ČESKÁ BANKOVNÍ ASOCIACE

Banking Activities Standards

Standard ČBA n. 38

Czech Standard for Open Banking

Changes

Date	Note	Author
15.11.2017	Basic document v.01	ČS Petr Michalík

Table of Contents

Introduction

On 16 November 2015, the European Banking Authority issued the Revised Payment Service Directive (hereinafter: PSD2). As of 13th January 2018, PSD2 will replace former Directive on Payment Services in the Internal Market. A need for this substation was caused by a rapid development of IT, by the birth of brand new types of payment services and by a number of overall changes. Thanks to PSD2 new subjects will be regulated in the field of electronic payments, subjects, which were not in the scope of the previous Directive. Among other things, new legislation opens banking for the so-called third parties.

PSD 2 should be accompanied by number of norms: on one hand certain regulatory technical standards (RTS) have to be adopted, on the other hand guidelines describing rules, which have to be followed in specific situations have to be issued by the EBA (European Banking Authority) too. Unfortunately, during preparation of these documents both the EBA and the European Commission caused some delays, therefore it was necessary to work only with drafts of these papers. Especially missing RTS on Strong Authentication and Secure Communication (RTS on SCA) caused troubles to all stakeholders.

The PSD 2 was implemented into the Czech legislation by the brand new No 370/2018 Coll., Act on Payment Systems, which will come into force on 13th January 2018. Suspended are only those paragraphs of that Act, which refer to the RTS on SCA.

In order to maintain equal competition between all stakeholders, the EBA seeks to maintain neutrality and defines basic rules of functioning in PSD2 at a general level. Despite the specification of open banking rules to third parties, the implementation across particular banks may vary. This versatility implies a more complex, and more expensive integration of third parties to become payment service providers keeping accounts. In order to support the European PSD2 directive and its proper implementation, the banks in the Czech market have decided to standardize their solutions.

The aim of the Czech Standard for Open Banking is to lay down rules for communication, mainly for services defined by the PSD2: Account Information Service Provider (AISP), Payment Initiation Service Provider (PISP), Card Issuing Service Provider (PISP)

The standard is designed and developed to maintain a high degree of versatility. The standard is voluntary, and it is up to each bank to consider joining the standard. With respect to the fact that systems and their operations differ across all payment service providers, standard participants may deviate from the standard at certain points, within the scope of their specific information systems. Since each payment service provider is obliged to document its solution according to PSD2, a large number of fields for optional information is defined in the Czech Open Banking Standard.

The main benefits of the Czech Standard are the easier integration of TPP into systems which banks are offering, including without any other intermediaries, ensuring a uniform interpretation of the PSD2 in the Czech market, compliance with the content of the transmitted data and security elements in communication with banks, support for the unified functioning of services for clients across banks, timely readiness for the implementation of PSD2.

The Czech Standard for Open Banking consists of these parts:

- Introduction and common features
    - Technical description
    - Security standard
- Definition of API Account Information
- Definition of API Payment Initiation
- Definition of API Balance Check
- Examples of requests and response of particular API resources in JSON format

1 COBS Content

1.1 A General Description of Standard Content

This standard includes a definition of the individual areas of the interface for third party access to bank information and services. In terms of the definition of this interface in the required detail and with a certain degree of freedom, the individual descriptions break down into the specification of the technical interface, the security interfaces / principles and the data content of the individual services.

The technical description is intended to specify the basic parameters of the communication and justifies their use.

The Security standard primarily defines the principle of obtaining a user access authorization (bank client) to individual bank services for a third party.

Data content specifies an overview of the elements of individual services and the principles of their use.

1.2 Technical Description

1.2.1 Transport protocol

The HTTP 1.1 protocol is used as the transport communication protocol for this API according to [RFC7230, RFC7231, RFC7232, RFC7233, RFC7234, RFC7235] or the HTTP 2.0 protocol according to [RFC7540].

1.2.2 Interface architecture

The **REST** (Representational State Transfer) is used to design and implement the API communication interface. The use of hypermedia links is optional and is not part of the API specification (suggested interfaces using hypermedia links do not work).

1.2.3 Data entry format

JSON (JavaScript Object Notation) is used as a format for the (request) data entry and (response) API. Undefined or unused elements are returned from the API with a null value, and thus they are not hidden. Should it be required to define an element as blank or inaccessible, it is possible to return the element with an empty string value. For blank fields, it is required to return an empty collection ([]) to the place of the null value.

1.2.4 Type of transmission media

Due to the use of the JSON data transfer format, this transmission character is defined by the relevant MIME/Content-type. For this API, the use of **application/json** as the content type is required, except for the said exceptions. The default coding of transfer content is UTF-8.

An example of the parameter in the request header:

```
Content-Type: application/json
```

1.2.5 Coding

The coding of the transfer character set is UTF-8 by default. A change in the desired coding of the character set can be done with the API, which enables this by using the HTTP request header **Contet-Type** with the parameter **charset**.

An example of the change of coding for the parameter Content-Type:

```
Content-Type: application/json; charset=UTF-16
```

1.2.6 Localization

Unless otherwise defined for specific resources, it is possible to use, for the setting of response localization, the parameters of request and response headers **Accept-Language** a **Content-Language**.

Accept-Language is a request parameter and allow defining a set of preferred localizations arranged according to preferences.

Content-Language is a parameter of the response header and defines the localization used in the response to a request.

An example of the request with preferred localization:

```
Accept-Language: en,en-US,fr;q=0.6
```

An example of the response for the required localization:

```
Content-Language: en
```

1.2.7 HTTP methods used in API and their specifications

METHOD	INDEMPOTENT	CHANGES STATUS	PURPOSE
GET	Yes	No	Select: read-only calling (e.g., a list of client accounts)
POST	No	Yes	Cretae: Creates a new facility / event (e.g. creating a new transaction), calling the function

PUT	Yes	Yes	Update: Change of the existing facility (e.g. change of the transaction parameter)
DELETE	Yes	Yes	Delete: Deleting the existing facility (e.g. deleting a message)

GET, PUT and DELETE calls are idempotent. It does not matter what the server status is like, if the call is used only once or more (with the same parameters / content). POST calling, which usually creates new records may create duplicate data on a recalling.

1.2.8 Use of filtering, sorting and pagination of API elements

This section specifies the principles of filtering, sorting, and pagination of API content. These principles apply only to selected API resources. Each API resource in its description specifies whether any of these properties is available within resource usage.

In the case that the use of pagination or parameter filtering results in an empty element set, this empty set is also passed as a valid API call response (e.g., in comparison with the response status 404).

1.2.8.1 Filtering (fromDate, toDate, …)

Specific parameters are defined for filtering for each API resource that allows filtering. These parameters define the range or scope of filtering and are explicitly defined for each such resource.

For example, for filtering a specific time interval, the query parameters **fromDate** and **toDate** can be introduced for the API resource.

Example of using parameters for filtering:

```
GET /transactions?fromDate=2016-05-12T05:37:30+02:00&toDate=2016-05-13T10:00:30+02:00


// It returns the list of transactions for the period defined from date and time stated in the parameter fromDate up to the date and time stated in the parameter toDate inclusive
```

1.2.8.2 Sorting on the part of API server (sort, order)

Specified API resources allow requesting data in a sorted order. The sorting property is explicitly listed in the description of each API resource. The API-specified sorting allows selecting an ascending or descending process for several fields at once with the option of defining the sorting direction for each field separately.

At the same time, the mechanism allows to define the ranking order by individual fields.

The following parameters are introduced for sorting

PARAMETR	MANDATORY	DEFAULT VALUE	PURPOSE
sort	Yes	-	It defines a list of fields for sorting. The individual fields in the list are separated by a comma
order	No	**asc**	It defines the sorting direction for each field listed in the sort parameter. Possible values are: - **asc** for ascending sorting - **desc** for descending sorting The individual sorting directions are separated by commas precisely as the field names in the sort parameter and are listed in the same order as the fields listed in the **sort** parameter.

An example of a multiple-order sorting request where the type field does not have the specified direction and will be sorted by default in ascending order:

```
GET /accounts?sort=createDate,type,accountNumber&order=desc,,desc
```

1.2.8.3 Selection of specific fields (fields)

The API resources for which this is explicitly specified allow requesting only the specific elements in the request response. Such a request applies if the client does not require all fields but only the selected elements from API.

The required elements can be specified as a comma-separated list into the parameter **fields**.

An example, when the client requires from API in the response to return only the defined element (accountNumber, createDate, and type):

```
GET /accounts?fields=accountNumber,createDate,type
```

1.2.8.4 Pagination (page, size)

For specific API resources that return sets (such as a transaction field), it is possible to require a paged list. For this query, query parameters **page** and **size** are used. Each resource that allows requiring a paged list has this feature specified explicitly.

Parameters of query for page

PARAMETR	MANDATORY	DEFAULT	PURPOSE

		VALUE	
page	No	0	It defines the required page number. Pages are numbered from 0. If the parameter is not specified, the API returns the first (zero) page.
size	No	**If not specified, API will return the entire set**	It defines the required number of records on the page. If the parameter is not specified, the API returns the entire set.

Resources that allow paging contain the specific structure in which the set of entries is presented. The structure always contains information about the data page such as the number of the next page, the number of pages, the number of entries on the current page, and possibly the total number of entries across all pages (optional data).

Parameters of response pagination

PARAMETR	MANDATORY	PURPOSE
pageNumber	Yes	Number of current page
pageCount	Yes	Total number of query pages
nextPage	No	The number of the next page. If the current page is at the same time the last page of the list, then this parameter is not listed or is given with a null value.
pageSize	Yes	Number of entries per page. This parameter can match the required **size** value from the query, except for the cases when it is the last page, or when the requested page range exceeded the maximum limit defined for the particular API resource.
totalCount	No	Optional data about the total number of entries for all pages. If this value cannot be precisely determined, it is not stated.
page/items/...	Yes	It is an element of field type that contains a set of required entries. The name of this element should match the contents of the set.

An example of query on paged API:

```
GET /transactions?size=25&page=1
```

This query returns the second page of records from the 26th record and further if the second page exists, or if there are more than 25 records.

An example of paged response:

```
{
    "pageNumber": 1,
    "pageCount": 12,
    "nextPage": 2,
    "pageSize": 25,
    "totalCount": 298,
    "page": [
        { ...item... },
        { ...item... },
        { ...item... }
    ]
}
```

1.2.9 Versioning of API

Should it be necessary to issue a new version of the API at the same time as the existing one, this information is part of the URI for the individual API resources. For each API, the API version is stated as a simple number from the sequence of versions, each time with the prefix "v".

An example of version provided in URI of API resource:

```
GET /api/v1/
```

The API definition in this specification does not include versioning within defined URI resources.

1.2.10 HTTP status codes used and addressing errors in API

To communicate and manage exceptions, these APIs use uniformly established HTTP status codes and defined error content.

For each API and its resources, a list of possible error statuses with their relationship to the appropriate HTTP status code, is given.

List of HTTP status codes used in API

STATUS CODE	MEANING	PURPOSE
200	OK	Correct response with content
201	OK	Correct response = new entry created
204	OK	Correct response = entry was deleted

304	Not Modified	Resource without change = possible to use cache data
400	Bad Request	This is an invalid query that cannot be answered. E.g., if JSON content is not valid for this resource.
401	Unauthorized	Querying requires user authentication
403	Forbidden	Access to the requested resource is not granted or is not possible for that user.
404	Not Found	The requested object / page does not exist or was not found
415	Unsupported media type	Request contains a request for unsupported transmission type (e.g. in relation to headers Accept and Accept-Language)
422	Unprocessable Entity	This error can be used if the requested object cannot be processed, or the required query parameter is absent.
500	Internal Server Error	Server error that may be triggered by technical problems or in case of an untreated error status.
501	Not implemented	It can be used if the server does not support the required operation.

1.2.10.1 Defined error content

For all error statuses addressed, specific error content is returned in JSON format. The content is defined by mandatory and optional parameters that the API client side can evaluate and use to properly route the response of the application or operation.

Since multiple error statuses can occur at the same time (e.g., multiple invalid elements when initiating a payment), the error always contains a set of errors defined by individual elements.

Parameters of error response

PARAMETR	TYPE	MANDATORY	PURPOSE
errors	Array	Yes	The superior element contains a set of all error statuses
errors.error	Text	Yes	It contains a specific error code
errors.parameters	Array	No	The field of additional elements specific for the given error code. These parameters are always specified in the description of a specific error code.
errors.scope	Text	No	It determines the JSON path of the request element that caused the error status.
errors.message	Text	No	Optional text description. It is not intended for interpretation to the end user but, for example, to enrich the error log.

An example of response with the set of errors:

```
HTTP/1.1 400 Bad Request
```

```
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "errors": [
    {
       "error":"SOME_ERROR_CODE",
       "parameters": null,
       "message": null
    },{
       "error":"ANOTHER_ERROR_CODE",
       "scope":"account.amount.currency",
       "parameters": null,
       "message": null
    },{
       "error":"OTHER_ERROR_CODE",
       "parameters": {
           "AMOUNT_ENTERED": 10000,
           "CURRENCY": "EUR",
           "LIMIT": 500,
         }
       "scope":"orders[3].amount.value",
       "message":"Requested amount is too large"
    }
  ],
 }
```

The error response status corresponds to the status defined for specific error codes or, for each error code defined in the API resource documentation, the HTTP response status that this error causes is also stated.

1.2.11 Authentication and Request Authorization (OAuth2)

API defined by this specification is not responsible for verifying user credentials (that the person for whom API is accessed is the one for which it impersonates, e.g. by checking the user's name and password), but is responsible for verifying that the person for whom the API is accessed has the right to obtain such data.

Therefore, the request authorization is based on the OAuth2 authorization flow concept secured with a token, and only has to check the validity of the tokens that the client provides for each call as evidence that it can access the requested data.

Within these APIs, the authorization token is considered to be a short-term and non-constitutive element that must be used in every API call that requires the request authorization.

The API provider interface uses a token to verify the client's communication privileges on a particular resource. Verification of the token is performed with an API service mechanism/provider and only upon this verification, the business logic linked to the required data source, is executed.

This API interface may work with any type of token if the following assumptions are valid:

- Token has the format / coding that may be transferred using HTTP of the field header "Authorization".
- Token the length of maximum 1024 bytes.
- It is possible (with the API Provider service) to verify the validity of the token.
- It is possible (with the API Provider service) to derive the user from the token.
- It is possible (with the API Provider service) to derive the authorization of the user from the token. This includes a separate authorization to read and write up to the level of individual business services/objects (e.g., one particular account, message). However, the Token should alternatively refer to privileges within a larger range of API Provider services, in particular entities owned by the user, e.g., "All own accounts".

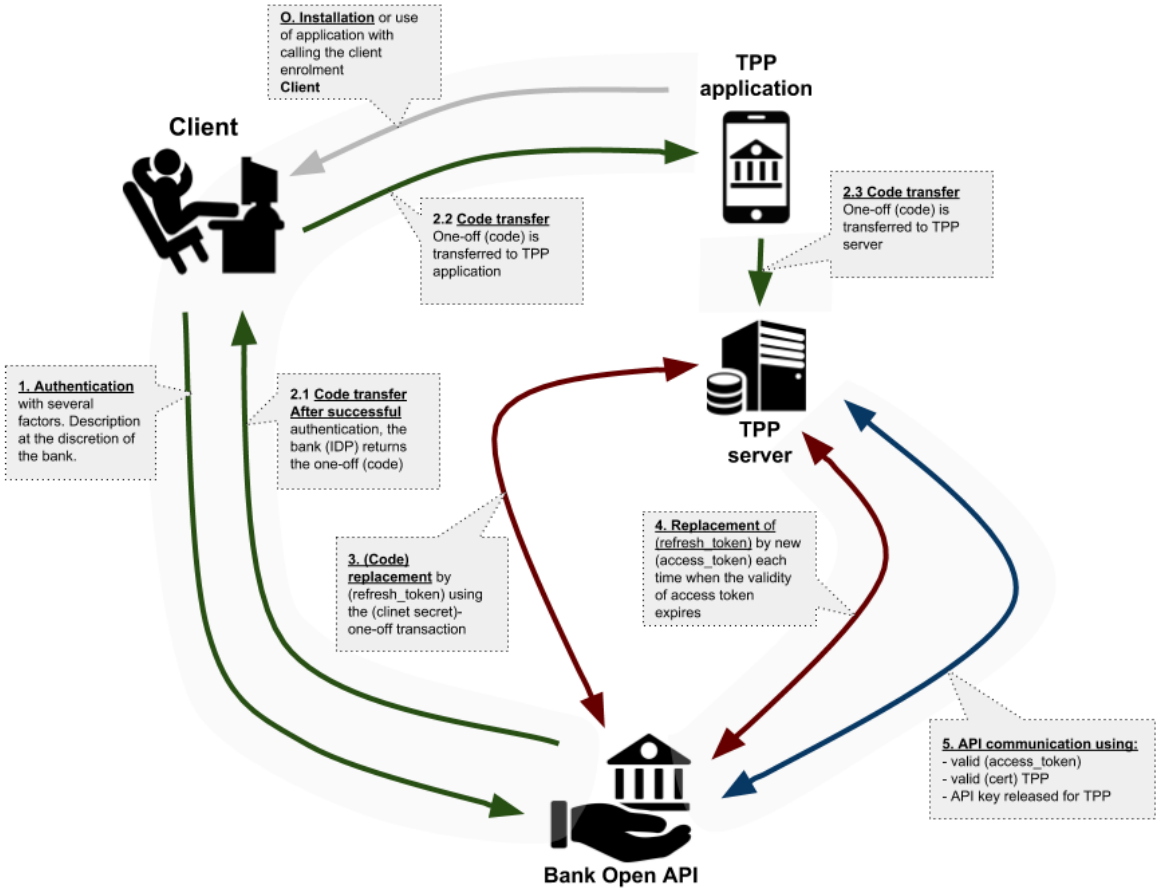
An example of using the token in the request header:

```
Content-Type: application/json
Authorization: Bearer dd3c4d08416c39af411ac0a37d110bea365fc1e
```

1.2.12 Security standard

The description of the client enrolment principle or of client product(s) at the bank to a third-party application. This principle is considered within this Standard. The described flow begins at the point where the client already has and in some way uses a third-party application.

1.3 Flow in the Process of Client Enrolment to the TPP Application



Flow enrolment chart

1.3.1 Individual flow steps

0. Installation - The client installs or accesses a third-party application and starts to use it. The Bank does not enter this process. During the use of the application, a third party allows the user, as a client of the Bank, to implement his own product to the application that the client has at the Bank. The introduction of a third-party product and service corresponds to the role under which the third party is licensed (AISP, PISP, CISP). The application starts the client authentication process by redirecting it to the Bank's authentication point (the identity provider).

1. **Authentication** – The client, through the authentication process ensured by the Bank, goes through the authentication in the SCA principle. The authentication may be, for instance, federated and is in the regime of each bank with a character similar to the one known to the client, e.g. from Internet banking.

2. **Code transfer** – The result of successful client authentication is a generated one-time *code* which is transferred to the third-party application in the response when routed from par. 0., see 2.1 - 2.3.

3. **Refresh_token acquisition** - One-time replacement of the *code* by the *refreshe_token*. For the replacement, the third-party must know the *client_secret* as a joint "secret" between the third party and the Bank (IDP of the bank). The *refresh_token* is valid for the relevant client and the specific third party within their role.

4. **Access_token acquisition** – Since to communicate with API, it is necessary to have a valid *access_token* for its acquisition it is necessary to execute the replacement of *refresh_token* by a valid *access_token* with a relatively short expiry period (e.g., 3 600s).

5. **Communication with API** - The API communication with the valid access_token and third-party certificate. Since the *access_token* is issued in the context of a specific user, and the third-party application is the request authorization at API executed with respect to the contract. At the same time, the validity of the third-party certificate and its role is verified.

The solution is the inspiration of providing authentication as a service with client impersonification in the sense of OpenID - http://openid.net/ standard. It is an open standard enabling the decentralization of user authentication. In most cases, it is used as a substitute for the user authentication solution, with authentication and, in some cases, authorization executed by the operator of the OpenID service.

To generalize the solution, the basis for this proposal is only to use the OAuth2 open protocol to issue authorization tokens. Then primarily, the use of **code grant** authorization framework. See Chapter 1.4.1.1 OAuth2 Code Grant.

TERM/ACRONYM	DESCRIPTION
TPP	*Third party provider (AISP, PISP, CISP)*
SCA	*Strong Customer Authentication*
code	*One-time token/code usable only for a specific client and TPP in replacement by a long-term refresh_token*
refresh_token	*A long-term token issued based on replacement by the one-time code.*
access_token	*A short-term (in some cases, one-time) token that may be generated again using a refresh_token. The token serves for the authorization of request at API.*
IDP	*Identity provider*
STS	*Secure Token Service*

1.3.1.1 OAuth2 Code Grant

Within the OAuth2 protocol, in the case of **code grant** authorization framework, it is a method how `refresh token` and `access token` may be released to the partner application as a result of the user identification and authentication.

The partner application uses the short-term `access token` to communicate with the API of the bank and after its expiry, the `refresh token` may be used to demand a new `access token`.

1.3.1.1.1 Basic properties

- `The access token` is issued as short-term (e.g., 3 600s) and may be revoked (by the user, application and the provider/Bank)
- `The access token` is issued for a specific application and a specific user it may not be used successfully for another application
- It is not possible to use the `refresh token` directly for communication with API, it has a long or unlimited validity (e.g., in the case of PSD2, 90 days) and the provider may have the possibility to revoke it and the provider may allow the possibility to the user
- The provider (the Bank) and the application (TPP) share a joint "secret" - `client secret`
- The result of the user identification and authentication is a code that may be replaced by the third-party application by the `refresh token` and `access token` using the client secret
- It is not possible to use the `code` without knowing the `client secret`

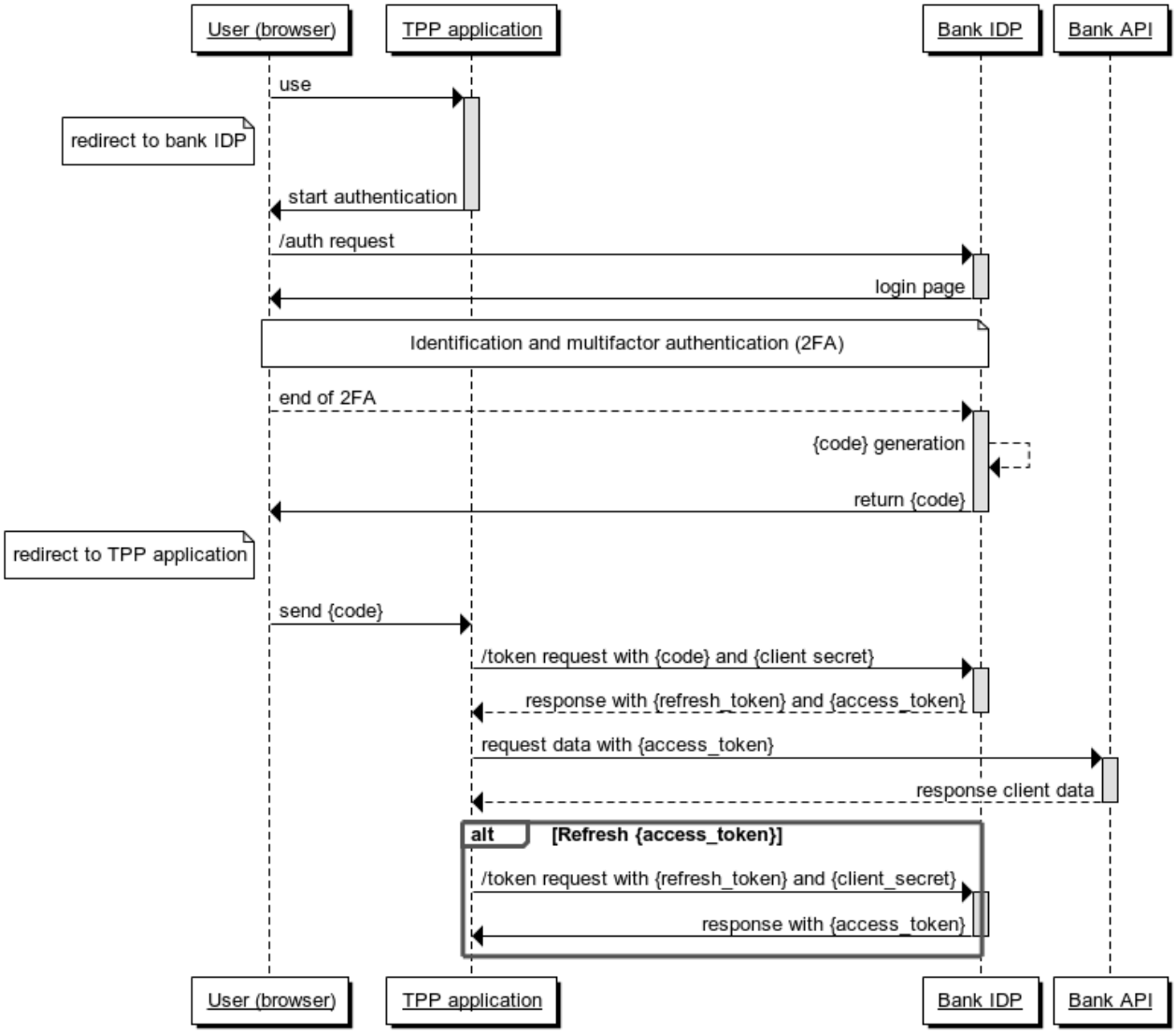
1.3.1.1.2 Description of code grant flow

Flow initiation:

- The TPP application has its own unique `client_id` assigned from the provider and knows _id and `client secret` for the user
- On the issue of `client_id` and `client_secret`, the provider obtains the information on `redirect uri` – i.e. on URL where the user should be directed after successful authentication

Individual steps of code grant flow:

1. The TPP application directs the user to `/auth` resource of the provider (bank) to execute the identification and authentication
2. The client identification and authentication run - the steps are fully at the discretion of the provider
3. After successful authentication, the provider issues a `code`, and directs the client to the application URI (`redirect_uri`) with it
4. The TPP application will use the resource `/token` to obtain the `refres_token` and the `access_token`. When calling the resource, the `code` and `client_secret` are delivered to the provider
5. The TPP application uses the `access_token` to communicate with the provider's API
6. The Provider will verify the `access_token` internally, e.g. using the internal STS (secure token service). On the verification, they acquire the user identity based on authentication of which the `access token` was issued



Described code grant OAuth2 flow

1.3.1.1.3 Solution preconditions

The Bank ensures

- Issuing the resource for dynamic registration
- Issued authorization resource (**resource 1**.)
- Issued resource for the issue of refresh_token and access_token (**resources 2.a, 2.b**)
- Issued resource for the revocation (revoke) of access_token (**resource 3**.)
- Issuing the client_id and client_secret to a third party for its application
- For issued client_id keep the redirect_uri in the third-party application

TPP ensures

- Starting the authorization flow in bank URI
- Replacement of the code by refresh_token and access_token using the resource of the bank (**resource 2.a**)

- Replacement of the refresh_token by the access_token using the resource of the bank (**resource 2.b**)
- Secure keeping of client_secret, refresh_token, access_token and the code

1.4 An Overview of Resources to Ensure Security Flow Standard

1.4.1 Optional registration resource issued by the Bank

1.4.1.1 0. Initialization / registration resource

By calling the resource, the TPP may demand the dynamic registration of client_id. It is necessary to use the valid certificate to call the resource. The output is the parameters client_id and client_secret that TPP needs to start and pass through the authentication process of the user (the client of the bank). And also the API_Key which is the carrier of the application configuration in calling the bank API.

