in which the creditor's bank operates in an unstructured format of a descriptive text.
+++++	memberIdentification	Max35Text	Creditor's bank code in a local format.
+++++	name	Max140Text	Creditor's bank.
+++++	postalAddress	+	Creditor's bank postal address.
+++++	streetName	Max70Text	Street name used in the creditor's postal address.
+++++	buildingNumber	Max16Text	Building number used in the creditor's postal address.
+++++	postCode	Max16Text	Postal code used in the creditor's postal address.
+++++	townName	Max35Text	Town name used in the creditor's postal address.

++++++	country	CountryCode	Country name used in the creditor's postal address.
++++++	addressLine	Max70Text	Creditor's postal address in an unstructured format.
+++++	other	+	Other identification of the creditor's bank, in particular in the form of the so-called local bank code.
++++++	identification	Max35Text	Information about the creditor's bank. Depending on the direction of the payment, filled in for the counterparty.
+++	purpose	+	The purpose of the payment.
++++	code	Code	The purpose of the payment expressed in the form of the code, as stated in the payment instruction.
++++	proprietary	Max35Text	The purpose of the payment expressed as unstructured information, as stated in the payment instruction.
+++	remittanceInformation	+	Additional information about the payment.
++++	unstructured	Max140Text	Additional details concerning the payment filled in as unstructured information. If the unstructured record of the additional details is contained more than once in the payment instruction, only the first occurrence is taken into account.
++++	structured	+	Payment symbols are expected here (a variable, specific, and constant symbol). If a variable, specific or constant symbol is not filled-in in the payment instruction, the entire Structured Remittance Information structure remains blank.
+++++	creditorReferenceInformation	+	Displaying information about the variable, specific, and constant symbol.

+++++	reference	Max35Text	The field should start with 'VS:' (for the variable symbol), or 'SS:' (for the specific symbol), or 'KS:' (for the constant symbol). One repetition of the 'reference' field may contain all 3 symbols, but each of them can occur there only once. The symbols are filled in as follows - VS: KS: SS:)[0-9]{1,10}.
+++	additionalTransactionInformation	Max500Text	Additional information provided by the bank; filled in only if they are contained in the record. Additional information concerning SEPA DD is given here (e.g. Creditor Identifier, Payment scheme, SEPA DD sequence etc.)

JSON – example of a response:

```
{
  "pageNumber": 5,
  "pageCount": 7,
  "pageSize": 100,
  "nextPage": 6,
  "transactions": [
    {
      "amount": {
        "amount": 15241.3,
        "currency": "EUR"
      },
      "creditDebitIndicator": "CRDT",
      "bookingDate": {
        "date": "2017-04-24T05:00:00.000Z"
      },
      "valueDate": {
        "date": "2017-04-24T05:00:00.000Z"
      },
      "bankTransactionCode": {
        "proprietary": {
          "code": 1000010,
          "issuer": "CBA"
        }
      },
      "entryDetails": {
        "transactionDetails": {
          "references": {
            "endToEndIdentification": "VS0250117002/SS0000000000/KS0000",
            "mandateIdentification": ""
          },
          "instructedAmount": {
            "amount": 15241.3,
            "currency": "EUR"
          },
          "counterValueAmount": {
            "currency": "EUR",
```

```
"amount": 15241.3
},
"currencyExchange": {
  "sourceCurrency": "EUR",
  "exchangeRate": "27.06"
},
"relatedParties": {
  "debtor": {
    "name": "Jan Novák",
    "identification": {
      "privateIdentification": {
        "other": {
          "identification": "000000-2108589434/2700",
          "schemeName": {
            "code": "CUST",
            "proprietary": "123456789"
          }
        }
      }
    },
    "organisationIdentification": {
      "other": {
        "identification": "000000-2108589434/2700",
        "schemeName": {
          "code": "TXID",
          "proprietary": "CZ1234567890"
        }
      }
    }
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ9501000000001234567899",
      "other": {
        "identification": "1234567899"
      }
    }
  },
  "creditor": {
    "name": "Jan Novák",
    "identification": {
      "privateIdentification": {
        "other": {
          "identification": "000000-2108589434/2700",
          "schemeName": {
            "code": "CUST",
            "proprietary": "123456789"
          }
        }
      }
    },
    "organisationIdentification": {
      "other": {
        "identification": "000000-2108589434/2700",
        "schemeName": {
          "code": "TXID",
          "proprietary": "CZ1234567890"
        }
      }
    }
  },
  "creditorAccount": {
    "identification": {
      "iban": "CZ9501000000001234567899",
      "other": {
        "identification": "1234567899"
      }
    }
  }
},
"relatedAgents": {
```

```

    "debtorAgent": {
      "financialInstitutionIdentification": {
        "bic": "KOMBCZPPXXX",
        "clearingSystemMemberIdentification": {
          "clearingSystemIdentification": {
            "code": "0100",
            "proprietary": "Komerční banka a.s."
          },
          "memberIdentification": "0100"
        }
      }
    },
    "creditorAgent": {
      "financialInstitutionIdentification": {
        "bic": "KOMBCZPPXXX",
        "clearingSystemMemberIdentification": {
          "clearingSystemIdentification": {
            "code": "0100",
            "proprietary": "Komerční banka a.s."
          },
          "memberIdentification": "0100"
        }
      }
    },
    "purpose": {
      "code": "0001",
      "proprietary": "PLATBA ZA SLUŽBY"
    },
    "remittanceInformation": {
      "structured": {
        "creditorReferenceInformation": {
          "reference": [
            "VS:0250117002"
          ]
        }
      }
    },
    "debtorNote": "VS0250117002/SS0000000000/KS0000SEPA převod",
    "creditorNote": "zaloha dle smlouvy o dodavkách c. 45678/2017",
    "description": "8201701069595 BIC: GIBACZPPXXX; #71A# SHA ZALOHA DLE SMLOUVY
O DODAVKACH"
  }
}
]
}