Endpoint: POST https://idp.banka.cz**/oauth2/register**

Request content:

PARAMETR	VALUES	MANDATORY	DESCRIPTION
application_type	web, native	y	The type of application that will use the client_id. In the case of the web type, the definition of rediect_uris is required in the format of the web uri in the form http/s scheme. For the native type, in redirect_uris it is possible to enter, e.g. the application package, or the own format.
redirect_uris	Field containing the strings, e.g. in the URL format [Max 3x 2047 B]	y	Enumeration of URL where the authentication flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
client_name	string [Max 255 B]	y	Name of the client application
client_name#en-US	string [Max 1024 B]	n	Name of the client application in the relevant language/coding.
logo_uri	URI [Max 2047 B]	n	URI of the application logo (or the place from where it may be downloaded on the registration)
contact	string e-mail [Max 320 B]	n	E-mail as a contact to the responsible person on the part of the client application.
scopes	Field of strings [Max 10x 255 B]	n	The field of applications of required scopes. On the registration, scopes are validated against the content of the certificate used.

Response content (only new fields are stated against the request):

PARAMETR	VALUES	MANDATORY	DESCRIPTION
client_id	string	y	The client_id assigned by the application. This ID starts the authentication process and communication is decoded in the replacement of code and refresh_token.
client_secret	string	y	Client secret - password/token issued by the bank IDP for the TPP application (client_id)
client_secret_expires_at	Time	n	Default value is 0 (client_id never expires). Otherwise, the value is stated in seconds from data 1970-01-01T0:0:0Z
api_key	string	y	API key that the application uses to communicate with the bank API. If the bank API does not support API keys, the value „NOT_PROVIDED“ will be returned.

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	unauthorized_client	The client is not unauthorized to execute the request.
401	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.
400	invalid_scope	Invalid request scope.
403	insufficient_scope	For instance, and insufficient authorization to use the required scope.
400	invalid_redirect_uri	The value of one or more redirect uri is not valid.

A request example:

```
POST /oauth2/register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz
```

```
{
   "application_type": "web",
   "redirect_uris":
     ["https://www.mymultibank.cz/start",
      "https://www.mymultibank.cz/start2"],
   "client_name": "Moje univerzální banka",
   "client_name#en-US": "My cool bank",
   "logo_uri": "https://www.mybank.cz/logo.png",
   "contact": "info@mybank.cz",
   "scopes": ["aisp", "pisp"]
}
```

A response example:

```
HTTP/1.1 201 Created
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
   "client_id": "a0b25291f0",
   "client_secret":
     "AAjkk45sd78ad454gddd8712_4555g5g5g5gg",
   "client_secret_expires_at": 0,
   "api_key":
     "00000000-1212-0f0f-a0a0-123456789abc",
   "application_type": "web",
   "redirect_uris":
     ["https://www.mymultibank.cz/start",
      "https://www.mymultibank.cz/start2"],
   "client_name": "Moje univerzální banka",
   "client_name#en-US": "My cool bank",
   "logo_uri": "https://www.mybank.cz/logo.png",
   "contact": "info@mybank.cz",
   "scopes": ["aisp", "pisp"]
}
```

1.4.1.2 0.1 Information on application registration data

By calling this resource, the TPP can request an overview of application-specific registration data. To call a resource, a valid certificate and client_id that is issued to this TPP, must be used. The output is an overview of the registration data

Endpoint: GET https://idp.banka.cz**/oauth2/register/{client_id}**

Response content:

PARAMETR	VALUES	MANDATORY	DESCRIPTION
client_id	string	y	The client_id assigned by the application. This ID starts the authentication process and communication is decoded in the replacement of code and refresh_token.
client_secret	string	y	Client secret - password/token issued by the bank IDP for the TPP application (client_id)
client_secret_expires_at	Time	n	Default value is 0 (client_id never expires). Otherwise, the value is stated in seconds from data 1970-01-01T0:0:0Z
api_key	string	y	API key that the application uses to communicate with the bank API. If the bank does not support API keys, the value is returned to the value „NOT_PROVIDED“
application_type	web, native	y	Type of application that will use the client_id. In the case of the web type, the definition of rediect_uris is required in the format of the web uri in the form http/s scheme. For the native type, in redirect_uris it is possible to enter, e.g. the application package, or the own format.
redirect_uris	Field containing the strings, e.g. in the URL format	y	Enumeration of URL where the authentication flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
client_name	string	y	Name of the client application
client_name#en-US	string	n	Name of the client application in the relevant language/coding.
logo_uri	URI	n	URI of the application logo (or the place from where it may be downloaded on the registration)

contact	string e-mail	n	E-mail as a contact to the responsible person on the part of the client application.
scopes	String field	n	The field of applications of required scopes. On the registration, scopes are validated against the content of the certificate used.

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	invalid_client	Invalid client_id.
401	unauthorized_client	The client is not unauthorized to execute the request.
401	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.

A request example:

```
GET /oauth2/register/a0b25291f0 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz
```

A response example:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
   "client_id": "a0b25291f0",
   "client_secret":
     "AAjkk45sd78ad454gddd8712_4555g5g5g5gg",
   "client_secret_expires_at": 0,
   "api_key":
     "00000000-1212-0f0f-a0a0-123456789abc",
   "application_type": "web",
```

```
   "redirect_uris":
     ["https://www.mymultibank.cz/start",
      "https://www.mymultibank.cz/start2"],
   "client_name": "Moje univerzální banka",
   "client_name#en-US": "My cool bank",
   "logo_uri": "https://www.mybank.cz/logo.png",
   "contact": "info@mybank.cz",
   "scopes": ["aisp", "pisp"]
}
```

1.4.1.3 0.2 Change of registration data

By calling this resource, the TPP can request a change of application-specific registration data. To call a resource, you need to use a valid certificate and client_id that is issued to this TPP. The output is an overview of the changed data

Endpoint: PUT https://idp.banka.cz**/oauth2/register/{client_id}**

Request content:

PARAMETR	VALUES	MANDATORY	DESCRIPTION
application_type	web, native	y	The type of application that will use the client_id. In the case of the web type, the definition of rediect_uris is required in the format of the web uri in the form http/s scheme. For the native type, in redirect_uris it is possible to enter, e.g. the application package, or the own format.
redirect_uris	Field containing the strings, e.g. in the URL format [Max 3x 2047 B]	y	Enumeration of URL where the authentication flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
client_name	string [Max 255 B]	y	Name of the client application
client_name#en-US	string [Max 1024 B]	n	Name of the client application in the relevant language/coding.
logo_uri	URI [Max 2047 B]	n	URI of the application logo (or the place from where it may be downloaded on the registration)
contact	string e-mail [Max 320 B]	n	E-mail as a contact to the responsible person on the part of the client application.

scopes	String field [Max 10x 255 B]	n	The field of applications of required scopes. On the registration, scopes are validated against the content of the certificate used.

Response content:

PARAMETR	VALUES	MANDATORY	DESCRIPTION
client_id	ID of TPP application	y	Unique identifier of the TPP application issued by the bank or the bank IDP. E.g., using the resource „0. Initialization/registration resource“
application_type	web, native	y	The type of application that will use the client_id. In the case of the web type, the definition of rediect_uris is required in the format of the web uri in the form http/s scheme. For the native type, in redirect_uris it is possible to enter, e.g. the application package, or the own format.
redirect_uris	Field containing the strings, e.g. in the URL format	y	Enumeration of URL where the authentication flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
client_name	string	y	Name of the client application
client_name#en-US	Any string	n	Name of the client application in the relevant language/coding.
logo_uri	URI	n	URI of the application logo (or the place from where it may be downloaded on the registration)
contact	string e-mail	n	E-mail as a contact to the responsible person on the part of the client application.
scopes	Field of strings	n	The field of applications of required scopes. On the registration, scopes are validated against the content of the certificate used.

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	invalid_client	Invalid client_id.
401	unauthorized_client	The client is not unauthorized to execute the request.

401	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.
400	invalid_scope	Invalid request scope.
403	insufficient_scope	For instance, and insufficient authorization to use the required scope.
400	invalid_redirect_uri	The value of one or more redirect uri is not valid.

A request example:

```
POST /oauth2/register/a0b25291f0 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz

{
   "application_type": "web",
   "redirect_uris":
     ["https://www.mymultibank.cz/start",
      "https://www.mymultibank.cz/start2"],
   "client_name": "Moje univerzální banka",
   "client_name#en-US": "My cool bank",
   "logo_uri": "https://www.mybank.cz/logo.png",
   "contact": "info@mybank.cz",
   "scopes": ["aisp", "pisp"]
}
```

A response example:

```
HTTP/1.1 200
Content-Type: application/json

{
   "client_id": "a0b25291f0",
   "client_secret_expires_at": 0,
```

```
   "application_type": "web",
   "redirect_uris":
     ["https://www.mymultibank.cz/start",
      "https://www.mymultibank.cz/start2"],
   "client_name": "Moje univerzální banka",
   "client_name#en-US": "My cool bank",
   "logo_uri": "https://www.mybank.cz/logo.png",
   "contact": "info@mybank.cz",
   "scopes": ["aisp", "pisp"]
}
```

1.4.1.4 0.3 Deleting the application

By calling this resource, the TPP can request deleting the data and access to a specific application. To call a resource, a valid certificate and client_id that is issued to this TPP, must be used. The output is a confirmation of deletion.

Endpoint: DELETE https://idp.banka.cz**/oauth2/register/{client_id}**

Error codes

HTTP STATUS	CODE	DESCRIPTION
401	invalid_client	Invalid client_id.
401	unauthorized_client	The client is not unauthorized to execute the request.
401	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.

A request example:

```
DELETE /oauth2/register/a0b25291f0 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz
```

A response example:

```
HTTP/1.1 201 Created
```

1.4.1.5 0.4 A request for new client_secret

By calling this resource, the TPP may demand the issue of a new client_secret. To call a resource, a valid certificate and client_id that is issued to this TPP, must be used. The original client_secret will be invalidated with the request.

Endpoint: POST https://idp.banka.cz**/oauth2/register/{client_id}/renewSecret**

A response content:

PARAMETR	VALUES	MANDATORY	DESCRIPTION
client_id	string	y	The client_id assigned by the application. This ID starts the authentication process and communication is decoded in the replacement of code and refresh_token.
client_secret	string	y	Client secret - password/token issued by the bank IDP for the TPP application (client_id)
client_secret_expires_at	Time	n	Default value is 0 (client_id never expires). Otherwise, the value is stated in seconds from data 1970-01-01T0:0:0Z

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	invalid_client	Invalid client_id.
401	unauthorized_client	The client is not unauthorized to execute the request.
401	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.

Example of request:

```
POST /oauth2/register/a0b25291f0/renewSecret HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz
```

Example of response:

```
HTTP/1.1 200 OK

{
   "client_id": "a0b25291f0",
   "client_secret": "BBjkk45sd78ad454gddd8712_4555g5g5g5gg",
   "client_secret_expires_at": 0
}
```

1.4.1.6 0.5 Request for a new API key – optional resource (if the Bank uses the API key)

By calling this resource, the TPP may demand the issue of a new API key. To call a resource, a valid certificate and client_id that is issued to this TPP, must be used. The original API key will be invalidated by the request.

Endpoint: POST https://idp.banka.cz**/oauth2/register/{client_id}/renewKey**

A response content:

PARAMETR	VALUES	MANDATORY	DESCRIPTION
client_id	string	y	The client_id assigned by the application. This ID starts the authentication process and communication is decoded in the replacement of code and refresh_token.
api_key	string	y	API key that the application uses to communicate with the bank API. If the API bank does not support API keys, the value „NOT_PROVIDED“ will be returned

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	invalid_client	Invalid client_id.
401	unauthorized_client	The client is not unauthorized to execute the request.
401	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.

A request example:

```
POST /oauth2/register/a0b25291f0/renewKey HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz
```

A response example:

```
HTTP/1.1 200 OK

{
   "client_id": "a0b25291f0",
   "api_key":   "00000001-1212-0f0f-a0a0-123456789abc"
}
```

1.4.2 Authentication resource issued by the Bank

1.4.3 1. Authentication resource

Endpoint: GET https**://**idp.banka.cz**/oauth2/auth**

A request content:

PARAMETR	VALUES	MANDATORY	DESCRIPTION
response_type	code	y	A mandatory parameter. It determines the authentication flow used. In this case, the code grant. For the authentication process it means that as the result of successful identification and authentication, a one-time code is expected instead of the access_token.
client_id	ID of TPP application	y	Unique identifier of the TPP application issued by the bank or the bank IDP. E.g., using the resource „0. Initialization/registration resource“
redirect_uri	URL	y	URL where the authentication flow is directed at the end. The URL is determined on the issue of client_id and within the authentication, the

			parameter is validated against the URL introduced for client_id in the IDP system of the bank. The value should be identical to one of the values introduced by using the resource „0. Initialization/registration resource“.
scope	List of authorizations separated by a space	n	It is a field of applications required by the scope (authorization). In the case of PSD2, it may be the roles of aisp and pisp. E.g., if the TPP is the holder of both authorizations, here it may ask for its application one of them or both, see a request example
state	Any string	n	This parameter may enrich the redirect_uri on the redirecting. It serves for the delivery of information from the application through the authentication flow.

An example of URL for the authentication:

https**://**idp.banka.cz**/oauth2/auth**?state=profil&redirect_uri=https://www.mypfm.cz/start&client_id=MyPFM&response_type=balance&scope=aisp

A response content:

FIELD	DESCRIPTION
code	Authorization field
state	The state parameter from the TPP request

Error codes

HTTP STATUS	CODE	DESCRIPTION
302	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
302	unauthorized_client	The client is not unauthorized to execute the request.
302	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.
302	invalid_scope	Invalid request scope.

A request example:

```
GET /oauth2/auth HTTP/1.1
Host: idp.banka.cz
```

```
Content-Type: application/x-www-form-urlencoded

client_id=MyPFM&
redirect_uri=https://www.mypfm.cz/start&
response_type=code&
scope=aisp pisp&
state=balance
```

A response example:

```
content-type: application/x-www-form-urlencoded
date: Wed, 8 Mar 2017 20:56:28 GMT
location: https://www.mypfm.cz/start?
          code=a200234062baa2ada828bbd33c1f6054&
          state=balance
status: 302
```

An example of error response:

```
HTTP/1.1 302 Found
Location: https://www.mymultibank.com/login?
    error=invalid_request
    &error_description=Unsupported%20response_uri
    &state=login_cz
```

1.4.4 2.a Get token resource

Endpoint: POST https**://**idp.banka.cz**/oauth2/token**

A request content:

FIELD	MANDATORY	DESCRIPTION

code	y	The authorization code returned from the authentication flow (code grant)
client_id	y	ID of TPP application
client_secret	y	Client secret - password/token issued by the bank IDP for the TPP application (client_id)
redirect_uri	y	Redirect URL identical to the URL delivered in the authentication request.
grant_type	y	According to the existing definition/practice, OAuth2 will be the value of authorization_code if the code is replaced by the refresh_token.

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	unauthorized_client	The client is not unauthorized to execute the request.
401	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.

A request example:

```
POST /oauth2/token HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded

code=a200234062baa2ada828bbd33c1f6054&
client_id=MyPFM&
client_secret={client_secret}&
redirect_uri=https://www.mypfm.cz/start&
grant_type=authorization_code
```

A response example:

A successfully processed request will reply with a response with such a defined JSON payload:

```
{
```

```
    "expires_in": 3600,
    "token_type": "Bearer",
    "access_token": "ae9eef9b0af42c674d0b1c1128c37c2d"
    "refresh_token": "be9eef9b0af42c674d0b1c1128c37c2g",
    "acr": "0"
}
```

A response content:

FIELD	DESCRIPTION
access_token	A short-term (in some cases, a one-time) token that may be generated again using the refresh_token. The token serves for the authorization of the request at API.
refresh_token	A long-term token issued based on replacement by the one-time code.
expires_in	A time left before the expiration of access_token - in seconds.
token_type	A token type, e.g., "Bearer"
acr	[optional] Verification level. It may have the values between 0 and 4. Default 3. The value „0" corresponds to nonSCA.

Note: *Within the concept, it is possible to send other unspecified fields, in the response. The system should not assess the response as an error.*

1.4.5 2.b Renewal of access token

The application may store the refresh token from the Get token resource and after the expiration of access_token, ask for a new one, through the refresh token. To that end, it is possible to use the Get token resource with these parameters:

Endpoint: POST https**://**idp.banka.cz**/oauth2/token**

A request content:

FIELD	MANDATORY	DESCRIPTION
client_id	n	ID of TPP application
grant_type	y	According to the existing definition/practice, OAuth2 will be the value of refresh_token, if the access_token is replaced by the refresh_token.
refresh_token	y	Valid refresh_token for which the replacement is carried out, e.g. be9eef9b0af42c674d0b1c1128c37c2g

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	invalid_grant	Invalid authorization. E.g., an invalid refresh token.
401	unauthorized_client	The client is not unauthorized to execute the request.
401	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.

A request example:

```
POST /oauth2/token HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&
refresh_token=be9eef9b0af42c674d0b1c1128c37c2g
```

A response example:

A successfully processed request will reply with a response with such a defined JSON payload:

```
{
   "expires_in": 3600,
   "token_type": "Bearer",
   "access_token": "ae9eef9b0af42c674d0b1c1128c37c2d",
   "acr": "3"
}
```

A response content:

FIELD	DESCRIPTION
access_token	A short-term (in some cases, a one-time) token that may be generated again using the refresh_token. The token serves for the authorization of the request at API.
expires_in	A time left before the expiration of access_token – in seconds.

token_type	A token type, e.g., “Bearer”
acr	[optional] Verification level. It may have the values between 0 and 4. Default 3 or 4. The value „0“ automatically corresponds to á nonSCA. The values 1 - 4 correspond to values defined by the ISO 29115 standard.

Note: *Within the concept, it is possible to send other unspecified fields, in the response. The system should not assess the response as an error.*

1.4.6 3. Token invalidation

A function to invalidate an access or refresh token.

Endpoint: POST https**://**idp.banka.cz**/oauth2/revoke**

PARAMETR	DESCRIPTION
token	OAuth2 access or refresh token acquired based on the authentication process after the replacement by code or refresh token (in the case of access_token)

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	invalid_grant	Invalid authorization. E.g., an invalid refresh token.
401	unauthorized_client	The client is not unauthorized to execute the request.
401	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.

A request example:

```
POST /oauth2/revoke HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded

token=be9eef9b0af42c674d0b1c1128c37c2g
```

1.4.7 Error codes used in the enrolment

An overview and description of the error statutes used in the enrolment flow (TPP registration and authentication). The format and use are based on the OpenID Connect standard.

Error elements

CODE	MANDATORY	DESCRIPTION
error	y	Contains the error code
error_description	n	Expanded text description of the error

Error codes

HTTP STATUS	CODE	DESCRIPTION
400, 302	invalid_request	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401, 302	invalid_client	Invalid client_id.
401	invalid_grant	Invalid authorization. E.g., an invalid refresh token.
401	invalid_token	An invalid token used.
401, 302	unauthorized_client	The client is not unauthorized to execute the request.
302	access_denied	Authorization server denied an access.
500, 503	server_error	Authorization server error.
400	invalid_scope	Invalid request scope.
403	insufficient_scope	For instance, and insufficient authorization to use the required scope.
400	invalid_redirect_uri	The value of one or more redirect uri is not valid.

An example of error response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "invalid_request",
    "error_description": "One or more values are invalid"
}
```

1.4.8 TPP authentication

The precondition of the solution is the use of a qualified eIDAS certificate to identify and authenticate a communicating third party. The directive allows the use of a web certificate or seal certificate. Each certificate has a specific method of use.

1.4.8.1 Web certificate

Using this certificate would mean using a two-way (mutual) TLS (the Transport Layer Security protocol) as a successor of SSL (Secure Sockets Layer). The certificate would be used to establish TLS communication on both sides of the connection so that a qualified web certificate would have to be used by the bank (ASPSP) and TPP. The third party would be identified by validating the validity and content of its web certificate.

1.4.8.2 Stamps (electronic signature)

Using a seal certificate in this solution would mean requiring a request to be signed by a third party when communicating with the bank (ASPSP). The third party would be verified in this model by evaluating the validity of the signature of the message and the content of the public key sent along with the signature (for example, according to the CAdES standard).

The use of the third-party certificate will be required for all resources described, except for the "1. Authorization Resource," which initiates redirecting to the federated authentication process of the bank (ASPSP).

1.4.9 Communication security

Recommended methods of communication security when using a TLS (web certificate) for TPP authentication to communicate with the ASPSP API. The goal is not to duplicate existing TLS mechanisms but to reduce vulnerability despite the known weaknesses of these protocols.

1.5 Nomenclature and basic concepts identical across COBS

NOMENCLATURE / ACRONYM / TERM	DESCRIPTION
API	Application Programming Interface - defines a machine-accessible interface for application programming
HTTP	Hypertext Transfer Protocol – Internet protocol
JSON	JavaScript Object Notation - platform-independent way of writing data (data format)
OAuth2	Protocol for request authorization
General name for PSD2 services	Open banking services

REST	As the Representational State Transfer - interface architecture designed for distributed instruments
PIS service	Payment initiation
AIS service	Account information
CIS service	Balance Check
TPP with PIS service	The provider of payment initiation service
TPP with AIS service	The provider account information service
TPP with CIS service	Issuer of card payment instrument
TPP	Provider of the <service name> service

2 Standard governance

The standard will be changed maximum once a year, suggestions for change may be given by a new mandatory regulation by the bank, a third party through the ČBA or by the working group itself. The suggestion for change must be approved within a proper amendment procedure 6 months before the due date of the changes in force. This implies that the suggestion for a major change must be submitted no later than 1 year before the planned implementation date.

3 Services defined within the standard

DESCRIPTION of three basic API wholes for provision of individual services. The description is broken down by service type, and contains an overview of API resources and an overview of the request elements and response elements of relevant messages, for each service.

Services defined by the standard:

SERVICE	DESCRIPTION
Payment initiation	A service defined as PIS (Payment Initiation Service) by the PSD2 directive
Account information	A service defined as AIS (Account Information Service) by the PSD2 directive
Balance Check	A service defined by the PSD2 directive as information about sufficient funds provided for CISP providers (Card based payment Instrument Issuer Service Provider)

3.1 API Account Information

3.1.1 How to read API account information

The sender of query for API account information must respect permitted characters. Otherwise, the failure to respect them may lead to rejection. None of these elements may contain separate "/" (slash) at the beginning or end, or two successive slashes in the text.

The permitted character set is based only on the swift character set (that is, exclusively without diacritics - a different character set from the supported characters in CERTIS), i.e. they are the following characters:

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

/ - ? : ( ) . , ' +

Space

Only one query can be sent and processed per call.

All bank account numbers of bank clients in the Czech Republic are defined according to Decree No. 169/2011 Coll. on the rules for the creation of account numbers for making payments. The data element „debtorAccount.identification.iban“ requires an account number in the IBAN format which is defined by the international standard ISO 13616.

Structure of the table of elements:

- **LEVEL** – the character plus „+“ determines the level of element embedding. The basic level is marked „+“ and each another is marked with another character, for instance, the level two is „++“

- **MESSAGE ELEMENT** –element name in the camel format
- **OCCURENCE** – describes the occurrence of fields:
    - [1…1] element is mandatory and only occurs once. In the case of a parent element, at least one embedded element must be filled in. If a mandatory element is embedded, the obligation does not pass to the parent element.
    - [0…1] element is optional and occurs only once.
    - [1…n] element is mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
    - [0…n] element is not mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
- **PAYMENT TYPE** – defines for which message types the current element is relevant
- **FORMAT TYPE** – defines the data format. It may be determined by the ISO 20022 standard valid for the item type PAIN.001 or CAAA.001 or another standard specifying the data structure (e.g., changes, dates, etc). Some items have a specific required format resulting from the payment system environment in the Czech Republic.
- **PRESENTATION** – contains a general role description

The format type defined as „±“ means the parent element which is then subdivided into other elements.

Conditions for element presence:

- When there is no OR but the parent element has more than 2 levels of elements, elements may be filled in cumulatively
- Parent element after [1…1] – at least one embedded element must be filled in

3.1.2 List of API resources Account Information

Specification of API Information about the account contains a description of resources for the acquisition of information about the payment account/accounts of the bank client.

Overview of resources:

- GET list of client payment accounts
- GET account balance
- GET transaction overview

3.1.3 List of client payment accounts (GET /my/accounts{?size,page,sort,order})

Paged list of client accounts. Each account contains a unique id usable for URI referencing, e.g. for the account detail.

Resource characteristics

URI:	/my/accounts{?size,page,sort,order}
HTTP Method:	GET
Authorization:	request **requires** the authorization of user/client as part of API calling
Use certificate:	request **requires** the use of the qualified third-party certificate
Paging:	yes
Sorting:	yes
Filtering:	no

Query parameters of the request:

PARAMETR	*TYPE*	*MANDATORY*	*PURPOSE*
size	Number	No	Paging. Number of entries per page
page	Number	No	Paging. Required page. + Default: 0
sort	Text	No	A list of fields separated by comma for sorting, arranged according to the meaning
order	Text	No	A list of arrangement methods (ASC, DESC) separated by comma. The order corresponds to the order of fields in the sort parameter.

Parameters of the request header:

PARAMETR	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETR	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of GET response for calling, please see Chapter 3.1.3.1 RESPONSE MESSAGE ELEMENTS List of client payment accounts

Error codes defined for the GET service, List of client payment accounts

HTTP STATUS CODE	ERROR CODE	PURPOSE
403	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
400	PAGE_NOT_FOUND	Request for a non-existing page
400	PARAMETER_INVALID	The value of the parameter is not valid

3.1.3.1 RESPONSE MESSAGE ELEMENTS List of client payment accounts

LEVEL	MESSAGE ELEMENT	OCCURENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	accounts	[1..1]	AISP	±	Set of client accounts
++	id	[1..1]	AISP	Text	API Payment account identifier
++	**Chyba! Nenalezen zdroj odkazů.**	[1..1]	AISP	±	Payer account identifier
+++	**Chyba! Nenalezen zdroj odkazů.**	[1..1]	AISP	IBAN2007Identifier	IBAN
+++	other	[0..1]	AISP	Max35Text	Another payer account identifier, e.g. the account number.
++	currency	[0..1]	AISP	CurrencyCode, ISO 4217	Payer account currency
++	servicer	[1..1]	AISP	±	
+++	bankCode	[0..1]	AISP	Text	
+++	countryCode	[0..1]	AISP	CountryCode, ISO 3166	Bank country
+++	bic	[0..1]	AISP	Max35Text	Bank BIC
++	nameI18N	[0..1]	AISP	Text	Account name
++	productI18N	[0..1]	AISP	Text	Product name

3.1.4 Account balance (GET /my/accounts/{id}/balance{?currency})

The balance of the specific client account according to the reference id account.

Resource characteristics

URI: /my/accounts/{id}/balance{?currency}
HTTP method: GET
Authorization: request **requires** the authorization of user/client as part of the API calling
Use certificate: request **requires** the use of the third-party qualified certificate
Paging: no
Sorting: no
Filtering: no

Query parameters of the request:

PARAMETR	*TYPE*	*MANDATORY*	*PURPOSE*
id	Text	Yes	System identifier of client account
currency	Text	No	Required account currency for multi-currency accounts.

Parameters of request header:

PARAMETR	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of response header:

PARAMETR	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of GET response for calling, please see Chapter 3.1.4.1 RESPONSE MESSAGE ELEMENTS Account balance

Error codes defined for the service GET List of client payment accounts

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	Invalid or unknown account ID
400	AC09	[InvalidAccountCurrency] – for multicurrency accounts, or currency not supported According to the table of exchanges

3.1.4.1 RESPONSE MESSAGE ELEMENTS Account balance

LEVEL	MESSAGE ELEMENT	OCCURENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	balances	[1..1]	AISP	±	Set of balances of client payment account
++	type	[1..1]	AISP	±	Determines the balance type to which Information on account balance applies
+++	codeOrProprietary	[1..1]	AISP	±	
++++	code	[1..1]	AISP	Type of balance	Codes of balance types
++	creditLine	[0..1]	AISP	±	Amount of agreed permitted debit/overdraft
+++	included	[0..1]	AISP	Bulean	
+++	amount	[0..1]	AISP	±	Amount of agreed permitted debit
++++	value	[0..1]	AISP	Number	Amount of agreed permitted debit
++++	currency	[0..1]	AISP	Text	Currency corresponds to the account currency to which the statement is generated
++	amount	[1..1]	AISP	±	Value/amount of account balance

					according to balance type. Currency corresponds to the account currency to which the statement is generated
+++	value	[1..1]	AISP	Number	Amount of account balance
+++	currency	[1..1]	AISP	Text	Currency corresponds to the account currency to which the statement is generated
++	creditDebitIndicator	[1..1]	AISP	Text	Indication of whether the balance in the account for which the statement is generated is positive or negative
++	date	[1..1]	AISP	±	Date (and time) of balance of the account for which the statement is generated
+++	dateTime	[1..1]	AISP	Text	Date or date and time of the balance according to ISO 8601

3.1.5 Overview of transactions (GET /my/accounts/{id}/transactions{?fromDate,toDate,currency,size,page,sort,order})

Paged list of transactions of a selected client account.

Resource characteristics

URI: /my/accounts/{id}/transactions{?fromDate,toDate,currency,size,page,sort,order}
HTTP Method: GET
Authorization: request **requires** the authorization of user/client as part of the API calling
Use certificate: request **requires** the use of the third-party qualified certificate
Paging: yes
Sorting: yes
Filtering: no

Query parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
id	Text	Yes	System identifier of client account
currency	Text	No	Required account currency for multi-currency accounts.
fromDate	Text	No	Date and time of the start of required transaction history
toDate	Text	No	Date and time of the end of required transaction history [inclusive]
currency	Text	No	Required account currency for multi-currency accounts
size	Number	No	Paging. Number of entries per page
page	Number	No	Paging. Required page. + Default: 0
sort	Text	No	A list of fields separated by comma for sorting, arranged according to the meaning
order	Text	No	A list of arrangement methods (ASC, DESC) separated by comma. The order corresponds to the order of fields in the sort parameter.

Parameters of the request header:

PARAMETR	TYPE	MANDATORY	PURPOSE
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETR	TYPE	MANDATORY	PURPOSE
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of GET response for calling, please see Chapter 3.1.5.1 RESPONSE MESSAGE ELEMENTS Overview of transactions

Error codes defined for the GET service List of client payment accounts:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	Invalid or unknow account ID
404	PAGE_NOT_FOUND	Query for not existing page
400	PARAMETER_INVALID	Parameter value is not valid
400	AC09	[InvalidAccountCurrency] – for multicurrency accounts, or currency not supported According to the table of exchanges
400	DT01	[InvalidDate] Invalid date

3.1.5.1 RESPONSE MESSAGE ELEMETS Overview of transactions

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
+	entryReference	Max35Text	Identification no. of the payment assigned by the bank.
+	amount	Amount	Payment amount in the account currency for which payment history is generated.
++	value	Amount	
++	currency	CurrencyCode	
+	creditDebitIndicator	CreditDebitCode	Indication of whether it is a debit payment or credit payment of the account. An expression of whether it is a debit or credit payment is marked with one of the codes below: DBIT: to the debit of CRDT: In other cases

+	reversalIndicator	TrueFalseIndicator	Indication of whether it is a cancellation. An expression of whether it is a cancellation or not is marked with one of the codes below: true: It is cancellation false: It is not cancellation
+	status	Code	Item status (debited or credited payments) of the account from the point of view of the bank. The statement will present only posted items, with the constant BOOK, or blocked items, with the constant PDNG.
+	bookingDate	±	Date of processing/posting of payment by bank in the format ISODate, or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.sTZD.
++	date	ISODate ISODateTime	Date of processing/posting of payment by bank in the format ISODate, or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.sTZD, depending on the transaction type and method how the bank presents data (and time) of payment processing/posting. Mainly for card or cash transactions it is posted as ISODateTime. 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	±	Due date/payment foreign currency in the format ISODate, or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.sTZD.

++	date	ISODate/ISODateTime	Due date/payment foreign currency in the format ISODate, , or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.sTZD, depending on the transaction type and method how the bank presents data (and time) of due date/ payment foreign currency. Mainly for card or cash transactions it is posted as ISODateTime. 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	±	The code of banking transaction according to the code list of the Czech Banking Association assigned to a specific payment. Each bank uses an own code list to identify payments, which is, however, based on the 1st to 3rd level of transaction code list according to the CBA Standard for camt.053.
++	proprietary	±	The code of banking transaction according to the code list of the Czech Banking Association assigned to a specific payment. Each bank uses an own code list to identify payments, which is, however, based on the 1st to 3rd level of transaction code list according to the CBA Standard for camt.053.
+++	code	Max35Text	The code of banking transaction according to the code list of the Czech Banking Association assigned to a specific payment. Each bank uses an own code list to identify payments, which is, however, based on the 1st to 3rd level of transaction code list according to the CBA Standard for camt.053.

+++	issuer	Max35Text	Identification of the code list issuer for banking transactions which acquires the value of the Czech Banking Association.
+	entryDetails	±	Turnover details. The level repeats only for the given item
++	transactionDetails	±	Payment details. The level repeats only for the given item.
+++	references	±	A set of references unambiguously identifying the payment.
++++	messageIdentification	Max35Text	Assumed payment identification entered by the client on its initiation or order of the payment in the payment history.
++++	accountServicerReference	Max35Text	A banking reference assigned to the payment, e.g., on initiation through direct banking services.
++++	paymentInformationIdentification	Max35Text	Another/next banking reference assigned to the payment assigned by the bank, for card payments, the card sequence number may be entered, or the specific symbol may be entered here.
++++	instructionIdentification	Max35Text	Payment identification entered by a third party, or a constant symbol may be entered.

++++	endToEndIdentification	Max35Text	A unique identifier specified by the client initiating the payment that serves for unambiguous payment identification and is passed in the unchanged state throughout the payment chain, or, there may be a variable symbol filled in here.
++++	mandateIdentification	Max35Text	For SEPA collections, Unique Mandate Reference stated for the given SEPA collection, as a mandatory field [1..1].
++++	chequeNumber	Max35Text	For cheque transactions there may be a cheque number; for card transactions there may be a card number with asterisks. The card no. is each time in the format xxxxxxxxxxxx1234 where according to the standard, only 0-9 may be used (8-28 char.) i.e. without illustrative xxxx.
++++	clearingSystemReference	Max35Text	A bank-defined code list value identifying the type of payment or payment type used. For card transactions, card association identification may be indicated.
+++	amountDetails	+	Details of the payment amount, especially if it is a conversion payment or cashback.

++++	instructedAmount	+	Amount and currency of payment in the currency that was required by the client for transfer. E.g., for intra-bank payments, the payer account currency and the payment currency if the client required the payment in the payer account currency.
+++++	amount	Amount	The original amount and currency of payment that was required by the client for transfer.
++++++	value	Amount	
++++++	currency	CurrencyCode	
++++	transactionAmount	+	The amount and currency of payment for cumulated payments and Cashback.
+++++	amount	Amount	The amount and currency of payment for cumulated payments and Cashback, where the total amount of payment including the part for Cashback is entered here.
++++++	value	Amount	
++++++	currency	CurrencyCode	
++++	counterValueAmount	+	The amount and currency of payment in the client account currency after conversion of amount that was required by the client for transfer.
+++++	amount	Amount	The final amount and payment currency that was required by the client for transfer.

++++++	value	Amount	
++++++	currency	CurrencyCode	
+++++	currencyExchange	+	Information on currencies and exchange rates used.
++++++	sourceCurrency	CurrencyCode	Client account currency (source/original currency/payer account currency for intrabank conversion payments).
++++++	targetCurrency	CurrencyCode	Payment currency (End/target currency/payee account currency for intrabank conversion payments).
++++++	exchangeRate	BaseOneRate	The exchange rate used to charge a payment. Only one rate is filled in, even in the case of cross conversion.
++++	proprietaryAmount	+	Amount of cash withdrawal through the Cashback service.
+++++	type	Max35Text	The constant "CASHBACK" is to be filled in.
+++++	amount	Amount	Cashback amount and currency – only the part of total payment, to which the cash amount withdrawn falls within the Cashback service.
++++++	value	Amount	
++++++	currency	CurrencyCode	
+++	charges	+	Information on fees.

++++++	bearer	Code	Indication of that payments will be charged to his debit (OUR, SHA, BEN).
+++	relatedParties	+	Information on the payer, the payer's account and the original payer and payee, the payee's account and the final payee in the payment.
++++	debtor	+	Information on the payer. It is according to the payment direction, and filled in in the case of counterparty.
+++++	name	Max140Text	Payer name.
+++++	postalAddress	+	Payer postal address.
++++++	streetName	Max70Text	Street name used for the payer postal address.
++++++	buildingNumber	Max16Text	Land registry no. used for the payer postal address.
++++++	postCode	Max16Text	Postcode used for the payer postal address.
++++++	townName	Max35Text	Town name used for the payer postal address.
++++++	country	CountryCode	Country name used for the payer postal address.
++++++	addressLine	Max70Text	Unstructured record of payer postal address.
+++++	identification	+	Payer identification.
++++++	organisationIdentification	+	Clear payer identification as the organization/legal entity.

+++++++	bicOrBei	BICIdentifier	Payer identification as the organization/legal entity in the form of BIC or BEI code.
+++++++	other	+	Other payer identification as the organization/legal entity.
++++++++	identification	Max35Text	Other payer identification as the organization/legal entity in unstructured form.
++++++++	schemeName	+	Code type for payer identification as the organization/legal entity.
+++++++++	code	Code	Code type for payer identification as the organization/legal entity in the form of code according to the ISO code list.
+++++++++	proprietary	Max35Text	Code type for payer identification as the organization/legal entity in the free text format.
++++++++	issuer	Max35Text	Code issuer for the payer identification as the organization/legal entity.
++++++	privateIdentification	+	Clear payer identification as the natural person.
+++++++	other	+	Another payer identification as the natural person.
++++++++	identification	Max35Text	Other payer identification as the natural person in unstructured form.

++++++++	schemeName	+	Code type for payer identification as the natural person.
+++++++++	code	Code	Code type for payer identification as the natural person in the form of code according to the ISO code list.
+++++++++	proprietary	Max35Text	Code type for payer identification as the natural person in free text format.
++++++++	issuer	Max35Text	Code issuer for payer identification as the natural person.
++++	debtorAccount	+	Information on the account payer. It is according to the payment direction, and filled in in the case of counterparty.
+++++	identification	+	Identification of account payer type.
++++++	iban	IBAN2007Identifier	Payer account no. in the international format of IBAN account no.
++++++	other	+	Payer account no. in other/local format of account no.
+++++++	identification	Max34Text	The value of payer account no. in other/local format of account no.
+++++	currency	CurrencyCode	Payer account currency.
+++++	name	Max70Text	Payer account name.
++++	ultimateDebtor	+	Information on the original/actual payer. It is according to the payment direction, and filled in in the case of counterparty.

+++++	name	Max140Text	Name of the original/actual payer.
+++++	postalAddress	+	Postal address of the original payer.
++++++	streetName	Max70Text	Street name used for the payer postal address of the original payer.
++++++	buildingNumber	Max16Text	Land registry no. used for the original payer postal address.
++++++	postCode	Max16Text	Postcode used for the original payer postal address.
++++++	townName	Max35Text	Town name used for the original payer postal address.
++++++	country	CountryCode	Country name used for the original payer postal address.
++++++	addressLine	Max70Text	Unstructured record of original payer postal address.
+++++	identification	+	Original payer identification.
++++++	organisationIdentification	+	Clear identification of the original payer as the organization/legal entity.
+++++++	bicOrBei	BICIdentifier	Identification of the original payer as the organization/legal entity in the form of BIC or BEI code.
+++++++	other	+	Other identification of the original payer as the organization/legal entity.

++++++++	identification	Max35Text	Other identification of the original payer as the organization/legal entity in unstructured form.
++++++++	schemeName	+	Code type for the identification of the original payer as the organization/legal entity.
+++++++++	code	Code	Code type for the identification of the original payer as the organization/legal entity in the form of code according to the ISO code list.
+++++++++	proprietary	Max35Text	Code type for the identification of the original payer as the organization/legal entity in free text format.
++++++++	issuer	Max35Text	Code issuer for the original payer identification as the organization/legal entity.
++++++	privateIdentification	+	Clear identification of the original payer as the natural person.
+++++++	other	+	Other identification of the original payer as the natural person.
++++++++	identification	Max35Text	Other identification of the original payer as the natural person in unstructured form.
++++++++	schemeName	+	Code type for the identification of the original payer as the natural person.

+++++++++	code	Code	Code type for the identification of the original payer as the natural person in the form of code according to the ISO code list.
+++++++++	proprietary	Max35Text	Code type for the identification of the original payer as the natural person in the free text format.
++++++++	issuer	Max35Text	Code issuer for the identification of the original payer as the natural person.
++++	creditor	+	Information on the payee. It is according to the payment direction, and filled in in the case of counterparty.
+++++	name	Max140Text	Payee name.
+++++	postalAddress	+	Payee postal address.
++++++	streetName	Max70Text	Street name used for the payee postal address.
++++++	buildingNumber	Max16Text	Land registry no. used for the payee postal address.
++++++	postCode	Max16Text	Postcode used for the payee postal address.
++++++	townName	Max35Text	Town name used for the payee postal address.
++++++	country	CountryCode	Country name used for the payee postal address.
++++++	addressLine	Max70Text	Unstructured record of the payee postal address.
+++++	identification	+	Payee identification.

++++++	organisationIdentification	+	Clear identification of the payee as the organization/legal entity.
+++++++	bicOrBei	BICIdentifier	Payee identification as the organization/legal entity in the form of BIC or BEI code.
+++++++	other	+	Other identification of the payee as the organization/legal entity.
++++++++	identification	Max35Text	Other identification of the payee as the organization/legal entity in unstructured form.
++++++++	schemeName	+	Code type for the payee identification as the organization/legal entity.
+++++++++	code	Code	Code type for the payee identification as the organization/legal entity in the form of code according to the ISO code list.
+++++++++	proprietary	Max35Text	Code type for the payee identification as the organization/legal entity in the free text format
++++++++	issuer	Max35Text	Code issuer for the payee identification as the organization/legal entity.
++++++	privateIdentification	+	Clear identification of the payee as the natural person.

+++++++	other	+	Other identification of the payee as the natural person.
++++++++	identification	Max35Text	Other identification of the payee as the natural person in unstructured form.
++++++++	schemeName	+	Code type for the payee identification as the natural person.
+++++++++	code	Code	Code type for the payee identification as the natural person in the form of code according to the ISO code list.
+++++++++	proprietary	Max35Text	Code type for the payee identification as the natural person in the free text format.
++++++++	issuer	Max35Text	Code issuer for the payee identification as the natural person.
++++	creditorAccount	+	Information on the payee account. It is according to the payment direction, and filled in in the case of counterparty.
+++++	identification	+	Identification of payee account type.
++++++	iban	IBAN2007Identifier	Payee account no. in the international format of IBAN account no.
++++++	other	+	Payee account no. in other/local format of account no.
+++++++	identification	Max34Text	The value of payee account no. in other/local format of account no.

+++++	currency	CurrencyCode	Payee account currency.
+++++	name	Max70Text	Payee account name.
++++	ultimateCreditor	+	Information on the final payee. It is according to the payment direction, and filled in in the case of counterparty.
+++++	name	Max140Text	Final payee name.
+++++	postalAddress	+	Final payee postal address.
++++++	streetName	Max70Text	Street name used for the final payee postal address.
++++++	buildingNumber	Max16Text	Land registry no. used for the final payee postal address.
++++++	postCode	Max16Text	Postcode used for the final payee postal address.
++++++	townName	Max35Text	Town name used for the final payer postal address.
++++++	country	CountryCode	Country name used for the final payer postal address.
++++++	addressLine	Max70Text	Unstructured record of final payee postal address.
+++++	identification	+	Final payee identification.
++++++	organisationIdentification	+	Clear identification of the final payee as the organization/legal entity.

+++++++	bicOrBei	BICIdentifier	Identification of the final payee as the organization/legal entity in the form of the BIC or BEI code.
+++++++	other	+	Other identification of the final payee as the organization/legal entity.
++++++++	identification	Max35Text	Other identification of the final payee as the organization/legal entity in unstructured form.
++++++++	schemeName	+	Code type for the identification of final payee as the organization/legal entity.
+++++++++	code	Code	Code type for the identification of final payee as the organization/legal entity in the form of code according to the ISO code list.
+++++++++	proprietary	Max35Text	Code type for the identification of final payee as the organization/legal entity in free text format.
++++++++	issuer	Max35Text	Code issuer for the identification of the final payee as the organization/legal entity.
++++++	privateIdentification	+	Clear identification of the final payee as the natural person.

+++++++	other	+	Other identification of the final payee as the natural person.
++++++++	identification	Max35Text	Other identification of the final payee as the natural person in unstructured form.
++++++++	schemeName	+	Code type for the identification of the final payee as the natural person.
+++++++++	code	Code	Code type for the identification of the final payee as the natural person in the form of code according to the ISO code list.
+++++++++	proprietary	Max35Text	Code type for the identification of the final payee as the natural person in the free text format.
++++++++	issuer	Max35Text	Code issuer for the identification of the final payee as the natural person.
++++	proprietary	+	For more detailed identification of on which ATM the card transaction occurred.
+++++	type	Max35Text	For identification of whether for the card transaction it is own / external ATM.
+++++	party	+	For identification of the ATM name/owner.
++++++	name	Max140Text	For identification of ATM name/location.
+++	relatedAgents	+	Information on the payer bank and the payee bank in the payment.

++++	debtorAgent	+	Information on the payer bank. It is according to the payment direction, and filled in in the case of counterparty.
+++++	financialInstitutionIdentification	+	Payer bank code in the international format BIC / SWIFT code. Value filled in (constant): KOMBCZPPXXX.
++++++	bic	BICIdentifier	BIC / SWIFT code of the payer bank Value filled in (constant): KOMBCZPPXXX.
++++++	clearingSystemMemberIdentification	+	Local format of the payer bank code – either in the code form or in the form of text description.
+++++++	clearingSystemIdentification	+	Payer bank identification in the local payment system in which the payer bank works.
++++++++	code	Code	Payer bank identification in the local payment system in which the payer bank works in the form of payment system code.
++++++++	proprietary	Max35Text	Payer bank identification in the local payment system in which the payer bank works in unstructured form of descriptive text.
+++++++	memberIdentification	Max35Text	Local code format of

			the payer bank code.
++++++	name	Max140Text	Payer bank code as stated in the code list of world banks (SWIFT Directory). Value filled in (constant): KOMERCNI BANKA A.S.
++++++	postalAddress	+	Postal address of the payer bank.
+++++++	streetName	Max70Text	Street name used for the payer postal address.
+++++++	buildingNumber	Max16Text	Land registry no. used for the payee postal address.
+++++++	postCode	Max16Text	Postcode used for the payer postal address.
+++++++	townName	Max35Text	Town name used for the payer postal address.
+++++++	country	CountryCode	Country name used for the payer postal address.
+++++++	addressLine	Max70Text	Unstructured record of payer postal address.
++++++	other	+	Other payee bank identification.
+++++++	identification	Max35Text	Other payer bank identification, mainly in the form of the so-called local bank code.
++++	creditorAgent	+	Information on the payee bank. It is according to the payment direction, and filled in in the case of counterparty.

+++++	financialInstitutionIdentification	+	Code of the payee bank in the international format BIC / SWIFT code. Value filled in (constant): KOMBCZPPXXX.
++++++	bic	BICIdentifier	BIC / SWIFT code of the payee bank Value filled in (constant): KOMBCZPPXXX.
++++++	clearingSystemMemberIdentification	+	Local format of the payee bank code - either in the code form or in the form of text description.
+++++++	clearingSystemIdentification	+	Payee bank identification in the local payment system in which the payee bank works.
++++++++	code	Code	Payee bank identification in the local payment system in which the payee bank works in the form of payment system code.
++++++++	proprietary	Max35Text	Payee bank identification in the local payment system in which the payee bank works in unstructured form of descriptive text.
+++++++	memberIdentification	Max35Text	Local code format of the payee bank code.
++++++	name	Max140Text	Payee bank name.
++++++	postalAddress	+	Payee bank postal address.

+++++++	streetName	Max70Text	Street name used for the payee postal address.
+++++++	buildingNumber	Max16Text	Land registry no. used for the payee postal address.
+++++++	postCode	Max16Text	Postcode used for the payee postal address.
+++++++	townName	Max35Text	Town name used for the payee postal address.
+++++++	country	CountryCode	Country name used for the payee postal address.
+++++++	addressLine	Max70Text	Unstructured record of the payee postal address.
++++++	other	+	Other payee bank identification, mainly in the form of the so-called local bank code.
+++++++	identification	Max35Text	Information on the payee bank. It is according to the payment direction, and filled in in the case of counterparty.
+++	purpose	+	Payment purpose.
++++	code	Code	Payment purpose expressed as a code that was entered in the payment.
++++	proprietary	Max35Text	Payment purpose expressed as unstructured information that was entered in the payment.
+++	remittanceInformation	+	Additional payment information.

++++	unstructured	Max140Text	Additional payment information filled in in the payment as unstructured information. If the payment states several repetitions of unstructured record supplementing payment information, then only the first of them will be considered.
++++	structured	+	The completion of payment symbols (variable, specific and constant) is expected. If the payment does not have a variable, specific or constant symbol, then the entire Structured Remittance Information will be left empty.
+++++	creditorReferenceInformation	+	To display information on variable, specific and constant symbol.

++++++	reference	Max35Text	Start the field with the VS code: (for variable symbol), or SS: (for specific symbol) or KS: (for constant symbol). In one repetition of the reference field, all three symbols may be stated, each time maximum one repetition of each individual symbol. The method of recording symbols is VS:|KS:|SS:)[0-9]{1,10}.
+++	additionalTransactionInformation	Max500Text	Additional information provided by the bank. Only stated if it is included in the record. Additional info for SEPA DD is entered here (e.g., Creditor Identifier, Payment scheme, SEPA collection order, etc.)