```

Table 17 CBA-standard defined error codes for the GET service – List of Client's Payment Accounts

HTTP CODE	STATUS	ERROR CODE	PURPOSE
401		UNAUTHORISED	Invalid/missing access token = the user has not been authenticated
401		UNAUTHORISED	Invalid/ missing certificate = the provider has not been authenticated
404		PAGE_NOT_FOUND	The query concerns a non-existent page
400		PARAMETER_INVALID	The parameter value is not valid
404		ID_NOT_FOUND	The query concerns a non-existent page
400		AC09	[InvalidAccountCurrency] – in case of multicurrency accounts, or the exchange list shows an unsupported currency
400		DT01	[InvalidDate] Invalid date

403	FORBIDDEN	Invalid certificate / token
400, 50x		General reason for rejection – will be specified
400	AG01	[TransactionForbidden] – account does not have a consent
400	AG02	[InvalidDebtorAccountNumber] – account validation

12. PSD2 glossary – selected terms

API – Application Programming Interface

REST – (Representational State Transfer) is an API architecture, which allows for accessing the data and execute CRUD operations. It usually uses the HTTP/HTTPS protocol. REST is stateless, which makes communication with the API much easier and allows for the parallel processing of its contents. At the same time, it can be easily used with HTTP, which is a widely used protocol. Last not least, it provides a standard of a kind so we can easily use an API created by somebody else or offer our API to a number of other users. The REST interface supports uniform and easy access to resources. Such resources can be data or application states (as long as they can be described using specific data). All resources have their URI identifier. REST defines four basic methods of access (GET, PUT, POST, and DELETE). The HTTP Verbs have the following meaning:

- GET – obtaining the data
- POST – creating
- PUT – editing (like SET, it edits an entire resource)
- DELETE – deleting
- PATCH – partial editing

REST API – Distributed environment interface oriented on data, not on calling procedures like RPC (XML-RPC) or SOAP. Web services define remote procedures and their calling protocol; REST decides how the data should be approached. REST API uses HTTP methods, such as @GET, @PUT, @POST, @DELETE, @PATCH.

TPP – Third Party Provider (a third party registered/licenced by the CNB)

AIS – Account Information Service

AISP – Account Information Service Provider

PIS – Payment Initiation Service

PISP – Payment Initiation Service Provider

CIS – Card-based Payment Instrument Issuance Service

CISP – Card-based Payment Instrument Issuance Service Provider

ASPSP – Account Servicing Payment Service Provider (a bank holding the debtor's payment account)