3.1.5.1 MESSAGE ELEMENTS Overview of transactions - occurrence in individual payment types

3.1.5.1.1 Interest – occurrence of elements

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]

+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[0..1]
++	transactionDetails	[0..1]
+++	references	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]
++++++	targetCurrency	[0..1]

++++++	exchangeRate	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++	relatedParties	[0..1]
++++	debtor	[0..1]
+++++	name	[0..1]
++++	debtorAccount	[0..1]
+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[0..1]
+++++	identification	[1..1]
++++++	iban	[1..1].. either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
+++	relatedAgents	[0..1]
++++	debtorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	name	[0..1]…or
++++++	other	[0..1]…or
+++++++	identification	[1..1]
++++	creditorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]… either
++++++	name	[0..1]… or
++++++	other	[0..1]… or

+++++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
++++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.2 Fee – occurrence of elements

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[0..1]
++	transactionDetails	[0..1]
+++	references	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]

++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]
++++++	targetCurrency	[0..1]
++++++	exchangeRate	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++	relatedParties	[0..1]
++++	debtor	[0..1]
+++++	name	[0..1]
++++	debtorAccount	[0..1]
+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification
++++++	other	[1..1]..either or IBAN
+++++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[0..1]

+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification
++++++	other	[1..1]..either or IBAN
+++++++	identification	[1..1]
+++	relatedAgents	[0..1]
++++	debtorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	name	[0..1]…or
++++++	other	[0..1]…or
+++++++	identification	[1..1]
++++	creditorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	name	[0..1]…or
++++++	other	[0..1]…or
+++++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
++++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.3 Domestic payment – occurrence of elements

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]

+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[1..1]
++	transactionDetails	[1..1]
+++	references	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]

++++++	targetCurrency	[0..1]
++++++	exchangeRate	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++	relatedParties	[1..1]
++++	debtor	[1..1]
+++++	name	[1..1]
++++	debtorAccount	[1..1]
+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[1..1]
+++++	identification	[1..1]
++++++	iban	[1..1].. either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
+++	relatedAgents	[1..1]
++++	debtorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	name	[0..1]…or
++++++	other	[0..1]…or
+++++++	identification	[1..1]
++++	creditorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	name	[0..1]…or

++++++	other	[0..1]…or
+++++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
++++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.4 SEPA payment – occurrence of elements

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[1..1]
++	transactionDetails	[1..1]
+++	references	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]

++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	mandateIdentification	[0..0]/[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]
++++++	targetCurrency	[0..1]
++++++	exchangeRate	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++	relatedParties	[1..1]
++++	debtor	[1..1]
+++++	name	[1..1]
+++++	postalAddress	[0..1]
++++++	streetName	[0..1]
++++++	buildingNumber	[0..1]
++++++	postCode	[0..1]
++++++	townName	[0..1]
++++++	country	[0..1]
++++++	addressLine	[0..7]
+++++	identification	[0..1]
++++++	organisationIdentification	[1..1]…either

+++++++	bicOrBei	[0..1]
+++++++	other	[0..n]
++++++++	identification	[1..1]
++++++++	schemeName	[0..1]
+++++++++	code	[1..1]
+++++++++	proprietary	[1..1]
++++++++	issuer	[0..1]
++++++	privateIdentification	[1..1]…or
+++++++	other	[0..n]
++++++++	identification	[1..1]
++++++++	schemeName	[0..1]
+++++++++	code	[1..1]
+++++++++	proprietary	[1..1]
++++++++	issuer	[0..1]
++++	debtorAccount	[1..1]
+++++	identification	[1..1]
++++++	iban	[1..1]
+++++	name	[0..1]
++++	ultimateDebtor	[0..1]
+++++	name	[0..1]
+++++	postalAddress	[0..1]
++++++	streetName	[0..1]
++++++	buildingNumber	[0..1]
++++++	postCode	[0..1]
++++++	townName	[0..1]
++++++	country	[0..1]
++++++	addressLine	[0..7]
+++++	identification	[0..1]
++++++	organisationIdentification	[1..1]…either
+++++++	bicOrBei	[0..1]
+++++++	other	[0..n]
++++++++	identification	[1..1]
++++++++	schemeName	[0..1]
+++++++++	code	[1..1]

+++++++++	proprietary	[1..1]
++++++++	issuer	[0..1]
++++++	privateIdentification	[1..1]…or
+++++++	other	[0..n]
++++++++	identification	[1..1]
++++++++	schemeName	[0..1]
+++++++++	code	[1..1]
+++++++++	proprietary	[1..1]
++++++++	issuer	[0..1]
++++	creditor	[1..1]
+++++	name	[1..1]
+++++	postalAddress	[0..1]
++++++	streetName	[0..1]
++++++	buildingNumber	[0..1]
++++++	postCode	[0..1]
++++++	townName	[0..1]
++++++	country	[0..1]
++++++	addressLine	[0..7]
+++++	identification	[0..1]
++++++	organisationIdentification	[1..1]…either
+++++++	bicOrBei	[0..1]
+++++++	other	[0..n]
++++++++	identification	[1..1]
++++++++	schemeName	[0..1]
+++++++++	code	[1..1]
+++++++++	proprietary	[1..1]
++++++++	issuer	[0..1]
++++++	privateIdentification	[1..1]…or
+++++++	other	[0..n]
++++++++	identification	[1..1]
++++++++	schemeName	[0..1]
+++++++++	code	[1..1]
+++++++++	proprietary	[1..1]
++++++++	issuer	[0..1]

++++	creditorAccount	[1..1]
+++++	identification	[1..1]
++++++	iban	[1..1]
+++++	name	[0..1]
++++	ultimateCreditor	[0..1]
+++++	name	[0..1]
+++++	postalAddress	[0..1]
++++++	streetName	[0..1]
++++++	buildingNumber	[0..1]
++++++	postCode	[0..1]
++++++	townName	[0..1]
++++++	country	[0..1]
++++++	addressLine	[0..7]
+++++	identification	[0..1]
++++++	organisationIdentification	[1..1]
+++++++	bicOrBei	[0..1]
+++++++	other	[0..n]
++++++++	identification	[1..1]
++++++++	schemeName	[0..1]
+++++++++	code	[1..1]
+++++++++	proprietary	[1..1]
++++++++	issuer	[0..1]
++++++	privateIdentification	[1..1]
+++++++	other	[0..n]
++++++++	identification	[1..1]
++++++++	schemeName	[0..1]
+++++++++	code	[1..1]
+++++++++	proprietary	[1..1]
++++++++	issuer	[0..1]
+++	relatedAgents	[1..1]
++++	debtorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[1..1]
++++++	name	[0..1]…or

++++	creditorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[1..1]
++++++	name	[0..1]…or
+++	purpose	[0..1]
++++	code	[1..1]…either
++++	proprietary	[1..1]…or
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
++++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.5 Foreign payment – occurrence of elements

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[1..1]
++	transactionDetails	[1..1]

+++	references	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]
++++++	targetCurrency	[0..1]
++++++	exchangeRate	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++	charges	[0..1]
++++++	bearer	[0..1]
+++	relatedParties	[1..1]
++++	debtor	[1..1]
+++++	name	[1..1]
+++++	postalAddress	[0..1]
++++++	streetName	[0..1]
++++++	buildingNumber	[0..1]
++++++	postCode	[0..1]

++++++	townName	[0..1]
++++++	country	[0..1]
++++++	addressLine	[0..7]
++++++++	issuer	[0..1]
++++	debtorAccount	[1..1]
+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
+++++	name	[0..1]
++++	creditor	[1..1]
+++++	name	[1..1]
+++++	postalAddress	[0..1]
++++++	streetName	[0..1]
++++++	buildingNumber	[0..1]
++++++	postCode	[0..1]
++++++	townName	[0..1]
++++++	country	[0..1]
++++++	addressLine	[0..7]
++++	creditorAccount	[1..1]
+++++	identification	[1..1]
++++++	iban	[1..1].. either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
+++++	name	[0..1]
+++	relatedAgents	[1..1]
++++	debtorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	clearingSystemMemberIdentification	[0..1]…or

+++++++	clearingSystemIdentification	[0..1]
++++++++	code	[1..1]…either
++++++++	proprietary	[1..1]…or
+++++++	memberIdentification	[1..1]
++++++	name	[0..1]…or
++++++	postalAddress	[0..1]…or
+++++++	streetName	[0..1]
+++++++	buildingNumber	[0..1]
+++++++	postCode	[0..1]
+++++++	townName	[0..1]
+++++++	country	[0..1]
+++++++	addressLine	[0..7]
++++++	other	[0..1]…or
+++++++	identification	[1..1]
++++	creditorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	clearingSystemMemberIdentification	[0..1]…or
+++++++	clearingSystemIdentification	[0..1]
++++++++	code	[1..1]…either
++++++++	proprietary	[1..1]…or
+++++++	memberIdentification	[1..1]
++++++	name	[0..1]…or
++++++	postalAddress	[0..1]…or
+++++++	streetName	[0..1]
+++++++	buildingNumber	[0..1]
+++++++	postCode	[0..1]
+++++++	townName	[0..1]
+++++++	country	[0..1]
+++++++	addressLine	[0..7]
++++++	other	[0..1]…or
+++++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]

++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
++++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.6 Cash – occurrence of elements

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[0..1]
++	transactionDetails	[0..1]
+++	references	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]

++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]
++++++	targetCurrency	[0..1]
++++++	exchangeRate	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++	relatedParties	[0..1]
++++	debtor	[0..1]
+++++	name	[0..1]
++++	debtorAccount	[0..1]
+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[0..1]
+++++	identification	[1..1]
++++++	iban	[1..1].. either or other identification
++++++	other	[1..1].. either or IBAN

+++++++	identification	[1..1]
+++	relatedAgents	[0..1]
++++	debtorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	name	[0..1]…or
++++++	other	[0..1]…or
+++++++	identification	[1..1]
++++	creditorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]… either
++++++	name	[0..1]… or
++++++	other	[0..1]… or
+++++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
++++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.7 Card transactions – occurrence of elements

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]

++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[0..1]
++	transactionDetails	[0..1]
+++	references	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]
++++++	targetCurrency	[0..1]
++++++	exchangeRate	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]

+++	relatedParties	[0..1]
++++	debtor	[0..1]
+++++	name	[0..1]
++++	debtorAccount	[0..1]
+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification
++++++	other	[1..1]..either or IBAN
+++++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[0..1]
+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
+++	relatedAgents	[0..1]
++++	debtorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	name	[0..1]…or
++++++	other	[0..1]…or
+++++++	identification	[1..1]
++++	creditorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[0..1]…either
++++++	name	[0..1]…or
++++++	other	[0..1]…or
+++++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]

++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
++++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.2 API Payment Initiation

3.2.1 How to read API Payment Initiation

The sender of query for API Payment Initiation must respect permitted characters mainly in the identifications and references that are sent out to the partner bank. Otherwise, the failure to respect them may lead to rejection. None of these elements may contain separate "/" (slash) at the beginning or end, or two successive slashes in the text.

The permitted character set is based only on the swift character set (that is, exclusively without diacritics - a different character set from the supported characters in CERTIS), i.e. they are the following characters:

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

/ - ? : ( ) . , ' +

Space

Only one query can be sent and processed per call.

All bank account numbers of bank clients in the Czech Republic are defined according to Decree No. 169/2011 Coll. on the rules for the creation of account numbers for making payments. The data element „debtorAccount.identification.iban“ requires an account number in the IBAN format which is defined by the international standard ISO 13616.

Structure of the table of elements:

- **LEVEL** – the character plus „+“ determines the level of element embedding. The basic level is marked „+“ and each another is marked with another character, for instance, the level two is „++“
- **MESSAGE ELEMENT** – element name in the camel format
- **OCCURRENCE** – describes the occurrence of fields:
  - [1…1] element is mandatory and only occurs once. In the case of a parent element, at least one embedded element must be filled in. If a mandatory element is embedded, the obligation does not pass to the parent element.
  - [0…1] element is optional and occurs only once.

  - [1…n] element is mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
  - [0…n] element is not mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
- **PAYMENT TYPE** – defines for which message types the current element is relevant
- **FORMAT TYPE** – defines the data format. It may be determined by the ISO 20022 standard valid for the item type PAIN.001, or CAAA.001 or another standard specifying the data structure (e.g., changes, dates, etc). Some items have a specific required format resulting from the payment system environment in the Czech Republic.
- **PRESENTATION** – contains a general role description

The format type defined as „±“ means the parent element which is then subdivided into other elements.

Conditions for element presence:

- When there is no OR but the parent element has more than 2 levels of elements, elements may be filled in cumulatively
- Parent element after [1…1] – at least one embedded element must be filled in

3.2.2 List of API Payment Initiation resources

Specification of API Payment Initiation contains a description of resources for the payment initiation and the Balance Check of the payer (client) at a specific financial institution.

Overview of resources:

- POST query for balance check
- POST new payment (Payment Initiation)
- GET status of entered/initiated payment
- DELETE deletion of the entered and not authorised payment
- POST generation of authorization ID
- Payment authorisation
  - GET Step I. payment authorisation detail
  - POST Step II. payment authorisation initiation - specific for each bank
  - PUT Step III. payment authorisation finalization - specific for each bank

3.2.3 Query for Balance Check (POST /my/payments/balanceCheck)

This is the resource for sending a request for balance check in a particular payer's payment account. This resource is authorized. Access to information must be granted by the client outside the interaction of this API before the resource is used.

Resource characteristics

URI:	/my/payments/balanceCheck
HTTP Method:	POST
Authorization:	request **requires** the authorization of user/client as part of API calling
Use certificate:	request **requires** the use of the qualified third-party certificate
Paging:	no
Sorting:	no
Filtering:	no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of POST request and response for calling, please see Chapter 3.2.3.1 MESSAGE ELEMENTS Query for balance check

Error codes defined for the POST service Query for balance check

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.

400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
403	AG01	[TransactionForbidden] – absent consent to access to balance check at the account.
400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.
400	FF01	[Invalid File Format] – invalid JSON forma tor other technical problem with the query processing.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RF01	[NotUniqueTransactionReference] – not unique request identifier.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.

3.2.3.1 MESSAGE ELEMENTS Query for Balance Check

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+	exchangeIdentification	[1..1]	Max18Text	Clear query identification
+	card	[0..1]	±	Transaction card
++	cardholderName	[0..1]	Max45Text	Card holder name
++	maskedPan	[1..1]	Max30Text	Masked card number
+	**Chyba! Nenalezen zdroj odkazů.**	[1..1]	±	Payer account
++	**Chyba! Nenalezen zdroj odkazů.**	[1..1]	±	Payer account identification
+++	**Chyba! Nenalezen zdroj odkazů.**	[1..1]	IBAN2007Identifier	IBAN
++	currency	[0..1]	CurrencyCode, ISO 4217	Payer account currency
+	authenticationMethod	[0..1]	CodeSet	Client verification

				method
+	merchant	[0..1]	±	Merchant executing the transaction
++	identification	[1..1]	Max35Text	Merchant identification
++	type	[0..1]	Code	Merchant type
++	shortName	[1..1]	Max35Text	Merchant name
++	commonName	[1..1]	Max70Text	Merchant name as stated in the payment receipt
++	address	[0..1]	Max140Text	Merchant address
++	countryCode	[0..1]	CountryCode, ISO 3166	Merchant country
++	merchantCategoryCode	[1..1]	Min3Max4Text, ISO 18245	Merchant code following the transaction type
+	transactionDetails	[1..1]	±	Transaction details
++	currency	[1..1]	CurrencyCode, ISO 4217	Balance query currency
++	totalAmount	[1..1]	Amount	Balance query amount

3.2.3.2 MESSAGE ELEMENTS Response for Balance Check

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+	responseIdentification	[1..1]	Number (integre)	Unique identification of response to query for Balance Check (from ASPSP).
+	exchangeIdentification	[1..1]	IntMax18Digits	Repeated identification of a payment transaction (query for Balance Check) from the issuer of the card to which the request for Balance Check linked to the account.

+	response	[1..1]	Code set	Result code of query for Balance Check.

Return codes for the parameter **„response“ – Code set**:

CODE	*DESCRIPTION*
APPR	Enough funds on this account
DECL	Unsufficient funds on this account

3.2.4 New payment - payment initiation (POST /my/payments)

Resource for entering a new payment.

Resource characteristics

URI: /my/payments
HTTP Method: POST
Authorization: request **requires** the authorization of user/client as part of API calling
Use certificate: request **requires** the use of the qualified third-party certificate
Paging: no
Sorting: no
Filtration: no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETER	*TYPE*	*MADATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of POST request and response for calling, please see Chapter 3.2.4.1 MESSAGE ELEMENTS New payment - payment initiation

Error codes defined for the service POST Query for Balance Check

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
403	AG01	[TransactionForbidden] – absent consent to access to Balance Check at the account.
400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.
400	FF01	[Invalid File Format] – invalid JSON forma tor other technical problem with the query processing.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RF01	[NotUniqueTransactionReference] – not unique request identifier.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.

3.2.4.1 MESSAGE ELEMENTS New payment – payment initiation

Payment types considered

PAYMENT CODE	*DESCRIPTION*
TUZEM	Domestic payment
SEPA	SEPA payment
EHP	Foreign payment within EEA
NONEHP	Foreign payment outside EEA

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*PAYMENT TYPE*	*FORMAT TYPE*	*PRESENTATION*
+	paymentIdentification	[1..1]	ALL	Max35Text	
++	instructionIdentification	[1..1]	ALL	Max35Text	
++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
++	transactionIdentification	[0..0]	ALL		
+	paymentTypeInformation	[0..1]	ALL	PymentType	
++	instructionPriority	[0..1]	ALL		
++	serviceLevel	[0..0]	ALL	ServiceType	
+++	code	[0..0]	ALL		
++	categoryPurpose	[0..0]	ALL	ExternalCategoryPurpose1Code	
+++	code	[0..0]	ALL	Max35Text	
+++	proprietary	[0..0]	ALL		
+	amount	[1..1]	ALL		
++	instructedAmount	[1..1]	ALL	Number	
+++	value	[1..1]	ALL	CurrencyCode	
+++	currency	[1..1]	ALL		
++	equivalentAmount	[0..0]	ALL		

+++	value	[0..0]	ALL		
+++	currency	[0..0]	ALL	ISODate	
+	requestedExecutionDate	[0..1]	ALL		
+	exchangeRateInformation	[0..0]	ALL	BaseOneRate	
++	exchangeRate	[0..0]	ALL	ExchangeRateType1Code	
++	rateType	[0..0]	ALL	Max35Text	
++	contractIdentification	[0..0]	ALL		
+	chargeBearer	[0..0] [0..0] [0..1] [0..1]	TUZEM SEPA EHP NONEHP		
+	chargesAccount	[0..0]	ALL		
++	identification	[0..0]	ALL		
+++	iban	[0..0]	ALL		
++	currency	[0..0]	ALL		
+	ultimateDebtor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	
++	name	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
++	postalAddress	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	
+++	streetName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max16Text	
+++	buildingNumber	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max16Text	
+++	postCode	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	

+++	townName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	CountryCode	
+++	country	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	
+++	addressLine	[0..0] [0..2] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
++	identification	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+++	organisationIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	BICIdentifier	
++++	bicOrBei	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
++++	other	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
+++++	identification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+++++	schemeName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
++++++	proprietary	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
+++++	issuer	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		

+++	privateIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
++++	other	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
+++++	identification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+++++	schemeName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
++++++	proprietary	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
+++++	issuer	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+	debtor	[0..0]	ALL		
++	name	[0..0]	ALL		
++	postalAddress	[0..0]	ALL		
+++	streetName	[0..0]	ALL		
+++	buildingNumber	[0..0]	ALL		
+++	postCode	[0..0]	ALL		
+++	townName	[0..0]	ALL		
+++	country	[0..0]	ALL		
+++	addressLine	[0..0]	ALL		
+	debtorAccount	[1..1]	ALL		
++	identification	[1..1]	ALL	IBAN2007Identifier	
+++	iban	[1..1]	ALL		
+++	other	[0..0]	ALL		
++++	identification	[0..0]	ALL	CurrencyCode	
++	currency	[0..1]	ALL		

+	intermediaryAgent1	[0..0]	ALL		
++	financialInstitutionIdentification	[0..0]	ALL		
+++	bic	[0..0]	ALL		
+++	clearingSystemMemberIdentification	[0..0]	ALL		
++++	clearingSystemIdentification	[0..0]	ALL		
+++++	code	[0..0]	ALL		
+++++	proprietary	[0..0]	ALL		
++++	memberIdentification	[0..0]	ALL		
+++	name	[0..0]	ALL		
+++	postalAddress	[0..0]	ALL		
++++	streetName	[0..0]	ALL		
++++	buildingNumber	[0..0]	ALL		
++++	postCode	[0..0]	ALL		
++++	townName	[0..0]	ALL		
++++	country	[0..0]	ALL		
++++	addressLine	[0..0]	ALL		
+++	other	[0..0]	ALL		
++++	identification	[0..0]	ALL		
+	creditorAgent	[0..0] [0..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP		
++	financialInstitutionIdentification	[0..0] [0..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	BICIdentifier	
+++	bic	[0..0] [1..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP		
+++	clearingSystemMemberIdentification	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP		

++++	clearingSystemIdentification	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	External Clearing System Identification1Code	
+++++	code	[0..0] [0..0] [0..0] [1..1]	TUZEM SEPA EHP NONEHP	Max35Text	
+++++	proprietary	[0..0] [0..0] [0..0] [1..1]	TUZEM SEPA EHP NONEHP	Max35Text	
++++	memberIdentification	[0..0] [0..0] [0..0] [1..1]	TUZEM SEPA EHP NONEHP	Max35Text	
+++	name	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max105Text	
+++	postalAddress	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max70Text	
++++	streetName	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max16Text	
++++	buildingNumber	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max16Text	
++++	postCode	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max35Text	
++++	townName	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	CountryCode	

++++	country	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max70Text	
++++	addressLine	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP		
+++	other	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max35Text	
++++	identification	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max140Text	
+	creditor	[0..0] [1..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	Max70Text Max70Text Max70Text Max35Text	
++	name	[0..0] [1..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	Max105Text	
++	postalAddress	[0..0] [0..1] [0..1] [1..1]	TUZEM SEPA EHP NONEHP	Max70Text	
+++	streetName	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	Max16Text	
+++	buildingNumber	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	Max16Text	
+++	postCode	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	Max35Text	
+++	townName	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	CountryCode	

+++	country	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	Max70Text Max70Text Max105Text Max105Text	
+++	addressLine	[0..0] [0..2] [0..2] [0..2]	TUZEM SEPA EHP NONEHP		
+	creditorAccount	[1..1]	ALL		
++	identification	[1..1]	ALL	IBAN2007Identifier	
+++	iban	[1..1]	ALL		
+++	other	[0..0] [0..0] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	Max35Text	
++++	identification	[0..0] [0..0] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	CurrencyCode	
++	currency	[0..1] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+	ultimateCreditor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	
++	name	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
++	postalAddress	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	
+++	streetName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max16Text	

+++	buildingNumber	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max16Text	
+++	postCode	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
+++	townName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	CountryCode	
+++	country	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	
+++	addressLine	[0..0] [0..2] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
++	identification	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+++	organisationIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	BICIdentifier	
++++	bicOrBei	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
++++	other	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
+++++	identification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+++++	schemeName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	

++++++	proprietary	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
+++++	issuer	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+++	privateIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
++++	other	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
+++++	identification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+++++	schemeName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
++++++	proprietary	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
+++++	issuer	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+	purpose	[0..0]	ALL	ExternalPurpose1Code	
++	code	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	
++	proprietary	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Airbank	
+	instructionForNextAgent	[0..0]	ALL	Algorithm for completing the variable	
+	remittanceInformation	[0..1]	ALL	Max140Tex	

				t	
++	unstructured	[0..1]	ALL		
++	structured	[0..1] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
+++	creditorReferenceInformation	[0..1] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		
++++	reference	[0..3] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP		

3.2.4.2 MESSAGE ELEMENTS Response for new payment – payment initiation

The table contains only the elements that appear only in the response message.

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+	transactionIdentification	[1..1]	Max35Text	Identifier of estabilished transaction
+	serviceLevel	[1..1]	-	Service placement (within type of payment)
++	code	[1..1]	Text	Type of entered payment
+	signInfo	[1..1]	-	Status information and id of unauthorized transactions
++	state	[1..1]	StateCode	Transaction authorization status
++	signId	[0..1]	Text	Identifier of the authorization process of a particular transaction.
++	signInfo	[1..1]	Status Code set	Transaction status identifier

Status codes of payment – StatusCode

HTTP STATUS CODE	*STATUS CODE*	*PURPOSE*
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not change

3.2.5 Status of entered/initiated payment (GET /payments/{paymentId}/status)

A resource to display the payment status. It is an entered payment which has not been authorised by the client yet or has already been authorised, and PISP requires its status (GET).

The resource returns the information only for transactions entered through a specific provider. Information on the provider is taken over from the certificate or licence information.

The user authorization of this resource is optional. Primarily, only a valid certificate of the provider is demanded

Resource characteristics

URI: /payments/{paymentId}/status
HTTP Method: GET
Authorization: request **does not require** the authorization of user/client as part of the API calling
Use certificate: request **requires** the use of the third-party qualified certificate
Paging: no
Sorting: no
Filtering: no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the

			precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.

Parameters of the response header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of POST for request and response calling, please see Chapter 3.2.5.1 MESSAGE ELEMENTS Status of entered/initiated payment

Error codes defined for the POST service Query for Balance Check

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	TRANSACTION_MISSING	Calling of the method which does not correspond to the licence, or invalid certificate.

3.2.5.1 MESSAGE ELEMENTS Status of entered/initiated payment

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*PAYMENT TYPE*	*FORMAT TYPE*	*PRESENTATION*
+	instructionStatus	[1..1]	PISP ALL	StatusCode	Status of entered payment

Status codes of payments – StatusCode

HTTP STATUS CODE	*STATUSCODE*	*PURPOSE*
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACSP	[AcceptedSettlementInProcess] - All preceding checks such as

		technical validation and customer pro⩽le were successful and therefore the payment initiation has been accepted for execution
200	ACSC	[AcceptedSettlementCompleted] - Settlement on the debtor´s account has been completed. Usage: this can be used by the ⩽rst agent to report to the debtor that the transaction has been completed. Warning: this status is provided for transaction status reasons, not for ⩽nancial information. It can only be used after bilateral agreement
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not change

3.2.6 Info on entered/initiated payment (GET /my/payments/{paymentId})

A resource to display the information on the entered payment. It is payment which is received for authorisation, but has not been authorised by the client yet. The resource only works with transactions entered **through a specific provider**.

The resource to find out the transaction detail. Information on the provider is taken from the certificate or information on licence.

Resource characteristics

URI: /my/payments/{paymentId}

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.

Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of POST request and response for calling, please see Chapter 3.2.6.1 MESSAGE ELEMENTS Info on entered/initiated payment

Error codes defined for the POST service Query for Balance Check

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Invalid/missing certificate = provider not authorised
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	Calling of the method which does not correspond to the licence, or invalid certificate.

3.2.6.1 MESSAGE ELEMENTS Info on entered/initiated payment

The result of the message is an overview of entered or already initiated payment. Therefore, the list of elements corresponds to the elements from resource New payment, see 3.2.4.1 MESSAGE ELEMENTS New payment - payment initiation.

3.2.7 Deleting the entered unauthorised payment (DELETE /my/payments/{paymentId})

A resource for deleting the unauthorised payment. Deleting is not conditioned by the transaction authorisation as it is not a payment received by the bank.

Resource characteristics

URI: /my/payments/{paymentId}

HTTP Method: DELETE

Authorization: request **requires** the authorization of user/client as part of the API calling
Use certificate: request **requires** the use of the third-party qualified certificate
Paging: no
Sorting: no
Filtering: no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETER	*TYPE*	*MABDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

Error codes defined for the POST service Query for Balance Check:

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	Calling of the method which does not correspond to the licence, or invalid certificate.

3.2.8 Generating the authorization ID (POST /my/payments/{paymentId}/sign)

Generating the authorization ID for payment before the initiation of the authorization process. The resource is implemented by banks that do not want to generate authorization ID for payment in the response when entering a new (unauthorized) payment through the resource POST /my/payments.

Each bank may define and describe in its documentation its own authorization scenarios. The CODE of authorization methods for these scenarios is used in the response of this resource.

The response to this resource is an authorization ID generated for the specific payment returned in the element signInfo.

Part of the response is also the set of scenarios that may contain one or more authorisation scenarios. If more scenarios are returned, it is up to the client to select what scenario to choose. The power of verifying each scenario should correspond to the power of other scenarios, or the scenarios are mutually replaceable.

Each scenario is defined as the sequence of codes of particular authorization methods. The code of authorization method is defined by the bank and should be described in its own API documentation outside the ČOBS standard.

Resource characteristics

URI:	/my/payments/{paymentId}/sign
HTTP method:	POST
Authorization:	request **requires** the authorization of user/client as part of the API calling
Use certificate:	request **requires** the use of the third-party qualified certificate
Paging:	no
Sorting:	no
Filtering:	no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	Ne	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the

			precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of POST request and response for calling, please see Chapter 3.2.8.1 MESSAGE ELEMENTS Generating the authorization ID

Error codes defined for the POST service Query for Balance Check:

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	Calling of the method which does not correspond to the licence, or invalid certificate.

3.2.8.1 MESSAGE ELEMENTS Generating the authorization ID

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*PAYMENT TYPE*	*FORMAT TYPE*	*PRESENTATION*
+	scenarios	[1..1]	PISP ALL	±	A set of possible authorization scenarios
+	signInfo	[1..1]	PISP ALL	±	Information on instruction authorization
++	state	[1..1]	PISP ALL	Text	Status of transaction authorization in a format supported by bank
++	signId	[1..1]	PISP ALL	Text	Unique identifier for current transaction authorization

3.2.9 Step I. Payment authorization detail (GET /my/payments/{paymentId}/sign/{signId})

An optional resource to identify the status and scenarios of payment authorization.

Part of the response is also the set of scenarios that may contain one or more authorisation scenarios. If more scenarios are returned, it is up to the client to select what scenario to choose. The power of verifying each scenario should correspond to the power of other scenarios, or the scenarios are mutually replaceable.

Each scenario is defined as the sequence of codes of particular authorization methods. Each bank may define and describe in its documentation its own authorization scenarios. CODE of authorization methods of these scenarios is used in the response of this resource.

Resource characteristics

URI:	/my/payments/{paymentId}/sign/{signId}
HTTP Method:	GET
Authorization:	request **requires** the authorization of user/client as part of the API calling
Use certificate:	request **requires** the use of the third-party qualified certificate
Paging:	no
Sorting:	no
Filtering:	no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETER	*TYPE*	*MANADATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of GET requests and response for calling, please see Chapter 3.2.9.1 MESSAGE ELEMENTS Step I. Payment authorization detail

Error codes defined for the POST service Query for Balance Check:

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method

3.2.9.1 MESSAGE ELEMENTS Step I. Payment authorization detail

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*PAYMENT TYPE*	*FORMAT TYPE*	*PRESENTATION*
+	scenarios	[1..1]	PISP ALL	±	A set of possible authorization scenarios
+	signInfo	[1..1]	PISP ALL	±	Information on instruction authorization
++	state	[1..1]	PISP ALL	Text	Status of transaction authorization in a format supported by the bank
++	signId	[1..1]	PISP ALL	Text	Unique identifier for current transaction authorization

3.2.10 Step II. Payment authorization initiation - specific for each bank (POST /my/payments/{paymentId}/sign/{signId})

This resource is designed to **start a specific authorization method** from the selected scenario.

The input is the JSON object containing the required type of authorization method - **CODE** and all elements specific for that step.

The output of this resource is an overview of values required for completing the authorization.

E.g., for CODE corresponding to the federated authorization, the response will be URL and parameters for redirecting to the federated authorization page.

And, for instance, for CODE corresponding to authorization through the OTP code sent by SMS, the response will only be the confirmation of sending the code. The sending as such is initiated by the bank.

Resource characteristics

URI:	/my/payments/{paymentId}/sign/{signId}
HTTP Method:	POST
Authorization:	request **requires** the authorization of user/client as part of the API calling
Use certificate:	request **requires** the use of the third-party qualified certificate
Paging:	no
Sorting:	no
Filtering:	no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of POST request and response for calling, please see chapter 3.2.10.1 MESSAGE ELEMENTS Step II. Payment authorization initiation - specific for each bank

Error codes defined for the POST service Query for Balance Check:

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method

3.2.10.1 MESSAGE ELEMENTS Step II. Payment authorization initiation – specific for each bank

Request parameters:

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*PAYMENT TYPE*	*FORMAT TYPE*	*PRESENTATION*
+	authorizationType	[1..1]	PISP ALL	Text	Code of required authorization (from authorization scenarios)

Response parameters:

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*PAYMENT TYPE*	*FORMAT TYPE*	*PRESENTATION*
+	authorizationType	[1..1]	PISP ALL	±	Code of required *authorization (from authorization scenarios)*
+	href	[0..1]	PISP ALL	±	Reference to call federated authorization
++	url	[1..1]	PISP ALL	Text	URL link or package of federated authorization
++	id	[0..1]	PISP ALL	Text	Possible id for calling federated authorization
+	method	[0..1]	PISP ALL	Text	Method to use href link and federated authorization.
+	formData	[0..1]	PISP ALL	±	Optional element. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains data for sending in redirecting to federated authorization.
++	SAMLRequest	[0..1]	PISP ALL	Text	Optional parameter. For the method

					POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains data of SAML request
++	relayState	[0..1]	PISP ALL	Text	Optional parameter. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains relayState for returnable value.
+	signInfo	[1..1]	PISP ALL	±	Information on instruction author
++	state	[1..1]	PISP ALL	Text	Status of transaction authorization in a format supported by the bank
++	signId	[1..1]	PISP ALL	Text	Unique identifier for current transaction authorization

3.2.11 Step III. Payment authorization initiation – specific for each bank (PUT /my/payments/{paymentId}/sign/{signId})

An optional resource designed for the finalization of the authorization process. The resource may be purely indicative or returns information on whether the authorization process was completed and with what result. E.g., in the case of federated authorization. Or what input, e.g. for entering the received OTP for the SMS method. Then the output is information on the authorization and also OTP verification.

Resource characteristics

URI: /my/payments/{paymentId}/sign/{signId}
HTTP Method: PUT
Authorization: request **requires** the authorization of user/client as part of the API calling
Use certificate: request **requires** the use of the third-party qualified certificate
Paging: no
Sorting: no
Filtering: no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case,

			application/json format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.

Parameters of the response header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of PUT request and response for calling, please see Chapter 3.2.11.1 MESSAGE ELEMENTS Step III. Payment authorization finalization - specific for each bank

Error codes defined for the POST service Query for Balance Check

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method

3.2.11.1 MESSAGE ELEMENTS Step III. Payment authorization finalization - specific for each bank

Request parameters

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*PAYMENT TYPE*	*FORMAT TYPE*	*PRESENTATION*
+	authorizationType	[1..1]	PISP ALL	Text	Code of required authorization (from authorization scenarios)

Request parameters

LEVEL	*MESSAGE*	*OCCURRENCE*	*PAYMENT*	*FORMAT TYPE*	*PRESENTATION*

	ELEMENT		TYPE		
+	state	[1..1]	PISP ALL	Text	Status of transaction authorization in a format supported by the bank

3.3 API Balance Check

3.3.1 How to read API Balance Check

The sender of query for API Balance Check must respect permitted characters mainly in the identifications and references that are sent out to the partner bank. Otherwise, the failure to respect them may lead to rejection. None of these elements may contain separate "/" (slash) at the beginning or end, or two successive slashes in the text.

The permitted character set is based only on the swift character set (that is, exclusively without diacritics - a different character set from the supported characters in CERTIS), i.e. they are the following characters:

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

/ - ? : ( ) . , ' +

Space

The standard is based on the version of XML message caa.001.001.05 and also freely assumes some elements of the XML message pain.001.001.03 due to higher flexibility of application.

Only one query can be sent and processed per call.

We recommend not completing the account currency „debtorAccount.currency“. In the case of differences of the currency filled in pain.001 against the actual currency which specifies an account, you will avoid any possible complications in the payer bank, which may reject the entire message in the event of different values (the payment currency „transactionDetails.currency“ must be always specified).

All bank account numbers of bank clients in the Czech Republic are defined according to Decree No. 169/2011 Coll. on the rules for the creation of account numbers for making payments. The data element „debtorAccount.identification.iban“ requires an account number in the IBAN format which is defined by the international standard ISO 13616.

Structure of the table of elements:

- **LEVEL** – the character plus „+“ determines the level of element embedding. The basic level is marked „+“ and each another is marked with another character, for instance, the level two is „++“
- **MESSAGE ELEMENT** –element name in the camel format
- **OCCURRENCE** – describes the occurrence of fields:

- [1…1] element is mandatory and only occurs once. In the case of a parent element, at least one embedded element must be filled in. If a mandatory element is embedded, the obligation does not pass to the parent element.
- [0…1] element is optional and occurs only once.
- [1…n] element is mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
- [0…n] element is not mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.

- **PAYMENT TYPE** – defines for which message types the current element is relevant
- **FORMAT TYPE** – defines the data format. It may be determined by the ISO 20022 standard valid for the item type PAIN.001 or CAAA.001 or another standard specifying the data structure (e.g., changes, dates, etc). Some items have a specific required format resulting from the payment system environment in the Czech Republic.
- **PRESENTATION** – contains a general role description

The format type defined as „±“ means the parent element which is then subdivided into other elements.

Conditions for element presence:

- When there is no OR but the parent element has more than 2 levels of elements, elements may be filled in cumulatively
- Parent element after [1…1] – at least one embedded element must be filled in
- Other see Sample description in the standard for XML pro pain.001, camt.053

3.3.2 List of API Balance Check resources

Specification of API Balance Check contains a description of resources for the acquisition of information on sufficient funds of the payer (client) at a specific financial institution.

An overview of resources:

- POST query for Balance Check

3.3.3 Query for Balance Check (POST /accounts/balanceCheck)

This is a resource for sending a query for Balance Check in a specific payment account of the account payer. The resource is not authorized directly by the account holder, through the authorization resource. Access to information must be granted by the client outside the interaction of this API before the resource is used.

A description of the process to grant access to the information is not part of this specification.

Resource characteristics

URI: /accounts/balanceCheck

HTTP Method: POST
Authorization: request **does not require** the authorization of user/client as part of the API calling
Use certificate: request **requires** the use of the third-party qualified certificate
Paging: no
Sorting: no
Filtering: no

Query parameters of the request: **not defined**

Parameters of the request header:

PARAMETR	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.
API-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as a communication configuration element.

Parameters of the response header:

PARAMETR	*TYPE*	*MANDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

The content of POST request and response for calling, please see Chapter 3.3.3.1 BASIC MESSAGE ELEMENTS Query for Balance Check

Return codes for the parameter **„response“**:

CODE	*DESCRIPTION*
APPR	Enough funds on this account
DECL	Unsufficient funds on this account

Error codes defined for the POST service Query for Balance Check

HTTP STATUS CODE	*ERROR CODE*	*PURPOSE*
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.

400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
403	AG01	[TransactionForbidden] – absent consent to access to Balance Check at the account.
400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.
400	FF01	[Invalid File Format] – invalid JSON forma tor other technical problem with the query processing.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RF01	[NotUniqueTransactionReference] – not unique request identifier.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.

3.3.3.1 BASIC MESSAGE ELEMENTS Query for Balance Check

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*PAYMENT TYPE*	*FORMAT TYPE*	*PRESENTATION*
+	exchangeIdentification	[1..1]	CISP	Max18Text	Clear query identification
+	card	[0..1]	CISP	±	Transaction card
++	cardholderName	[0..1]	CISP	Max45Text	Card holder name
++	maskedPan	[1..1]	CISP	Max30Text	Masked card number
+	**Chyba! Nenalezen zdroj odkazů.**	[1..1]	CISP	±	Payer account
++	**Chyba! Nenalezen zdroj odkazů.**	[1..1]	CISP	±	Payer account identification
+++	**Chyba! Nenalezen zdroj odkazů.**	[1..1]	CISP	IBAN2007Identifier	IBAN
++	currency	[0..1]	CISP	CurrencyCode,	Payer account

				ISO 4217	currency
+	authenticationMethod	[0..1]	CISP	CodeSet	Client verification method
+	merchant	[0..1]	CISP	±	Merchant executing the transaction
++	identification	[1..1]	CISP	Max35Text	Merchant identification
++	type	[0..1]	CISP	Code	Merchant type
++	shortName	[1..1]	CISP	Max35Text	Merchant name
++	commonName	[1..1]	CISP	Max70Text	Merchant name as stated in the payment receipt
++	address	[0..1]	CISP	Max140Text	Merchant address
++	countryCode	[0..1]	CISP	CountryCode, ISO 3166	Merchant country
++	merchantCategoryCode	[1..1]	CISP	Min3Max4Text, ISO 18245	Merchant code following the transaction type
+	transactionDetails	[1..1]	CISP	±	Transaction details
++	currency	[1..1]	CISP	CurrencyCode, ISO 4217	Balance query currency
++	totalAmount	[1..1]	CISP	Max18.5Amount	Balance query amount

3.3.3.2 MESSAGE ELEMENTS Response for Balance Check

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+	responseIdentification	[1..1]	Number (integre)	Unique identification of response to query for Balance Check (from ASPSP).
+	exchangeIdentification	[1..1]	IntMax18Digits	Repeated identification of a payment transaction (query for Balance Check) from the issuer of the card to which the request for Balance Check linked to the

				account.
+	response	[1..1]	Code set	Result code of query for Balance Check.

Return codes for the parameter **„response“ – Code set**:

CODE	*DESCRIPTION*
APPR	Enough funds on this account
DECL	Unsufficient funds on this account

4 Description of Elements of Messages for Services Payment Initiation, Account Information and Balance Check

4.1 Amount

JSON record: amount

Occurrence ALL: [1..1]

Definition: An amount of money that is to be transferred between the payer and the payee before deducting fees. It may be stated as only the Instructed Amount or the Equivalent Amount.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	InstructedAmount	[1..1]		Amount and currency in instruction
++	EquivalentAmount	[0..0]		Equivalent amount and currency

JSON example of element:

```
"amount": {
   "instructedAmount": {
      "value": 10050.15,
      "currency": "CZK"
}
```

4.1.1 InstructedAmount (Amount and currency in the instruction)

JSON record: amount.instructedAmount

Occurrence ALL: [1..1]

Definition: The amount to be transferred between the payer and the recipient before charges

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	value	[1..1]	Amount	Amount in instruction
+++	currency	[1..1]	Currency code ISO4217	Currency in instruction

Data type: Amount

Format: CurrencyAndAmount

4.1.1.1 Value (Amount of the transfer)

JSON record: amount.instructedAmount.value

Occurrence ALL: <1..1>

Definition: The amount to be transferred between the payer and the recipient before charges

Typ formátu: Amount

TUZEM: decimal places: 2, max. 1000000000000.00, min. 0.01

SEPA: decimal places: 2, max. 999999999.99, min. 0.01

EHP decimal places: 2, max. 999999999999999.99, min. 0.01

NONEHP: decimal places: 2, max. 999999999999999.99, min. 0.01

4.1.1.2 Currency (Měna převodu)

JSON record: amount.instructedAmount.currency

Occurrence ALL: <1..1>

Definition: The currency to which the amount in the instruction applies

Format: [A-Z]{3,3} - CurrencyCode, ISO 4217

TUZEM: CZK, or in another currency for payments within the bank, and only for the currency that the bank supports the payer.

SEPA: only for currency code EUR

EHP: only the currency the payer supports

NONEHP: only the currency the payer supports

4.1.2 EquivalentAmount (Equivalent amount and currency)

JSON record: amount.equivalentAmount

Occurrence *ALL:* <0..0>

Definition: The amount expressed in the currency of the payer's account, which is the equivalent of the amount to be transferred between the payer and the payee, before deducting the charges, expressed in the currency of the transfer. The element is only used for FX transactions.

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+++	value	[1..1]	Amount	Equivalent transaction amount
+++	currency	[1..1]	Currency code ISO4217	Currency equivalent to transaction amount

Data type: Amount

Format: CurrencyAndAmount

4.2 authenticationMethod (Authentication method)

***JSON record**:* authenticationMethod

***Occurrence**: [0…1]*

***Definition**: Verification of the card holder*

Application: The field is used if a card was used to verify sufficient account funds. It described the method and data that would be used for the transaction to verify the card holder.

***Format type**: CodeSet*

CODE	*NAME*	*DEFINITION*
NPIN	OnLinePIN	On-line PIN authentication (PersonalIdentification Number).
PPSG	PaperSignature	Handwritten paper signature.
PSWD	Password	Authentication by a password.
SCRT	SecureCertificate	Electronic commerce transaction secured with the X.509 certificate of a customer.
SCNL	SecuredChannel	Channel-encrypted transaction.
SNCT	SecureNoCertificate	Secure electronic transaction without cardholder certificate.
CPSG	SignatureCapture	Electronic signature capture (handwritten signature).
ADDB	BillingAddressVerification	Cardholder billing address verification.
BIOM	Biometry	Biometric authentication of the cardholder
CDHI	CardholderIdentificationData	Cardholder data provided for verification, for instance social security number, driver license number, passport number.
CRYP	CryptogramVerification	Verification of a cryptogram generated by a chip card or another device, for instance ARQC (Authorisation Request Cryptogram).
CSCV	CSCVerification	Verification of Card Security Code.
PSVE	PassiveAuthentication	Authentication based on statistical cardholder behaviour.
CSEC	SecureElectronicCommerce	Authentication performed during a secure electronic commerce transaction.
ADDS	ShippingAddressVerification	Cardholder shipping address verification.
TOKP	PaymentToken	Verification or authentication related to the use of a payment token, for instance the validation of the authorised use of a token.

JSON example of element:

```
"authenticationMethod": "NPIN",
```

4.3 bankTransactionCode (Code of bank transaction)

JSON record: bankTransactionCode

Occurrence: [1..1]

Definition: The code of bank transaction according to the code list of the Czech Banking Association assigned to a specific payment. Use: Each bank uses its own detailed code list to identify payments, which is, however, based on the 1st-3rd level of the code list of transactions according to the CBA standard for camt.053.

Transaction codes:

PAYMENTS	
Transaction code	**Description**
10000101000	Domestic Payment - Outgoing domestic payment
10000102000	Domestic Payment - Outgoing domestic payment - express
10000103000	Domestic Payment - Collection - Credit
10000104000	Domestic Payments - Invoice - Debit
10000105000	Domestic Payment System - Standing Order
10000106000	Domestic Payment System - SIPO
10000107000	Domestic Payment System - Incoming domestic payment
10000201000	Foreign Payments - Outbound Foreign Payment
10000202000	Foreign Payments - Incoming foreign payment
10000301000	Others - Others
10000401000	SEPA CT - Outgoing SEPA payment
10000402000	SEPA CT - Outgoing SEPA payment - express
10000403000	SEPA CT - Incoming SEPA payment
10000501000	SEPA DD - Incoming SEPA DD Core
10000502000	SEPA DD - Outgoing SEPA DD Core
10000503000	SEPA DD - Incoming SEPA DD B2B
10000504000	SEPA DD - Outgoing SEPA DD B2B

CASH OPERATIONS	
Transaction code	**Description**
20000101000	Deposit - Deposit without conversion in local currency
20000102000	Deposit - Deposit without conversion in foreign currency
20000103000	Deposit - Deposit with conversion
20000104000	Deposit - Deposit on account in another bank

20000201000	Selection - Choose without conversion in local currency
20000202000	Selection - Choose without conversion in a foreign currency
20000203000	Select - Choose with Conversion
20000301000	Other - Cash Processing
20000302000	Other - Other services

CARD OPERATIONS	
Transaction code	**Description**
30000101000	ATM - ATM withdrawals
30000102000	ATM - Choice of ATMs abroad
30000103000	ATM - Deposit
30000201000	POS - Card payment at the merchant
30000202000	POS - Cashback
30000203000	POS - Cash advance
30000301000	Other - Other services

CHARGES	
Transaction code	**Description**
40000101000	Domestic Payment - Outgoing Payment
40000102000	Domestic Payment - Incoming Payment
40000103000	Domestic Payment System - Standing Order
40000104000	Domestic Payment System - Other
40000201000	Foreign Payments - Outgoing Payment
40000202000	Foreign Payments - Incoming Payment
40000203000	Foreign Payments - Others
40000301000	SEPA - Outgoing SEPA payment
40000302000	SEPA - Incoming SEPA payment
40000303000	SEPA - SEPA DD B2C
40000304000	SEPA - SEPA DD B2B
40000401000	Cash operations - Deposit
40000402000	Cash Operations - Selection
40000403000	Cash operations - Other
40000501000	Card Operations - ATM
40000502000	Card Operations - POS
40000503000	Card Operations - Others
40000601000	Others - Loans
40000602000	Others - Checks
40000603000	Others - Trade finance

40000604000	Others - Treasury
40000605000	Other - Other services

R-TRANSACTIONS	
Transaction code	**Description**
50000201001	SEPA DD - SEPA DD B2C - Return
50000201002	SEPA DD - SEPA DD B2C - Refund
50000201003	SEPA DD - SEPA DD B2C - Reversal
50000201004	SEPA DD - SEPA DD B2C - Reject
50000201005	SEPA DD - SEPA DD B2C - Refusal
50000201006	SEPA DD - SEPA DD B2C - Revocation
50000201007	SEPA DD - SEPA DD B2C - Request for cancellation
50000202001	SEPA DD - SEPA DD B2B - Return
50000202002	SEPA DD - SEPA DD B2B - Refund
50000202003	SEPA DD - SEPA DD B2B - Reversal
50000202004	SEPA DD - SEPA DD B2B - Reject
50000202005	SEPA DD - SEPA DD B2B - Refusal
50000202006	SEPA DD - SEPA DD B2B - Revocation
50000202007	SEPA DD - SEPA DD B2B - Request for cancellation

OTHERS	
Transaction code	**Description**
90000101000	Loans
90000201000	Interest
90000301000	Checks
90000401000	Trade finance
90000501000	Treasury
90000601000	Bonuses
90000701000	Other services

JSON example of element:

```
"bankTransactionCode": {
   "proprietary": {
      "code": 4000010,
      "issuer": "CBA"
}
```

4.4 bookingDate (Date of processing)

JSON record: bookingDate

Occurrence: [1..1]

Definition: Date of payment processing/posting by the bank.

Application: Depends on the transaction type and the method how the bank presents data (and time) of payment processing/posting

Type: This message element consists of the following elements:

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
++	date	[1..1]	ISODate in format YYYY-MM-DD 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)	Date of payment processing/posting by the bank

JSON example of element:

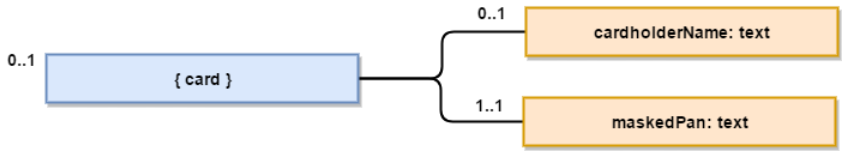
```
    "bookingDate": {
       "date": "2016-09-05T00:00:00+01:00"
    }
```

4.5 card (Card)

***JSON record**: card*

Occurrence: [0…1]

Definition: Payment card linked to the transaction, if the card was used for initiation



Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
++	cardholderName	[0..1]	CISP	Max45Text	Card holder name
++	maskedPan	[1..1]	CISP	Max30Text	Masked card number

JSON example of element:

```
"card": {
  "cardHolderName": "Jan Novák",
  "maskedPAN": "1234**********6789"
}
```

4.5.1 cardholderName (Card holder name)

***JSON record**: card.cardholderName*

***Occurrence**: [0…1]*

***Definition**: Card holder name*

***Application**: The field contains the name of card holder stated on the card.*

***Format type**: Max45Text*

JSON example of element:

```
"cardHolderName": "Jan Novák"
```

4.5.2 maskedPan (Masked card number)

JSON record: card.maskedPan

Occurrence: [1...1]

Definition: Masked card number

Application: The field contains a partially masked card number, which is stated in the payment receipt or is displayed to the card holder after verifying the balance. Masked numbers may be replaced by spaces or the character „*“.

***Format type**: Max30Text*

JSON example of element:

```
"maskedPAN": "1234**********6789"
```

4.6 creditDebitIndicator (Indication of debit/credit payment)

***JSON record**:* creditDebitIndicator

Occurrence: [1..1]

Definition: Indication of whether this is a debit payment or credit payment in the account.

Application: The indication of whether this is a debit or credit payment is marked with one of the following codes:

Format type: CreditDebitCode

CODE	DESCRIPTION
DBIT	To the debit of
CRDT	In other cases

JSON example of element:

```
"creditDebitIndicator": "CRDT"
```

4.7 creditor (Payee)

JSON record: creditor

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [1..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: The party to which a financial amount is paid.

Type: This message element contains the following elements **PartyIdentification32CZ2**:

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	Name	Max70Text	Payer name
++	Postal Address		Payer postal address

JSON example of element:

```
"creditor": {
   "name": "1. wiena investment",
   "postalAddress": {
      "streetName": "Reisnerstraße",
      "buildingNumber": "20",
      "postCode": "1030",
      "townName": "Wiena",
      "country": "AT"
   }
}
```

4.7.1 name (Name)

JSON record: creditor.name

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [1..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: Name under which the party is known, and which is normally used to identify the party.

Data type: Max70Text

JSON example of element:

```
"name": "1. wiena investment"
```

4.7.2 postalAddress (Payee postal address)

JSON record: *creditor.postalAddress*

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: *Information that localizes and identifies the specific address such as the postal address.*

Type: *This message element contains the following elements* ***PostalAddress6CZ***

JSON example of element:

```
"postalAddress": {
   "streetName": "Reisnerstraße",
   "buildingNumber": "20",
   "postCode": "1030",
   "townName": "Wiena",
```

```
        "country": "AT"
    }
```

4.7.2.1 streetName (Street)

JSON record: creditor.postalAddress.streetName

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Street name or section.

Data type: Max70Text

JSON example of element:

```
    "streetName": "Reisnerstraße",
```

4.7.2.2 buildingNumber (Building number)

JSON record: creditor.postalAddress.buildingNumber

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Number that identifies the position of building in the street.

Data type: Max16Text

JSON example of element:

```
    "buildingNumber": "20",
```

```
```

4.7.2.3 postCode (Postcode)

JSON record: creditor.postalAddress.postCode

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

JSON example of element:

```
"postCode": "1030"
```

4.7.2.4 townName (Town)

JSON record: creditor.postalAddress.townName

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "Wiena",
```

4.7.2.5 country (Country)

JSON record: creditor.postalAddress.country

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: A country with its own government

Data type: CountryCode

Data format: [A-Z]{2,2}

Rule: Country code is checked against the list of country names under the ISO 3166.

JSON example of element:

```
"country": "AT"
```

4.7.2.6 addressLine (Unstructured address record)

JSON record: creditor.postalAddress.addressLine

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..2]

EHP (Foreign payments within EEA): [0..2]

NONEHP (Foreign payments outside EEA): [0..2]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.8 creditorAccount (Payee account)

JSON record: creditorAccount

Occurrence All: [1. .1]

Definition: Clear identification of the payee account to which the credit item will be transferred as the transaction result.

Type: This message element consists of the following elements **CashAccount16CZ**:

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	identification		Identification
++	currency		Currency

JSON example of element:

```
"creditorAccount": {
   "identification": {
      "iban": "CZ3908000000000204533335",
      "other": {
         "identification": "123/0800"
      }
   },
   "currency": "CZK"
},
```

4.8.1 identification (Identification)

JSON record: creditorAccount.identification

Occurrence All: [1. .1]

Definition: A clear and unambiguous account identification between the account holder and the party administering the account.

Type: This message element consists of the following elements:

TUZEM (Domestic payments): **AccountIdentification4ChoiceCZ**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	IBAN	[1..1]	IBAN2007Identifier	Account no. in the IBAN format
+++	Other	[0..0]		Other format of account

				number

SEPA (SEPA payments): ***AccountIdentification4CZ***

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+++	IBAN	[1..1]	IBAN2007Identifier	Account no. in the IBAN format

EHP (Foreign payments within EEA): ***AccountIdentification4ChoiceCZ***

NONEHP (Foreign payments outside EEA): ***AccountIdentification4ChoiceCZ***

LEVEL	*OR*	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+++	{Or	IBAN	[1..1]	IBAN2007Identifier	Account no. in the IBAN format
+++	Or}	Other	[1..1]		Other format of account number

JSON example of element:

```
"identification": {
   "iban": "CZ3908000000000204533335",
   "other": {
      "identification": "123/0800"
   }
}
```

4.8.1.1 IBAN (Account number in the IBAN format)

JSON record: creditorAccount.identification.iban

Occurrence All: [1. .1]

Definition: The international account number format used by financial institutions for clear and unambiguous client account identification. A more detailed specification of the format and content IBAN is available in the standard ISO 13616.

Data type:

IBAN2007Identifier Format: [A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}, valid IBAN consists of all the following components: country code, check digit and account no. in local BBAN format.

Account no. in IBAN format, which is issued by the Czech bank has exactly 24 alphanumeric characters.

JSON example of element:

```
"iban": "CZ3908000000000204533335"
```

4.8.1.2 other (Other account number format)

JSON record: creditorAccount.identification.other

Occurrence only EHP (Foreign payments within EEA) and NONEHP (Foreign payments outside EEA): [1. .1]

Definition: Unique account identification assigned by the party administering the account, using the identification scheme.

Type: This message element consists of the following elements **GenericAccountIdentification1CZ**:

LEVEL	*MESSAGE ELEMENT*	*FORMAT TYPE*	*PRESENTATION*
++++	identification	Max34Text	Account no. in local format

JSON example of element:

```
"other": {
   "identification": "123/0800"
}
```

4.8.1.2.1 identification (Account number in local BBAN format)

JSON record: creditorAccount.identification.other.identification

Occurrence only EHP (Foreign payments within EEA) and NONEHP (Foreign payments outside EEA): [1. .1]

Definition: An account number used by financial institutions for clear and unambiguous identification of the client account. The account number is issued by the client's bank. The account number in the local BBAN format which is issued by the Czech bank has maximum 16 numeric characters, minimum 2 numeric characters. It consists of an antenumber, which has minimum 2 numeric characters, maximum 6 numeric characters, and of a basic form of the account no. which has minimum 2 numeric characters, maximum 10 numeric characters.

Data type: Max34Text

JSON example of element:

```
"identification": "123/0800"
```

4.8.2 currency (Currency)

JSON record: creditorAccount.currency

Occurrence only TUZEM (Domestic payments): [0..1]

Definition: Identification of the currency in which the payer account is administered. If the currency is stated, it must correspond to the administration, otherwise it is rejected.

Data type: CurrencyCode according to ISO 4217

Format: [A-Z]{3,3}

JSON example of element:

```
"currency": "CZK"
```

4.9 creditorAgent (Payee bank)

JSON record: creditorAgent

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: Financial institution that administers the payee account.

Type: This message element consists of the following elements
BranchAndFinancialInstitutionIdentification4CZ:

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION

++	Financial Institution Identification		Financial Institution Identification

JSON example of element:

```
"creditorAgent": {
   "financialInstitutionIdentification": {
      "bic": "ABNYUS33",
      "name": "New York Commercial Bank",
      "postalAddress": {
         "streetName": "Merrick Avenue615",
         "buildingNumber": "61511590-6644",
         "postCode": "11590-6644",
         "townName": "WESTBURY, NY",
         "country": "USA"
      }
   }
}
```

4.9.1 financialInstitutionIdentification (Financial institution identification)

JSON record: creditorAgent.financialInstitutionIdentification

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: A unique and unambiguous financial institution identification assigned by the international standard.

Type: This message element consists of the following elements **FinancialInstitutionIdentification7CZ**

JSON example of element:

```
"financialInstitutionIdentification": {
```

```
      "bic": "ABNYUS33",
      "clearingSystemMemberIdentification": {
         "memberIdentification": "2700"
      },
      "name": "New York Commercial Bank",
      "postalAddress": {
         "streetName": "Merrick Avenue 615",
         "buildingNumber": "61511590-6644",
         "postCode": "11590-6644",
         "townName": "WESTBURY, NY",
         "country": "USA"
      }
   }
```

4.9.1.1 BIC (BIC / SWIFT bank code)

JSON record: creditorAgent.financialInstitutionIdentification.bic

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [1..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Bank Identifier Code. A code assigned to financial institutions by the Registration authority according to the international identification scheme as described in the latest version of the ISO 9362. The bank code in the format of BIC / SWIFT code has exactly 8 or exactly 11 alphanumeric characters.

Data type: BICIdentifier

JSON example of element:

```
   "bic": "ABNYUS33"
```

4.9.1.2 clearingSystemMemberIdentification (Identification of Clearing system participant)

JSON record: creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Identification of participant in the local clearing system

Type: This message element consists of the following elements **ClearingSystemMemberIdentification2**

JSON example of element:

```
"clearingSystemMemberIdentification": {
   "memberIdentification": "2700"
}
```

4.9.1.2.1 clearingSystemIdentification (Clearing system identification)

JSON record:
creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Clearing system identification

Type: This message element consists of the following elements **ClearingSystemIdentification2Choice**

4.9.1.2.1.1 code (Code)

JSON record:
creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification.code

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: A code identifying the local clearing system as stated in the external code list.

Data type: ExternalClearingSystemIdentification1Code

4.9.1.2.1.2 proprietary (Free format)

JSON record:
creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification.proprietary

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Identification of the local clearing system, in free format.

Data type: Max35Text

4.9.1.2.2 memberIdentification (Participant's identification code – clearing code)

JSON record:
creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification.memberIdentification

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Participant‘s identification code or bank’s clearing code.

Data type: Max35Text

JSON example of element:

```
"memberIdentification": "2700"
```

4.9.1.3 name (Name)

JSON record: creditorAgent.financialInstitutionIdentification.name

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Name under which the party is known, and which is normally used to identify that party.

Data type: Max70Text

JSON example of element:

```
"name": "New York Commercial Bank",
```

4.9.1.4 postalAddress (Postal address)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Information that localizes and identifies the specific address as a postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**

For foreign payments, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two address lines from the Address Line field. *Typically, the first line shows the street and the Land registry no., the city and city's postal code in the second line.*

JSON example of element:

```
"postalAddress": {
   "streetName": "Merrick Avenue 615",
   "buildingNumber": "61511590-6644",
   "postCode": "11590-6644",
   "townName": "WESTBURY, NY",
   "country": "USA"
}
```

4.9.1.4.1 streetName (Street)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.streetName

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Street name or section.

Data type: Max70Text

JSON example of element:

```
"streetName": "Merrick Avenue 615"
```

4.9.1.4.2 buildingNumber (Building no.)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.buildingNumber

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Number that identifies the position of a building in the street.

Data type: Max16Text

JSON example of element:

```
"buildingNumber": "61511590-6644"
```

4.9.1.4.3 postCode (Postcode)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.postCode

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

JSON example of element:

```
"postCode": "11590-6644"
```

4.9.1.4.4 townName (Town)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.townName

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "WESTBURY, NY"
```

4.9.1.4.5 country (Country)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.country

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: A country with its own government

Data type: CountryCode

JSON example of element:

```
"country": "USA"
```

4.9.1.4.6 addressLine (Unstructured address record)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.addressLine

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.9.1.5 other (Another bank identification)

JSON record: creditorAgent.financialInstitutionIdentification.other

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Unique agent identification assigned to the institution using the identification scheme

Type: This message element consists of the following elements **GenericFinancialIdentification1CZ**

JSON example of element:

```
"other": {
   "identification": "123/0800"
}
```

4.9.1.5.1 identification (Local bank code)

JSON record: creditorAgent.financialInstitutionIdentification.other.identification

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Local format of the bank code

Data type: Max35Text

JSON example of element:

```
"identification": "123/0800"
```

4.10 debtor (Payer)

JSON record: debtor

Occurrence All: [0..0]

Definition: A party that owes money to the payee.

Type: This message element consists of the following elements **PartyIdentification32CZ2**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	Name	[0..0]	Max70Text	Payer name
++	Postal Address	[0..0]		Payer postal address

JSON example of element:

```
"debtor": {
   "name": "RENWORTH s.r.o",
   "postalAddress": {
         "streetName": "Merrick Avenue 615",
         "buildingNumber": "61511590-6644",
         "postCode": "11590-6644",
         "townName": "WESTBURY, NY",
         "country": "USA"
   }
},
```

4.10.1 name (Name)

JSON record: debtor.name

Occurrence All: [0..0]

Definition: Name under which the party is known, and which is normally used to identify that party.

Data type: Max70Text

Payer name is not normally filled in, unless agreed otherwise with the payer bank.

JSON example of element:

```
"name": "RENWORTH s.r.o"
```

```
```

4.10.2 postalAddress (Postal address of the original payer)

JSON record: debtor.postalAddress

Occurrence All: [0..0]

Definition: Information that localizes and identifies the specific address as a postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**.

The payer address is not normally filled in, unless otherwise agreed with the payer bank. In this case, you can fill it in for domestic payments in a structured or unstructured format. If you use unstructured format, it is recommended to fill in two repeated Address Lines. Typically, the first line shows the street and the Land registry no., and the second line the town and the postal code. For SEPA payments and foreign payments, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two lines of address from the Address Line field. Typically, the first line shows the street and the Land registry no., and the second line the town and the postal code.

JSON example of element:

```
"postalAddress": {
   "streetName": "Merrick Avenue 615",
   "buildingNumber": "61511590-6644",
   "postCode": "11590-6644",
   "townName": "WESTBURY, NY",
   "country": "USA"
}
```

4.10.2.1 streetName (Street)

JSON record: debtor.postalAddress.streetName

Occurrence All: [0..0]

Definition: Street name or section.

Data type: Max70Text

JSON example of element:

```
"streetName": "Merrick Avenue 615"
```

```

```

4.10.2.2 buildingNumber (Building number)

JSON record: debtor.postalAddress.buildingNumber

Occurrence All: [0..0]

Definition: Number that identifies the position of building in the street.

Data type: Max16Text

JSON example of element:

```
"buildingNumber": "61511590-6644"
```

4.10.2.3 postCode (Postcode)

***JSON record**: debtor.postalAddress.postCode*

***Occurrence** All: [0..0]*

***Definition**:* Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

***Data type**: Max16Text*

JSON example of element:

```
"postCode": "11590-6644"
```

4.10.2.4 townName (Town)

JSON record: debtor.postalAddress.townName

Occurrence All: [0..0]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "WESTBURY, NY"
```

4.10.2.5 country (Country)

JSON record: debtor.postalAddress.country

Occurrence All: [0..0]

Definition: A country with its own government

Data type: CountryCode

Data format: [A-Z]{2,2}

Rule: Country code is checked against the list of country names under the ISO 3166.

JSON example of element:

```
"country": "USA"
```

4.10.2.6 addressLine (Unstructured address record)

JSON record: debtor.postalAddress.addressLine

Occurrence All: [0..0]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

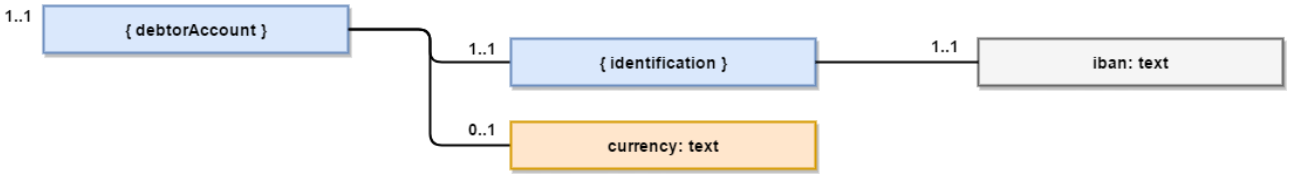
Maximum 2 lines with 70 characters per line

4.11 debtorAccount (Payer account)

JSON record: debtorAccount

***Occurrence**: [1…1]*

***Definition**:* Clear identification of the payer account in which a debit item will be executed as the transaction result.



Type: This message element consists of the following elements **CashAccount16CZ**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
++	**Chyba! Nenalezen zdroj odkazů.**	[1..1]	CISP	±	Payer account identification
++	currency	[0..1]	CISP	CurrencyCode, ISO 4217	Payer account currency

JSON example of element:

```
"debtorAccount":    {
  "identification": {
    "iban": "CZ0708000000001019382023"
  },
   "currency": "CZK"
}
```

4.11.1 identification (Identification)

***JSON record**: debtorAccount.identification*

***Occurrence**: [1…1]*

Definition: Identification of payer account in which the Balance Check are verified. Only one type of identification is possible. Other formats are not permitted.

***Type**: This message element consists of the following elements **AccountIdentification4ChoiceCZ**:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	IBAN	[1..1]	IBAN2007IdentifierCZ	Account no. in the IBAN format

+++	Other	[0..0]		Other format of account number

JSON example of element:

```
"identification": {
   "iban": "CZ0827000000002108589434",
      "other": {
         "identification": "000000-2108589434"
      }
}
```

4.11.1.1 IBAN (Account number in the IBAN format)

JSON record: debtorAccount.identification.iban

Occurrence: [1…1]

Definition: Payer account number in which Balance Check are verified, in the IBAN format

Format type: ISO 13616

Has the following structure:

- *2 characters (position 1-2) – country code „CZ“*
- *2 characters (position 3-4) – check digit for module 98 – allows programme check of the number – protection against wrong entered account no. (e.g., resulting from any typing error)*
- *4 characters (position 5-8) – payment system code – code assigned to the bank in the „Code list of payment system in the Czech Republic maintained by the CNB*
- *16 characters (position 9-24) – account no. in the format according to the Decree No. 169/2011 Coll.*
- *Maximum length 24 characters*

[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}, valid IBAN consists of all the following components: country code, check digit and account no. in local BBAN format.

Other formats for account identification are not permitted.

***Data type**: IBAN2007Identifier*

JSON example of element:

```
"iban": "CZ0708000000001019382023"
```

4.11.1.2 other (Other account number format)

JSON record: debtorAccount.identification.other

Occurrence All: [0..0]

Definition: Unique account identification assigned by the party administering the account, using the identification scheme.

Type: This message element consists of the following elements **GenericAccountIdentification1CZ**:

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
++++	Identification	[0..0]		Account no. in local format

JSON example of element:

```
"other": {
   "identification": "000000-2108589434"
}
```

4.11.1.2.1 identification (Account number in local BBAN format)

JSON record: debtorAccount.identification.other.identification

Occurrence All: [0..0]

Definition: An account number used by financial institutions for clear and unambiguous identification of the client account. The account number is issued by the client‘s bank. The account number in the local BBAN format which is issued by the Czech bank has maximum 16 numeric characters, minimum 2 numeric characters. It consists of an antenumber, which has minimum 2 numeric characters, maximum 6 numeric characters, and of a basic form of the account no. which has minimum 2 numeric characters, maximum 10 numeric characters.

Data type: Max16Num

JSON example of element:

```
"identification": "000000-2108589434"
```

4.11.2 currency (Currency)

JSON record: debtorAccount.currency

Occurrence: [0...1]

Definition: Currency of the payer account in which Balance Check are verified

Application: The field defines the account currency in case that the debit account is multi-currency.

Format type: CurrencyCode, code under the ISO-4217, [A-Z]{3,3}

JSON example of element:

```
"currency": "CZK"
```

4.12 SEPA identification (Identification)

For SEPA payments, the payer, the payee, the original payer and the final payee the client has the option to pass the identification based on whether it is the organization identification or identification of a natural person.

The payee demands this information as the identification of the (original) payer and the identification of the (final) payee:

- Identification type – Organization / Natural person
- Detailed information
  - Either BIC / SWIFT code
  - Or Other identification with these data:
    - Document type
    - Document issuer

JSON record: identification

Occurrence only SEPA payments: [0..1]

Definition: A unique and unambiguous identification of the party.

Type: This message element consists of the following elements **Party6Choice**:

LEVEL	OR	MESSAGE ELEMENT	OCCURRENCE	PRESENTATION
+++	{Or	Organisation Identification	[1..1]	Organization identification
+++	Or}	Private Identification	[1..1]	Private individual identification

JSON example of element privateIdentification:

```
"identification": {
   "privateIdentification": {
      "other": {
         "identification": "12356879131",
         "schemeName": {
            "proprietary": "passport",
            "issuer": "WIENA"
         }
      }
   }
}
```

JSON example of element organisationIdentification:

```
"identification": {
   "organisationIdentification": {
      "other": {
         "identification": "48135283",
         "schemeName": {
            "code": "1.2.203.48135283",
            "proprietary": "RENWORTH s.r.o"
         }
      }
   }
}
```

4.12.1 organisationIdentification (Organization identification)

JSON record: identification.organisationIdentification

Occurrence only SEPA payments: [1..1]

Definition: A unique and irreplaceable method for identifying an organization.

Type: This message element consists of the following elements **OrganisationIdentification4CZ**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION

++++	BIC Or BEI	[0..1]	BICIdentifier	BIC / SWIFT code
++++	Other	[0..1]		Other identification

JSON example of element:

```
"organisationIdentification": {
   "other": {
      "identification": "48135283",
      "schemeName": {
         "proprietary": "RENWORTH s.r.o"
       }
    }
 }
```

4.12.1.1 BIC Or BEI (BIC / SWIFT code)

JSON record: identification.organisationIdentification.bic

Occurrence *only SEPA payments: [0..1]*

Definition: Bank Identifier Code. A code assigned to financial institutions by the Registration authority according to the international identification scheme as described in the latest version of the ISO 9362. An 8 or 11-digit bank code in the format of BIC / SWIFT code.

Data type: BICIdentifier

Format: [A-Z]{6,6}[A-Z2-9][A-NP-Z0-9]([A-Z0-9]{3,3}){0,1}

Rule: Valid BICs are registered at the ISO 9362 Registration authority and consist of eight (8) or eleven (11) successive characters containing the first three or all four of the following components: bank code, country code and location code are mandatory, while the branch code is optional.

JSON example of element:

```
"bic": "ABNYUS33"
```

4.12.1.2 other (Other identification)

JSON record: identification.organisationIdentification.other

Occurrence *only SEPA payments: [0..1]*

Definition: A unique identification of the organization assigned by the institution using an identification scheme.

Type*: This message element consists of the following elements* ***GenericOrganisationIdentification1****:*

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+++++	Identification	[1..1]	Max35Text	Identification data
+++++	Scheme Name	[0..1]		Document type
+++++	Issuer	[0..1]	Max35Text	Document issuer

JSON example of element:

```
"other": {
   "identification": "48135283",
   "schemeName": {
      "proprietary": "RENWORTH s.r.o"
    }
 }
```

4.12.1.2.1 identification (Identification data)

JSON record: identification.organisationIdentification.other.identification

Occurrence only SEPA payments: [1..1]

Definition: Identification assigned by the institution issuing the document.

Data type: Max35Text

JSON example of element:

```
"identification": "48135283"
```

4.12.1.2.2 schemeName (Document type)

JSON record: identification.organisationIdentification.other.identification.schemeName

Occurrence *only SEPA payments: [0..1]*

Definition*: ID document name.*

***Type**: This message element consists of the following elements **OrganisationIdentificationSchemeName1CZ**:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++++	Proprietary	[1..1]	Max35Text	Free format

JSON example of element:

```
"schemeName": {
   "proprietary": "RENWORTH s.r.o"
}
```

4.12.1.2.2.1 proprietary (Free format)

JSON record: identification.organisationIdentification.other.identification.schemeName

***Occurrence** only SEPA payments: [1..1]*

***Definition**: Name of document type, in the free text format.*

***Data type**: Max35Text*

JSON example of element:

```
"proprietary": "RENWORTH s.r.o"
```

4.12.1.2.2.2 issuer (Document issuer)

JSON record: identification.organisationIdentification.other.identification.issuer

***Occurrence** only SEPA payments: [0..1]*

***Definition**: Name of document issuer, i.e. the entity assigning the identification.*

***Data type**: Max35Text*

JSON example of element:

```
"issuer": "WIENA"
```

4.12.2 privateIdentification (Private individual identification)

JSON record: identification.privateIdentification

Occurrence *only SEPA payments: [1..1]*

Definition*: A unique and irreplaceable method for identifying a natural person.*

Type*: This message element consists of the following elements* ***PersonIdentification5CZ****:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++	Other	[0..1]	Max35Text	Other identification

JSON example of element:

```
"privateIdentification": {
   "other": {
      "identification": "12356879131",
      "schemeName": {
         "proprietary": "passport",
         "issuer": "WIENA"
      }
   }
}
```

4.12.2.1 other (Other identification)

JSON record: identification.privateIdentification.other

Occurrence *only SEPA payments: [0..1]*

Definition*: A unique identification of the person assigned by the institution using an identification scheme.*

Type*: This message element consists of the following elements* ***GenericPersonIdentification1***

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	Identification	[1..1]	Max35Text	Identification data
+++++	Scheme Name	[0..1]		Document type
+++++	Issuer	[0..1]	Max35Text	Document issuer

JSON example of element:

```
    "other": {
       "identification": "12356879131",
       "schemeName": {
          "proprietary": "passport",
          "issuer": "WIENA"
       }
    }
```

4.12.2.1.1 identification (Identification data)

JSON record: identification.privateIdentification.other.identification

Occurrence *only SEPA payments: [1..1]*

Definition*: A unique and irreplaceable method for identifying a person.*

Data type*: Max35Text*

JSON example of element:

```
    "identification": "12356879131",
```

4.12.2.1.2 schemeName (Document type)

JSON record: identification.privateIdentification.other.schemeName

Occurrence *only SEPA payments: [0..1]*

Definition*: Name of the identification document.*

Type*: This message element consists of the following elements* ***PersonIdentificationSchemeName1Choice****:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++++	Proprietary	[1..1]	Max35Text	Free format

JSON example of element:

```
    "schemeName": {
       "proprietary": "passport",
       "issuer": "WIENA"
```

```
    }
```

4.12.2.1.2.1 proprietary (Free format)

JSON record: identification.privateIdentification.other.schemeName.proprietary

Occurrence *only SEPA payments: [1..1]*

Definition*: Name of the document type, in the free text format.*

Data type*: Max35Text*

JSON example of element:

```
    "proprietary": "passport"
```

4.12.2.1.2.2 issuer (Document issuer)

JSON record: identification.privateIdentification.other.schemeName.issuer

Occurrence *only SEPA payments: [0..1]*

Definition*: Name of the document issuer, i.e. the entity assigning the identification.*

Data type*: Max35Text*

JSON example of element:

```
    "issuer": "WIENA"
```

4.13 entryDetails (Entry details)

JSON record: entryDetails

Occurrence: [0..n]

Definition: Entry details.

Application: This level only repeats once for the relevant item

Type*: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION

++	transactionDetails	[0..n]	-	Payment detail

JSON example of element:

```
   "entryDetails": {
      "transactionDetails": {
         "amountDetails": {
            "instructedAmount": {
               "amount": {
                  "value": 122.22,
                  "currency": "CZK"
               }
            }
         },
             "additionalTransactionInformation": "PŘIPSÁNÍ ÚROKU ZE
ZUSTATKU"
       }
    }
```

4.13.1 transactionDetails (Payment details)

JSON record: entryDetails.transactionDetails

Occurrence: [0..n]

Definition: Payment details.

Application: This level only repeats once for the relevant item.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	references	[0..1]	-	Set of references
+++	amountDetails	[0..1]		Payment details
+++	charges	[0..n]		Information on fees
+++	relatedParties	[0..1]		Information on payer
+++	relatedAgents	[1..1]		Information on

				payer bank
+++	purpose	[0..1]		
+++	remittanceInformation	[0..1]		
+++	additionalTransactionInformation	[0..1]		Additional information

JSON example of element:

```
"transactionDetails": {
    "amountDetails": {
       "instructedAmount": {
          "amount": {
             "value": 122.22,
             "currency": "CZK"
          }
       }
    },
    "additionalTransactionInformation": "PŘIPSÁNÍ ÚROKU ZE ZUSTATKU"
}
```

4.13.1.1 references (Reference)

JSON entry: entryDetails.transactionDetails.references

Occurrence: [0..1]

Definition: Set of references unambiguously identifying the payment.

Application: According to technical possibilities of the bank, it contains references of bank processing systems or client references.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++	messageIdentification	[0..1]	Max35Text	Payment identification
++++	accountServicerReference	[0..1]	Max35Text	Bank payment reference
++++	instructionIdentification			
++++	endToEndIdentification			
++++	paymentInformationIdentification	[0..1]	Max35Text	Payment identification
++++	mandateIdentification	[0..1]|[1..1]	Max35Text	Mandate

				reference
++++	chequeNumber	[0..1]	Max35Text	Cheque no.
++++	clearingSystemReference	[0..1]	Max35Text	Clearing reference

JSON example of element:

```
"references": {
   "endToEndIdentification": "VS0250117002/SS0000000000/KS0000"
}
```

4.13.1.1.1 messageIdentification (Assumed payment identification)

JSON record: entryDetails.transactionDetails.references.messageIdentification

Occurrence: [0..1]

Definition: Payment identification

Application: the assumed identification of a payment entered by a client on its initiation or order of the payment in the statement of payment history.

Format type: Max35Text

4.13.1.1.2 accountServicerReference (Bank payment reference)

JSON record: entryDetails.transactionDetails.references.accountServicerReference

Occurrence: [0..1]

Definition: Payment identification

Application: A bank reference assigned to the payment, e.g. on the initiation through direct banking.

Format type: Max35Text

4.13.1.1.3 paymentInformationIdentification (Payment identification)

JSON record: entryDetails.transactionDetails.references.paymentInformationIdentification

Occurrence: [0..1]

Definition: Payment identification

Application: A bank reference assigned to the payment by the bank. For card payments, the sequence no. of payment card may be added. For a domestic payment, the specific symbol may be entered.

Format type: Max35Text

4.13.1.1.4 mandateIdentification (Mandate reference)

JSON record: entryDetails.transactionDetails.references.mandateIdentification

Occurrence: [0..1], for SDD (Sepa Direct Debit) the occurrence is mandatory [1..1]

Definition: Identification of SDD mandate

Application: For payment processed in the SEPA Direct Debit scheme, a mandate reference is mandatory in the field, through which the client gave consent to debit the account.

Format type: Max35Text

4.13.1.1.5 chequeNumber (Cheque number)

JSON record: entryDetails.transactionDetails.references.chequeNumber

Occurrence: [0..1]

Definition: Payment identification

Application: For cheque transactions, the cheque no. may be stated here. For card transactions, a hidden card no. may be stated with asterisks, i.e. the card no. is given in the format xxxxxxxxxxxx1234.

Format type: Max35Text

4.13.1.1.6 clearingSystemReference (Clearing reference)

JSON record: entryDetails.transactionDetails.references.clearingSystemReference

Occurrence: [0..1]

Definition: Payment identification

Application: A code list value defined by the bank, identifying the payment type or the payment type name used. For card transactions, the card association may be stated.

Format type: Max35Text

4.13.1.2 amountDetails (Amount details)

JSON record: entryDetails.transactionDetails.amountDetails

Occurrence: [0..1]

Definition: Payment amount specification

Application: Details for payment amount, mainly for conversion payment or cashback.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++	instructedAmount			
++++	transactionAmount	[0..0]	-	Transaction amount
++++	counterValueAmount	[0..1]		Converted amount
++++	proprietaryAmount	[0..0]		Cashback amount

JSON example of element:

```
"amountDetails": {
    "instructedAmount": {
       "amount": {
          "value": 122.22,
          "currency": "CZK"
       }
    }
},
```

4.13.1.2.1 transactionAmount (Transaction amount)

JSON record: entryDetails.transactionDetails.amountDetails.instructedAmount.transactionAmount

Occurrence: [0..0]

Definition: Payment amount and currency

Application: A field not supported by the standard which, however, may be provided by any bank individually. It is used in cases where cumulated payments and Cashback are cleared.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	amount	[1..1]	-	Amount definition

4.13.1.2.2 counterValueAmount (Converted amount)

JSON record: entryDetails.transactionDetails.amountDetails.instructedAmount.counterValueAmount

Occurrence: [0..1]

Definition: Payment amount and currency in the client account currency

Application: Transaction value after conversion of amount which was demanded by the client for transfer.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	amount	[1..1]	-	Amount definition
+++++	currencyExchange	[0..1]	-	Exchange rate

JSON example of element:

```
"counterValueAmount": {
   "amount": {
      "currency": "EUR",
      "value": 86200
   },
   "currencyExchange": {
      "sourceCurrency": "EUR",
      "targetCurrency": "CZK",
      "exchangeRate": 27.01
   }
}
```

4.13.1.2.2.1 currencyExchange (Exchange rate)

JSON record:
entryDetails.transactionDetails.amountDetails.instructedAmount.counterValueAmount.currencyExchange

Occurrence: [0..1]

Definition: Information on currencies and Exchange rates used.

Application: Used in case that the payment currency and account currency are different, and an Exchange rate was used to make the transaction.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	sourceCurrency	[1..1]	-	Amount definition

+++++	targetCurrency	[0..1]	-	Exchange rate

JSON example of element:

```
"currencyExchange": {
   "sourceCurrency": "EUR",
   "targetCurrency": "CZK",
   "exchangeRate": 27.01
}
```

4.13.1.2.2.1.1 sourceCurrency (Original currency)

JSON record:
entryDetails.transactionDetails.amountDetails.instructedAmount.counterValueAmount.currencyExchange.sourceCurrency

Occurrence: [1..1]

Definition: Client account currency

Application: Source/original currency/payer account currency for intrabank conversion payments.

Format type: CurrencyCode according to ISO 4217

JSON example of element:

```
"sourceCurrency": "EUR"
```

4.13.1.2.2.1.2 targetCurrency (Target currency)

JSON record:
entryDetails.transactionDetails.amountDetails.instructedAmount.counterValueAmount.currencyExchange.targetCurrency

Occurrence: [0..1]

Definition: Payment currency

Application: Final/target currency/payee account currency for intrabank conversion payments.

Format type: CurrencyCode according to ISO 4217

JSON example of element:

```

```

```
    "targetCurrency": "CZK",
```

4.13.1.2.3 proprietaryAmount (Cashback amount)

JSON record: entryDetails.transactionDetails.amountDetails.instructedAmount.proprietaryAmount

Occurrence: [0..0]

Definition: Cash withdrawal amount through the Cashback.

Application: Only in case that the bank supports the field.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	type	[0..0]	-	Transaction type
+++++	amount	[1..1]	-	Transaction amount

4.13.1.2.3.1 type (Transaction type)

JSON record: entryDetails.transactionDetails.amountDetails.instructedAmount.proprietaryAmount.type

Occurrence: [0..0]

Definition: Defines the transaction type

Application: Only for Cashback

Format type: constant „CHASHBACK“

4.13.1.3 charges (Fees)

JSON record: entryDetails.transactionDetails.charges

Occurrence: [0..n]

Definition: Information on fees

Application: The field is defined only for foreign payments when from the original transaction amount, a fee may be deducted by the correspondent bank.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++	bearer	[0..1]	-	Fee clearing side

4.13.1.3.1 bearer (Side for fees)

JSON record: entryDetails.transactionDetails.charges.bearer

Occurrence: [0..1]

Definition: Information on the side to which fees will be charged

Application: The side which bears the associated costs is defined by the code phrase:

DEBT – All fees paid by the payer

CRED – All fees paid by the payee

SHAR – Shared fees between the payer and payee.

SLEV – Fees required by the legislation or the payment scheme used, are applied

Format type: Code

4.13.1.4 relatedParties (Payer information)

JSON record: entryDetails.transactionDetails.relatedParties

Occurrence: [0..1]

Definition: Information on the payer, payer account and original payer and payee, payee account and final payee participating in the payment.

Application: The field is mandatory in the case of domestic, foreign and SEPA transfers

***Type**: This message element consists of the following elements:*

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+++++	debtor	[1..1]	-	Payer
+++++	debtorAccount			
+++++	ultimateDebtor			
+++++	creditor			
+++++	creditorAccount			
+++++	ultimateCreditor			
+++++	proprietary			

JSON example of element:

```
"relatedParties": {
   "debtor": {
      "name": "RENWORTH s.r.o",
      "identification": {
         "organisationIdentification": {
```

```
                "other": {
                   "identification": "48135283",
                   "schemeName": {
                      "code": "1.2.203.48135283",
                      "proprietary": "RENWORTH s.r.o"
                   }
                }
             }
          }
       }
    }
```


4.13.1.4.1.1 type (Card transaction type)

JSON record: entryDetails.transactionDetails.relatedParties.proprietary.type

Occurrence: [0..0]

Definition: For card transactions, it is used for marking of whether it is own ATM or ATM of another.

Application: Only for card transactions. The field is not supported by the standard, but may be used by individual banks.

Format type: Max35Text

4.13.1.4.1.2 party (Operator)

JSON record: entryDetails.transactionDetails.relatedParties.proprietary.party

Occurrence: [0..0]

Definition: The field informs on the ATM owner/operator.

Application: Only for card transactions. The field is not supported by the standard, but may be used by individual banks.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++++	name	[1..1]	-	Payer

4.13.1.5 relatedAgents (Payer and payee bank)

JSON record: entryDetails.transactionDetails.relatedAgents

Occurrence: [1..1]

Definition: Information on payer bank and payee bank.

Application: The element is present in payment transactions. In other cases (fees, interest, etc…) it makes no sense.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	debtorAgent	[1..1]	-	Payer bank

JSON example of element:

```
"relatedAgents": {
   "debtorAgent": {
      "financialInstitutionIdentification": {
         "bic": "GIBACZPXXXX"
      }
   }
}
```

4.13.1.5.1 debtorAgent (Payer bank)

JSON record: entryDetails.transactionDetails.relatedAgents.debtorAgent

Occurrence: [1..1]

Definition: Information on payer bank.

Application: It is according to the payment direction, to be filled in for the counterparty. The element is present in payment transactions. In other cases (fees, interest, etc…) it makes no sense.

***Type**: This message element consists of the following elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	financialInstitutionIdentification	[1..1]	-	Financial institution identification

4.13.1.6 additionalTransactionInformation (Additional information)

JSON record: entryDetails.transactionDetails.additionalTransactionInformation

Occurrence: [0..1]

Definition: Additional information stated by the bank

Application: For information which is not defined in the standard structure. For instance, additional info for SEPA Direct Debit i.e. Creditor Identifier, Payment scheme, the order of SEPA collection, etc.

Format type: Max500Text

JSON example of element:

```
    "additionalTransactionInformation": "8201701069595 BIC: GIBACZPXXXX;
#71A# SHA ZALOHA DLE SMLOUVY O DODAVKACH,zaloha dle smlouvy o dodavkach c.
45678/2017,VS0250117002/SS0000000000/KS0000SEPA převod"
```

4.14 entryReference (Payment number)

JSON record: entryReference

Occurrence: [0..1]

Definition: Payment identification number

Application: A clear transaction identifier assigned by the bank.

Format type: Max35Text

JSON example of element:

```
    "entryReference": "RB-4567813",
```

4.15 exchangeIdentification (Identifier assigned by third party)

JSON record: entryReference

***Occurrence**: [1…1]*

Definition: Clear identification of request

Application: Identification is assigned by the third party; the value will be contained in the response to request.

Format type: Max18Text

JSON example of element:

```
```

```
  "exchangeIdentification": 123456,
```

4.16 xchangeRateInformation (Contractual Exchange rate)

JSON record: exchangeRateInformation

Occurrence ALL: [0..0]

Definition: The element provides details about the exchange rate and Contract. The use and support of embedded element depends on the agreement with a specific bank.

Type: This message element consists of the following elements **ExchangeRateInformation1**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	exchangeRate	[0..0]	BaseOneRate	Agreed Exchange rate
++	rateTyp	[0..0]	ExchangeRate Type1Code	Type of Agreed Exchange rate
++	contractIdentification	[0..0]	Max35Text	Indicator of the use of Agreed Exchange rate

4.16.1 exchangeRate (Agreed Exchange rate)

JSON record: exchangeRateInformation.exchangeRate

Occurrence ALL: [0..0]

Definition: Detail of Agreed Exchange rate

Type: BaseOneRate

4.16.2 rateType (Type of Agreed Exchange rate)

***JSON record**: exchangeRateInformation.rateType*

***Occurrence ALL**: [0..0]*

***Definition** : Information on the type of Exchange rate used – AGREED, SALE, SPOT (agreed, sale, spot).*

***Type**: ExchangeRateType1Code*

4.16.3 contractIdentification (Identifier of the use of Agreed Exchange rate)

JSON record: exchangeRateInformation .contractIdentification

Occurrence ALL: [0..0]

Definition : Detail of Contract agreed with client for the execution of payments at the Agreed Exchange rate.

Type: Max35Text

4.17 chargesAccount (Account for fees)

JSON record: chargesAccount

Occurrence ALL: [0..0]

Definition: The account used to process transaction fees. The fee account is used if the fees are not to be charged to the debit of account marked as the payer's account. The fee account must be kept in the same bank as the payer's account.

Type: This message element consists of the following elements:

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
++	identification	[0..0]		Identification of account no. for fees
++	currency	[0..0]	CurrencyCode – ISO 4217	Account currency for fees

4.17.1 identification (Identification of account number for fees)

JSON record: chargesAccount.identification

Occurrence ALL: [0..0]

Definition: Unique and unambiguous identification of fee account administered at the payer's bank.

Type: This message element consists of the following elements **AccountIdentification4CZ**:

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+++	IBAN	[0..0]	IBAN2007Identifier	Account no. for fees in the IBAN format

4.17.1.1 IBAN (Account number for fees in the IBAN format)

JSON record: amount.equivalentAmount.curren chargesAccount.identification.iban

Occurrence ALL: [0..0]

Definition: International account number format used by financial institutions to uniquely and unambiguously identify a client account in accordance with the standard ISO 13616.

Data type: IBAN2007Identifier

Format for the Czech Rep.: [A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,20}, valid IBAN consists of all the following components: country code = CZ, check digit and account no. in local BBAN format, whereas BBAN consists of a four-digit numerical code of the bank, a six-digit prefix and a ten-digit account no. Initial zeros have no meaning.

Example: **CZ6907101781240000000415**

4.17.2 currency (Fee account currency)

JSON record: chargesAccount.currency

Occurrence ALL: [0..0]

Definition: The currency in which the charge account is kept. If that currency does not match the charge account number, the payment order may be rejected.

Format type: [A-Z]{3,3} - CurrencyCode, ISO 4217

4.18 chargeBearer (Fee payer)

JSON record: chargeBearer

Occurrence:

TUZEM (Domestic payments): [0. .0]

SEPA (SEPA payments): [0..0]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Provides the party (parties) that will pay fees associated with the payment transaction processing.

TUZEM (Domestic payments): The code of the fee payer cannot be chosen (the payer always pays the fees of the payer's bank and the payee pays the fees of the payee's bank). When processing domestic payments within CNB clearing, the fee payer's code is not stated.

SEPA (SEPA payments): The code of the fee payer cannot be chosen (the payer always pays the fees of the payer's bank and the payee pays the fees of the payee's bank). In SEPA payment processing, the payer's bank will always complete the SLEV value.

EHP (Foreign payments within EEA): Permitted values are DEBT = OUR or SHAR = SHA. If no value is filled in, the payer's bank will fill in the SHAR value.

NONEHP (Foreign payments outside EEA): Permitted values are DEBT = OUR or CRED = BEN or SHAR = SHA. If no value is filled in, the payer's bank will fill in the SHAR value.

Data type: ChargeBearerType1Code

CODE	NAME	DEFINITION
DEBT	BorneByDebtor	All transaction charges are to be borne by the debtor.
CRED	BorneByCreditor	All transaction charges are to be borne by the creditor.
SHAR	Shared	In a credit transfer context, means that transaction charges on the sender side are to be borne by the debtor, transaction charges on the receiver side are to be borne by the creditor.
SLEV	FollowingServiceLevel	Charges are to be applied following the rules agreed in the service level and/or scheme.

4.19 intermediaryAgent1 (Intermediary bank 1)

JSON record: intermediaryAgent1

Occurrence All: [0..0]

Definition: Financial institution (Intermediary bank), through which money goes to the payee's bank.

Type: This message element consists of the following elements **BranchAndFinancialInstitutionIdentification4CZ**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	financialInstitutionIdentification	[0..0]		Financial institution identification

4.19.1 financialInstitutionIdentification (Financial institution identification)

JSON record: intermediaryAgent1.financialInstitutionIdentification

Occurrence All: [0..0]

Definition: Unique and unambiguous identification of the financial institution assigned by the international standard or using own identification scheme.

Type: This message element consists of the following elements **FinancialInstitutionIdentification7CZ**

4.19.1.1 BIC (BIC / SWIFT code of bank)

JSON record: intermediaryAgent1.financialInstitutionIdentification.bic

Occurrence All: [0..0]

Definition: Bank Identifier Code. A code assigned to financial institutions by the Registration authority according to the international identification scheme as described in the latest version of the ISO 9362. The bank code in the format of BIC / SWIFT code has exactly 8 or exactly 11 alphanumeric characters.

Data type: BICIdentifier

4.19.1.2 clearingSystemMemberIdentification (Identification of clearing system participant)

JSON record: intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification

Occurrence All: [0..0]

Definition: Identification of the participant in the local clearing system

Type: This message element consists of the following elements **ClearingSystemMemberIdentification2**

4.19.1.2.1 clearingSystemIdentification (Identification of clearing system)

JSON record: intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification

Occurrence All: [0..0]

Definition: Identification of the clearing system

Type: This message element consists of the following elements **ClearingSystemIdentification2Choice**:

4.19.1.2.1.1 code (Code)

JSON record: intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification.code

Occurrence All: [0..0]

Definition: A code identifying the local clearing system as specified in the external list of codes.

Data type: ExternalClearingSystemIdentification1Code

4.19.1.2.1.2 proprietary (Free format)

JSON record: intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification.proprietary

Occurrence All: [0..0]

Definition: Identification of local clearing system, in free format.

Data type: Max35Text

4.19.1.2.2 memberIdentification (Participant's Identification code – clearing code)

JSON record: intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification.memberIdentification

Occurrence All: [0..0]

Definition: Identification code of the Participant or the bank's clearing code.

Data type: Max35Text

4.19.1.3 name (Name)

JSON record: intermediaryAgent1.financialInstitutionIdentification.name

Occurrence All: [0..0]

Definition: Name under which the party is known, and which is normally used to identify the party.

Data type: Max70Text

4.19.1.4 postalAddress (Postal address)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress

Occurrence All: [0..0]

Definition: Information that localizes and identifies the specific address such as the postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**

If for foreign payments it is agreed that this element should be stated, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two address lines from the ***Address Line*** field. Typically, the first line shows the street and the Land registry number, the city and city's postal code in the second line.

4.19.1.4.1 streetName (Street)

***JSON record**: intermediaryAgent1.financialInstitutionIdentification.postalAddress.streetName*

***Occurrence All**: [0..0]*

***Definition:** Street name or section.*

***Data type**: Max70Text*

4.19.1.5 buildingNumber (Building number)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.buildingNumber

Occurrence All: [0..0]

Definition: Number that identifies the position of building in the street.

Data type: Max16Text

4.19.1.6 postCode (Postcode)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.postCode

Occurrence All: [0..0]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

4.19.1.7 townName (Town)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.townName

Occurrence All: [0..0]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

4.19.1.8 country (Country)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.country

Occurrence All: [0..0]

Definition: A country with its own government

Data type: CountryCode

4.19.1.9 addressLine (Unstructured address record)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.addressLine

Occurrence All: [0..0]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.19.1.10 other (Another bank identification)

JSON record: intermediaryAgent1.financialInstitutionIdentification.other

Occurrence All: [0..0]

Definition: Unique agent identification assigned to the institution using the identification scheme

Type: This message element consists of the following elements **GenericFinancialIdentification1CZ**

4.19.1.10.1 identification (Local bank code)

JSON record: intermediaryAgent1.financialInstitutionIdentification.other.identification

Occurrence All: [0..0]

Definition: Local format of the bank code

Data type : Max35Text

4.20 instructionForNextAgent (Instruction for another bank)

JSON record: instructionForNextAgent

Occurrence All: [0..0]

Definition: Other information related to the payment order processing which is to be executed by another bank.

Type: Instruction code

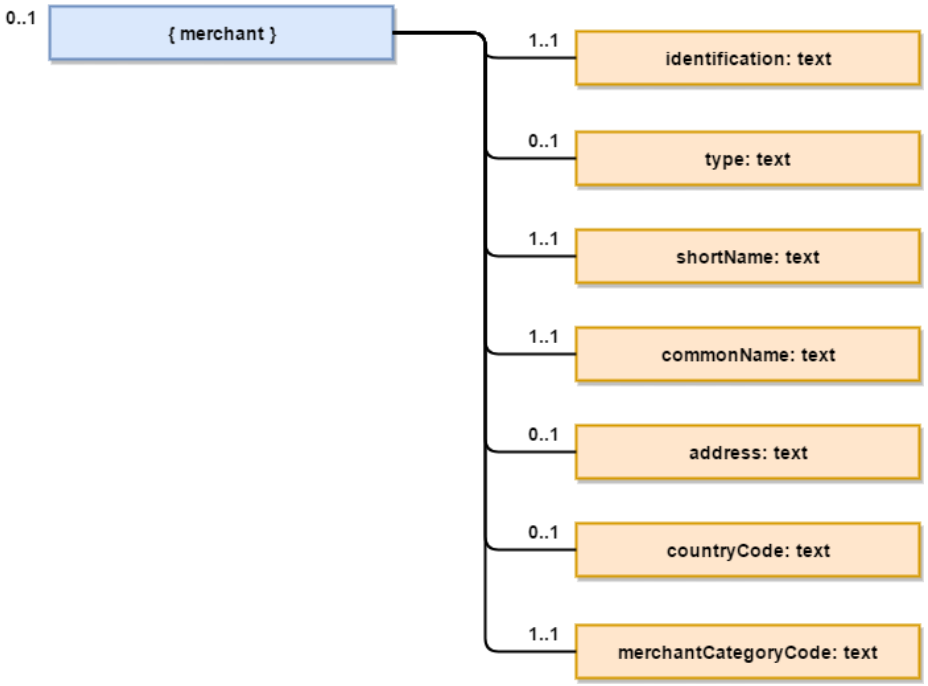
4.21 merchant (Merchant)

JSON record: merchant

Occurrence: [0…1]

Definition: A merchant executing a balance query

Application: In case that the third party and merchant are different entities.



*Element **merchant** contains the following embedded elements:*

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*	*LEVEL*
++	identification	[1..1]	CISP	Max35Text	Merchant identification
++	type	[0..1]	CISP	Code	Merchant type
++	shortName	[1..1]	CISP	Max35Text	Merchant name
++	commonName	[1..1]	CISP	Max70Text	Merchant name as stated in the payment receipt
++	address	[0..1]	CISP	Max140Text	Merchant address
++	countryCode	[0..1]	CISP	CountryCode, ISO 3166	Merchant country
++	merchantCategoryCode	[1..1]	CISP	Min3Max4Text, ISO 18245	Merchant code following the transaction type

JSON example of element:

```
"merchant": {
    "identification": "471 16 129",
    "shortName": "NEOLUXOR",
```

```
    "commonName": "Neoluxor s.r.o.",
    "address": "Hlavní 5, Praha 1",
    "countryCode": "CZ",
    "merchantCategoryCode": "5192"
},
```

4.21.1 identification (Identification)

JSON record: merchant.identification

Occurrence: [1...1]

Definition: Identification of an entity requesting a disposable balance

Application: Please specify the Reg. No. (IČO), in the case of an international payment, it is possible to use its variant of the country concerned.

Format type: Max35Text

JSON example of element:

```
    "identification": "471 16 129"
```

4.21.2 type (Entity type)

JSON record: merchant.type

Occurrence: [0...1]

Definition: Type of entity executing a query for disposable balance

Format type: Code

CODE	NAME	DEFINITION
OPOI	OriginatingPOI	Point Of Interaction initiating the card payment transaction.
MERC	Merchant	Merchant providing goods and service in the card payment transaction.
ACCP	Acceptor	Card acceptor, party accepting the card and presenting transaction data to the acquirer.
ITAG	IntermediaryAgent	Party acting on behalf of other parties to process or forward data to other parties.

ACQR	Acquirer	Entity acquiring card transactions.
CISS	CardIssuer	Party that issues cards.
DLIS	DelegateIssuer	Party to whom the card issuer delegates to authorise card payment transactions.

4.21.3 shortName (Entity name)

JSON record: merchant.shortName

Occurrence: [1...1]

Definition: Name of entity executing a query for disposable balance – shortened form.

Format type: Max35Text

JSON example of element:

```
"shortName": "NEOLUXOR"
```

4.21.4 commonName (Entity name)

JSON record: merchant.commonName

Occurrence: [1...1]

Definition: Full name of entity as it will be stated in the payment receipt.

Format type: Max70Text

JSON example of element:

```
"commonName": "Neoluxor s.r.o.",
```

4.21.5 address (Address)

JSON record: merchant.address

Occurrence: [0...1]

Definition: Address of entity executing a query for disposable balance.

Application: If the address of the establishment is not the same as the entity address, the address of the establishment or the place where the disposable balance request was initiated, will be given. The address is given without the country name.

Format type: Max140Text

JSON example of element:

```
"address": "Hlavní 5, Praha 1",
```

4.21.6 countryCode (Country code)

JSON record: merchant.address

Occurrence: [0...1]

Definition: The country of the entity making the query for disposable balance in ISO format of 3 alphanumeric characters supported by SWIFT

Format type: CountryCode, ISO 3166

JSON example of element:

```
"countryCode": "CZ"
```

4.21.7 merchantCategoryCode (Category code)

JSON record: merchant.merchantCategoryCode

Occurrence: [1...1]

Definition: ISO 18245 category code that applies to the type of service or goods according to the purpose for which the query for disposable balance is made.

Format type: Min3Max4Text, ISO 18245

JSON example of element:

```
"merchantCategoryCode": "5192"
```

4.22 paymentIdentification (Payment identification)

JSON record: paymentIdentification

***Occurrence ALL**: [1...1]*

***Definition**: Set of elements used to identify the payment instruction.*

***Type**: This message element consists of the following elements **PaymentIdentification1**:*

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	instruction Identification	Max35Text	Instruction Identification
++	endToEndIdentification	Max35Text	End To End Identification
++	transactionIdentification	Max35Text	Transaction identification

JSON example of element:

```
"paymentIdentification": {
    "instructionIdentification": "NOTPROVIDED",
    "transactionIdentification": "048885570000001020045"
}
```

4.22.1 instructionIdentification (Instruction identification)

JSON record: paymentIdentification.instructionIdentification

Occurrence ALL: [1..1]

Definition: Unique identification assigned by PISP that uniquely / unambiguously identifies the instruction. Instruction identification is a Point-to-Point reference that can be used between the instructor and the instructed party related to this one (individual) instruction. It may appear in several messages related to the instruction.

Format type: Max35Text

JSON example of element:

```
"instructionIdentification": "NOTPROVIDED",
```

4.22.2 endToEndIdentification (End To End identification)

JSON record: paymentIdentification.endToEndIdentification

Occurrence:

TUZEM (Domestic payments): [0. .0]

EHP (Foreign payments within EEA): [0..0]

NONEHP (Foreign payments outside EEA): [0..0]

SEPA (SEPA payments): [1..1]

Definition: Identification agreed between the payer and the payee. If not filled in by the payer, then PISP will fill in „NOTPROVIDED“.

Format type: Max35Text

JSON example of element:

```
"endToEndIdentification": "048885570000001020045",
```

4.22.3 transactionIdentification (Transaction identification)

JSON record: paymentIdentification.transactionIdentification

Occurrence request ALL: [0..0]

Occurrence response ALL: [1..1]

Definition: The unique reference of the payer's bank. This identification is returned by the provider who maintains the payer's account in response to the received request for a new payment and / or a payment status query.

Format type: Max35Text

JSON example of element:

```
"transactionIdentification": "048885570000001020045"
```

4.23 paymentTypeInformation (Information on payment type)

JSON record: paymentTypeInformation

Occurrence ALL: [0..1]

Definition: A set of elements used to further determine the transaction type. It is used to determine the speed and method of payment processing.

Type: This message element consists of the following elements **PaymentTypeInformation19**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	instructionPriority	[0..1]		Instruction priority
++	serviceLevel	[0..0]		Service level
++	categoryPurpose	[0..0]		Payment purpose category

JSON example of element:

```
  "paymentTypeInformation": {
      "instructionPriority": "NORM",
      "serviceLevel": {
        "code": "DOMESTIC"
      }
},
```

4.23.1 instructionPriority (Instruction priority)

JSON record: paymentTypeInformation.instructionPriority

Occurrence ALL: [0..1]

Definition: If the element is not filled in, the payer's provider assigns a NORM value and makes a payment by default. If a "HIGH" value is indicated, the payer's provider will make the payment as priority / express / urgent, usually at D + 0 foreign currency. If the "NORM" is specified, they will make the payment by default, i.e. D + 1 foreign currency, where D, the date of debiting funds from the payer's account. The deadlines for making standard and priority payments may vary depending on the type of payment and are governed by the terms of each payer's conditions.

Data type: Priority2Code

JSON example of element:

```
  "instructionPriority": "NORM"
```

4.23.2 serviceLevel (Service level)

JSON record: paymentTypeInformation.serviceLevel

Occurrence ALL: [0..0]

Definition: An agreement or rules according to which the transaction is to be processed.

Data type**: This message element consists of the following elements **ServiceLevel8CZ

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	Code	[0..0]	Code	Code

JSON example of element:

```
"serviceLevel": {
   "code": "DOMESTIC"
}
```

4.23.2.1 code (Code)

JSON record: paymentTypeInformation.serviceLevel.code

Occurrence ALL: [0..0]

Definition: An agreement or rules according to which the transaction is to be processed.

Data type: ExternalServiceLevel1Code

JSON example of element:

```
"serviceLevel": {
   "code": "DOMESTIC"
}
```

4.23.3 categoryPurpose (Payment purpose category)

JSON record: paymentTypeInformation.categoryPurpose

Occurrence ALL: [0..0]

Definition: Instruction code to the payee's bank or payer's bank for an agreed specific payment processing method.

Data type: This message element consists of the following elements **CategoryPurpose1Choice**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	Code	[0..0]	Code	Code
+++	Proprietary	[0..0]	Max35Text	Free format

4.23.3.1 code (Code)

JSON record: paymentTypeInformation.categoryPurpose.code

Occurrence ALL: [0..0]

Definition: An agreement or rules according to which the transaction is to be processed.

Data type: ExternalCategoryPurpose1Code

4.23.3.2 proprietary (Free format)

JSON record: paymentTypeInformation.categoryPurpose.proprietary

Occurrence ALL: [0..0]

Definition: Payment purpose category, in free format..

Data type: Max35Text

4.24 purpose (Payment purpose)

JSON record: purpose

Occurrence All: [0..0]

Definition: Normalized payment reason code – optional field. To fill in payment information, for example, what kind of goods / services are paid. More details in ISO codes, at this link: *www.ISO20022.org*.

Type: This message element consists of the following elements **Purpose2Choice**:

LEVEL	*OR*	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
++	{Or	code	[0..0]	Code	Code
++	Or}	proprietary	[0..0]	Max35Text	Free format

JSON example of element:

```
"purpose": {
   "proprietary": "PLATBA ZA SLUŽBY"
},
```

4.24.1 code (Code)

JSON record: purpose.code

Occurrence All: [0..0]

Definition: Payment purpose as stated in the external list of payment purpose codes.

Data type: ExternalPurpose1Code

4.24.2 proprietary (Free format)

JSON record: purpose.proprietary

Occurrence All: [0..0]

Definition: Payment purpose, in free format.

Data type: Max35Text

JSON example of element:

```
"proprietary": "PLATBA ZA SLUŽBY"
```

4.25 remittanceInformation (Payment information)

JSON record: remittanceInformation

Occurrence All: [0..1]

Definition: Information that allows match (i.e. pairing, reconciliation) of payments with which the payment should be settled, such as commercial invoices in the receivables system. Use the Message for payee, e.g., to forward a variable, specific, and constant symbol.

Type: This message element consists of the following elements **RemittanceInformation5CZ**:

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	unstructured	Max140Text	Unstructured message for payee
++	structured		Structured message for payee – variable, specific, and constant symbol

JSON example of element:

```
"remittanceInformation": {
```

```
        "unstructured": "``",
        "structured": {
           "creditorReferenceInformation": {
              "reference": "VS:123456\",\"KS:456789\",\"SS:879213546"
           }
        }
     },
```

4.25.1 unstructured (Unstructured message for payee)

JSON record: remittanceInformation.unstructured

Occurrence All: [0..1]

Definition: If you require a variable, specific, and / or constant symbol to be forwarded to the payee, specify the unstructured information at the beginning in the form of VS/max.10 digits/SS/max.10 digits/KS/max.10 digits.

Data type: Max140Text

JSON example of element:

```
    "unstructured": "``"
```

4.25.2 structured (Structured message for payee – variable, specific and constant symbol)

JSON record: remittanceInformation.structured

Occurrence only TUZEM (Domestic payments): [0..3]

Definition: Information allowing the matching of the variable, specific or constant symbol for the items with which the transfer should be settled. Structured Remittance Information can be used to fill in maximum one variable symbol, one specific symbol, or one constant symbol.

Type: This message element consists of the following elements **StructuredRemittanceInformation7CZ**:

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
+++	creditorReferenceInformation	-	Information on payee reference

JSON example of element:

```
    "structured": {
```

```
      "creditorReferenceInformation": {
         "reference": "VS:123456\",\"KS:456789\",\"SS:879213546"
      }
   }
```

4.25.2.1 creditorReferenceInformation (Information on payee reference)

JSON record: remittanceInformation.structured.creditorReferenceInformation

Occurrence only TUZEM (Domestic payments): [0..1]

Definition: Reference information provided by the payee that allows identification of underlying documents (e.g. invoice).

Type: This message element consists of the following elements **CreditorReferenceInformation2CZ**:

LEVEL	*MESSAGE ELEMENT*	*FORMAT TYPE*	*PRESENTATION*
++++	Reference	Max35Text VS – variable, SS – specific, KS – constant symbol. Each of the symbols as separate structured data.	Value of VS, SS, KS

JSON example of element:

```
   "creditorReferenceInformation": {
      "reference": "VS:123456\",\"KS:456789\",\"SS:879213546"
   }
```

4.25.2.1.1 reference (Value of VS, SS, KS)

JSON record: remittanceInformation.structured.creditorReferenceInformation.reference

Occurrence only TUZEM (Domestic payments): [0..1]

Definition: Information allowing to pass on information about the value of a variable symbol, a specific symbol, or a constant symbol. The value of variable symbol is recorded as VS:max.10 digits (e.g. VS:3451859072). The value of specific symbol is recorded as SS:max.10 digits (e.g. SS:8451201274). The value of constant symbol is recorded as KS:max.10 characters (e.g. KS:0308).

Data type: Max35Text

JSON example of element:

```
"reference": "VS:123456\",\"KS:456789\",\"SS:879213546"
```

4.26 requestedExecutionDate (Required payment execution date)

JSON record: interbankSettlementDate/requestedExecutionDate

Occurrence ALL: [0..1]

Definition: The date on which the initiating party requests payment processing by a settlement agent. On this date, funds will be debited from the payer's account if the agreed terms are met. The bank terms pertaining to the execution date may vary, for example, depending on the business hours, the work day specification for the type of payment, the number of days to track the available balance, the number of days for future maturity, the date older than the current day, etc. If the requested execution date is not filled in, the payer's bank will debit funds from the payer's account immediately after meeting the agreed terms.

Data type: ISODate

JSON example of element:

```
"requestedExecutionDate": "2017-01-31"
```

4.27 reversalIndicator (Cancellation)

JSON record: reversalIndicator

Occurrence: [0..1]

Definition: Indication of whether it is cancellation.

Application: Only acquires the values ""true"", or ""False"":

true: It is cancellation

false: It is not cancellation

Format type: TrueFalseIndicator

JSON example of element:

```
"reversalIndicator": false
```

4.28 status (Status)

JSON record: status

Occurrence: [1..1]

Definition: Item status in the account from the point of view of the bank

Application: In the statement, only posted items will be shown, with the constant BOOK, or blocked items, with the constant PDNG.

Format type: Code

JSON example of element:

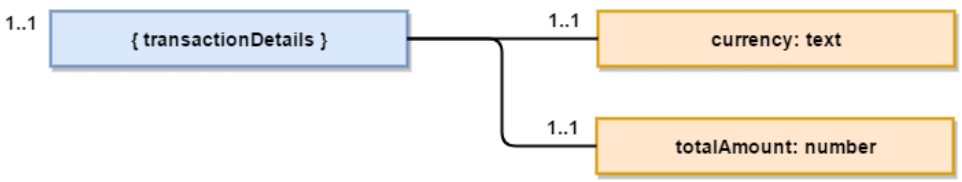
```
"status": "BOOK",
```

4.29 transactionDetails (Transaction details)

JSON record: transactionDetails

***Occurrence**: [1…1]*

***Definition**: Transaction details*



*Element **transactionDetails** contains the following embedded elements:*

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
++	currency	[1..1]	CISP	CurrencyCode, ISO 4217	Currency of query for balance
++	totalAmount	[1..1]	CISP	Amount	Amount of query for balance

JSON example of element:

```
"transactionDetails": {
  "currency": "CZK",
  "totalAmount": 10050.15
}
```

4.29.1 currency (Currency)

JSON record: transactionDetails.currency

Occurrence: [1...1]

Definition: The currency of the amount that is verified by querying the available balance.

Format type: CurrencyCode, ISO 4217

JSON example of element:

```
"currency": "CZK"
```

4.29.2 totalAmount (Total amount)

JSON record: *transactionDetails.totalAmount*

Occurrence: *[1...1]*

Definition: *Total amount that is verified by querying the available balance.*

Format type: *Max18.5Amount*

JSON example of element:

```
"currency": "CZK"
```

4.30 ultimateCreditor (Final payee)

JSON record: ultimateCreditor

Occurrence only SEPA (SEPA payments): [0..1]

Definition: Final party to whom the financial amount is due.

Type: This message element consists of the following elements **PartyIdentification32CZ1**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	Name	[0..1]	Max70Text	Name of final payee
++	Postal Address	[0..1]		Postal address of final payee
++	Identification	[0..1]		Identification of final payee

JSON example of element:

```
"ultimateCreditor": {
   "name": "Franz Schubert",
   "postalAddress": {
      "streetName": "Dianagasse",
      "buildingNumber": "6",
      "postCode": "1030",
      "townName": "Wiena",
      "country": "AT"
    },
    "identification": {
       "privateIdentification": {
          "other": {
           "identification": "12356879131",
           "schemeName": {
              "proprietary": "passport",
              "issuer": "WIENA"
           }
        }
    }
  }
}
```

4.30.1 name (Name)

JSON record: ultimateCreditor.name

Occurrence only SEPA (SEPA payments): [0..1]

Definition: The name under which the party is known, and which is commonly used to identify that party.

Data type: Max70Text

JSON example of element:

```
"name": "Franz Schubert"
```

4.30.2 postalAddress (Postal address)

JSON record: ultimateCreditor.postalAddress

Occurrence only SEPA (SEPA payments): [0..1]

Definition: Information that locates and identifies a specific address as a postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**

For SEPA payments, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two lines of address from the Address Line field. Typically, the first line shows the street and the Land registry number, and the second line the town and the postal code.

JSON example of element:

```
"postalAddress": {
   "streetName": "Dianagasse",
   "buildingNumber": "6",
   "postCode": "1030",
   "townName": "Wiena",
   "country": "AT"
}
```

4.30.2.1 streetName (Street)

JSON record: ultimateCreditor.postalAddress.streetName

Occurrence only SEPA payments: [0..1]

Definition: Name of street or section.

Data type: Max70Text

JSON example of element:

```
"streetName": "Dianagasse",
```

4.30.2.2 buildingNumber (Building number)

JSON record: ultimateCreditor.postalAddress.buildingNumber

Occurrence only SEPA payments: [0..1]

Definition: Number identifying the position of the building in the street.

Data type: Max16Text

JSON example of element:

```
"buildingNumber": "6",
```

4.30.2.3 postCode (Postcode)

JSON record: ultimateCreditor.postalAddress.postCode

Occurrence only SEPA payments: [0..1]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

JSON example of element:

```
"postCode": "1030",
```

4.30.2.4 townName (Town)

JSON record: ultimateCreditor.postalAddress.townName

Occurrence only SEPA payments: [0..1]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "Wiena",
```

4.30.2.5 country (Country)

JSON record: ultimateCreditor.postalAddress.country

Occurrence only SEPA payments: [0..1]

Definition: A country with its own government

Data type: CountryCode

Data format: [A-Z]{2,2}

Rule: Country code is checked against the list of country names under the ISO 3166.

JSON example of element:

```
"country": "AT"
```

4.30.2.6 addressLine (Unstructured address record)

JSON record: ultimateCreditor.postalAddress.addressLine

Occurrence only SEPA payments: [0..2]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.30.3 identification (Identification)

JSON record: ultimateCreditor.identification

Occurrence only SEPA payments: [0..1]

Definition: A unique and unambiguous identification of the party.

Type: This message element consists of the following elements **Party6Choice**:

LEVEL	OR	MESSAGE ELEMENT	OCCURRENCE	PRESENTATION
+++	{Or	Organisation Identification	[1..1]	Identification of organization
+++	Or}	Private Identification	[1..1]	Identification of private individual

JSON example of element:

```
"identification": {
   "privateIdentification": {
      "other": {
         "identification": "12356879131",
         "schemeName": {
            "proprietary": "passport",
            "issuer": "WIENA"
         }
      }
   }
}
```

4.31 ultimateDebtor (Original payee)

JSON record: ultimateDebtor

Occurrence only SEPA payments: [0..1]

Definition: The final debtor paying through the payer.

Type: This message element consists of the following elements (for SEPA payments):

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	Name	Max70Text	Name of original payee
++	Postal Address		Postal address of original payee
++	Identification		Identification

JSON example of element:

```
  "ultimateDebtor": {
      "name": "Karel Novák",
      "postalAddress": {
        "streetName": "Vodičkova",
        "buildingNumber": "12",
        "postCode": "12000",
        "townName": "Praha 2",
        "country": "CZ"
   },
      "identification": {
        "privateIdentification": {
          "other": {
            "identification": "1245789528",
            "schemeName": {
              "proprietary": "ID CARD",
              "issuer": "OU Praha 2,CZ"
          }
        }
      }
    }
  },
```

4.31.1 name (Name of original payee)

JSON record: ultimateDebtor.name

Occurrence only SEPA payments: [0..1]

Definition: The name under which the party is known, and which is commonly used to identify that party.

Data type: Max70Text

JSON example of element:

```
  "name": "Karel Novák"
```

4.31.2 postalAddress (Postal address of original payer)

JSON record: ultimateDebtor.postalAddress

Occurrence only SEPA payments: [0..1]

Definition: Information that locates and identifies a specific address as a postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**.

For SEPA payments, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two lines of address from the Address Line field. Typically, the first line shows the street and the Land registry number, and the second line the town and the postal code.

JSON example of element:

```
"postalAddress": {
   "streetName": "Vodičkova",
   "buildingNumber": "12",
   "postCode": "12000",
   "townName": "Praha 2",
   "country": "CZ"
}
```

4.31.2.1 streetName (Street)

JSON record: ultimateDebtor.postalAddress.streetName

Occurrence only SEPA payments: [0..1]

Definition: Name of street or section.

Data type: Max70Text

JSON example of element:

```
"streetName": "Vodičkova
```

4.31.2.2 buildingNumber (Building number)

JSON record: ultimateDebtor.postalAddress.buildingNumber

Occurrence only SEPA payments: [0..1]

Definition: Number identifying the position of the building in the street.

Data type: Max16Text

JSON example of element:

```
"buildingNumber": "12"
```

4.31.2.3 postCode (Postcode)

***JSON record**: ultimateDebtor.postalAddress.postCode*

Occurrence only SEPA payments*: [0..1]*

***Definition**: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.*

***Data type**: Max16Text*

JSON example of element:

```
"postCode": "12000"
```

4.31.2.4 townName (Town)

JSON record: ultimateDebtor.postalAddress.townName

Occurrence only SEPA payments: [0..1]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "Praha 2"
```

4.31.2.5 country (Country)

JSON record: ultimateDebtor.postalAddress.country

Occurrence only SEPA payments: [0..1]

Definition: A country with its own government

Data type: CountryCode

Data format: [A-Z]{2,2}

Rule: Country code is checked against the list of country names under the ISO 3166.

JSON example of element:

```
"country": "CZ"
```

4.31.2.6 addressLine (Unstructured address record)

JSON record: ultimateDebtor.postalAddress.addressLine

Occurrence only SEPA payments: [0..2]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.31.3 identification (Identification)

JSON record: ultimateDebtor. identification

Occurrence only SEPA payments: [0..1]

Definition: A unique and unambiguous identification of the party.

Type: This message element consists of the following elements **Party6Choice**:

LEVEL	*OR*	*MESSAGE ELEMENT*	*OCCURRENCE*	*PRESENTATION*
+++	{Or	organisationIdentification	[1..1]	Identification of organization
+++	Or}	privateIdentification	[1..1]	Identification of private individual

JSON example of element:

```
    "identification": {
          "privateIdentification": {
            "other": {
              "identification": "1245789528",
              "schemeName": {
                "proprietary": "ID CARD",
                "issuer": "OU Praha 2,CZ"
            }
          }
        }
      }
```

4.32 valueDate (Due date)

JSON record: valueDate.date, bookingDate.date

Occurrence: [0..1]

Definition: Due date/payment currency

Application: Depending on the Due date / payment currency and the way the bank presents data (and time)

Type: This message element consists of the following elements **Party6Choice**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PRESENTATION
+++	date	[1..1]	Date of processing

JSON example of element:

```
    "valueDate": {
       "date": "2016-09-05T00:00:00+01:00"
    }
```

4.32.1 date (Date)

JSON record: valueDate.date, bookingDate.date

Definition: Date of processing/posting the payment by the bank.

Application: Format ISODate i.e. YYYY-MM-DD, or ISODateTime i.e. YYYY-MM-DDThh:mm:ss.sTZD is depending on the transaction type and the way the bank presents data (and time) of payment processing/posting 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)"

Format type: ISODate or ISODateTime

JSON example of element:

```
"date": "2016-09-05T00:00:00+01:00"
```

5 Examples of Request and Response for Individual Messages

5.1 Example of request and response for Query for Balance Check at account.

Query URI used: /accounts/balanceCheck

5.1.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
```

Request body:

```
{
  "exchangeIdentification": 123456,
  "card": {
    "cardHolderName": "Jan Novák",
    "maskedPAN": "1234**********6789"
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ0708000000001019382023"
    },
    "currency": "CZK"
  },
  "authenticationMethod": "NPIN",
  "merchant": {
    "identification": "471 16 129",
    "shortName": "NEOLUXOR",
```

```
    "commonName": "Neoluxor s.r.o.",
    "address": "Hlavní 5, Praha 1",
    "countryCode": "CZ",
    "merchantCategoryCode": "5192"
  },
  "transactionDetails": {
    "currency": "CZK",
    "totalAmount": 10050.15
  }
}
```

5.1.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body

```
{
   "responseIdentification": 98765,
   "exchangeIdentification": 123456,
   "response": "APPR"
}
```

5.1.3 Example of response to wrong call

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
```

Response **400 Bad Request** Body


```
{
  "errors": [
    {
      "error":"FIELD_MISSING",
      "scope":"merchant.identification",
      "parameters": null,
      "message": null
    },
    {
      "error":"AC09",
      "scope": null,
      "parameters": null,
     "message": null
    }
  ]
}
```


5.2 Example of request and response for Query for list of accounts of bank's client

Query URI used: /my/accounts

5.2.1 Example of request

Request headers:

```
```

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

5.2.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body


```
{
   "pageNumber": 0,
   "pageCount": 2,
   "pageSize": 100,
   "nextPage": 1,
   "accounts": [
      {
         "id": "D2C8C1DCC51A3738538A40A4863CA288E0225E52",
         "identification": {
               "iban": "CZ0708000000001019382023",
               "other": "1019382023"
      },
         "currency": "CZK",
         "servicer": {
               "bankCode": "0800",
               "countryCode": "CZ",
               "bic": "GIBACZPX"
      },
         "nameI18N": "Muj hlavni osobni ucet",
```

```
            "productI18N": "Osobní účet ČS"
        }
    ]
}
```

5.2.3 Example of response to wrong call

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
```

Response **400 Bad Request** Body

```
{
  "errors": [
    {
      "error": "PAGE_NOT_FOUND"
    },
    {
      "error": "PARAMETER_INVALID",
      "scope": "sort"
    }
  ]
}
```

5.3 Example of request and response for Query for balance in a specific account of bank's client

Query URI used: /my/accounts/D2C8C1DCC51A3738538A40A4863CA288E0225E52/balance

5.3.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

5.3.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body

```
{
   "balances": [
   {
         "type": {
            "codeOrProprietary": {
              "code": "PRCD"
         }
      },
          "creditLine": {
              "included": true,
```

```
            "amount": {
                "value": 10000,
                "currency": "CZK"
        }
      },
          "amount": {
              "value": 4520.15,
              "currency": "CZK"
          },
          "creditDebitIndicator": "DBIT",
          "date": {
              "dateTime": "2017-02-17T12:32:41.0Z"
      }
    }
  ]
}
```


5.3.3 Example of response to wrong call

5.3.3.1 Error 404

Response **404 Not Found** Headers


```
HTTP/1.1 404 Not Found
Content-Type: application/json
```


Response **404 Not Found** Body


```
{
  "errors": [
    {
```

```
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.3.3.2 Error 400

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
```

Response **400 Bad Request** Body

```
{
  "errors": [
    {
      "error": "AM03",
      "scope": "currency"
    }
  ]
}
```

5.4 Example of request and response for Query for a transaction overview in a specific account of bank's client

Query URI used: /my/accounts/D2C8C1DCC51A3738538A40A4863CA288E0225E52/transactions

5.4.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

5.4.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body

```
{
      "pageNumber": 0,
      "pageCount": 2,
      "pageSize": 100,
      "nextPage": 1,
      "transactions": [
    {
          "entryReference": "RB-4567813",
          "amount": {
            "value": 10000,
            "currency": "CZK"
       },
          "status": "BOOK",
          "creditDebitIndicator": "DBIT",
          "bookingDate": {
            "date": "2017-01-31T00:00:00.000+01"
       },
```

```
        "valueDate": {
          "date": "2017-01-31T00:00:00.000+01"
      },
        "bankTransactionCode": {
          "proprietary": {
            "code": 1000010,
            "issuer": "CBA"
        }
      },
        "entryDetails": {
          "transactionDetails": {
            "amountDetails": {
              "instructedAmount": {
                "amount": {
                  "value": 10000,
                  "currency": "CZK"
              }
            }
          },
            "relatedParties": {
              "debtor": {
                "name": "Novák Jan"
            },
              "debtorAccount": {
                "identification": {
                  "iban": "CZ0827000000002108589434",
                  "other": {
                    "identification": "000000-2108589434"
                }
              }
            }
          },
            "relatedAgents": {
              "debtorAgent": {
                "financialInstitutionIdentification": {
                  "bic": "BACXCZPP",
                  "clearingSystemMemberIdentification": {
```

```
                    "memberIdentification": "2700"
                  }
                }
              }
            },
               "remittanceInformation": {
                 "unstructured": "``",
                 "structured": {
                   "creditorReferenceInformation": {
                     "reference":
"VS:123456\",\"KS:456789\",\"SS:879213546"
                 }
               }
            },
               "additionalTransactionInformation": "Domácí platba -
S24/IB,záloha plyn Bohemia Energy"
          }
        }
      },
      {
           "amount": {
             "value": 105.25,
             "currency": "CZK"
        },
           "status": "BOOK",
           "creditDebitIndicator": "DBIT",
           "bookingDate": {
             "date": "2016-09-05T00:00:00+01:00"
        },
           "valueDate": {
             "date": "2016-09-05T00:00:00+01:00"
        },
           "bankTransactionCode": {
             "proprietary": {
               "code": 4000050,
               "issuer": "CBA"
          }
        },
```

```
          "entryDetails": {
            "transactionDetails": {
              "references": {
                "chequeNumber": "xxxxxxxxxxxx1248"
           },
              "amountDetails": {
                "instructedAmount": {
                  "amount": {
                    "value": 10,
                    "currency": "GBP"
                }
              },
                "counterValueAmount": {
                  "amount": {
                    "currency": "CZK",
                    "value": 105.25
                },
                  "currencyExchange": {
                    "sourceCurrency": "GBP",
                    "targetCurrency": "CZK",
                    "exchangeRate": 10.525
                }
              }
           },
              "additionalTransactionInformation": "PLATBA KARTOU"
        }
      }
    },
    {
          "entryReference": "FC-4567513951",
          "amount": {
            "value": 1844777,
            "currency": "CZK"
      },
          "status": "BOOK",
          "creditDebitIndicator": "CRDT",
          "bookingDate": {
```

```
            "date": "2017-01-31T00:00:00.000+01"
      },
         "valueDate": {
            "date": "2017-01-31T00:00:00.000+01"
      },
         "bankTransactionCode": {
            "proprietary": {
               "code": 1000020,
               "issuer": "CBA"
        }
      },
         "entryDetails": {}
   },
   {
         "entryReference": "CDR-13457893331",
         "amount": {
            "value": 2,
            "currency": "CZK"
      },
         "status": "BOOK",
         "creditDebitIndicator": "DBIT",
         "bookingDate": {
            "date": "2016-09-05T00:00:00+01:00"
      },
         "valueDate": {
            "date": "2016-09-05T00:00:00+01:00"
      },
         "bankTransactionCode": {
            "proprietary": {
               "code": 4000010,
               "issuer": "CBA"
        }
      },
         "entryDetails": {
            "transactionDetails": {
               "amountDetails": {
                  "instructedAmount": {
```

```
                  "amount": {
                    "value": 2,
                    "currency": "CZK"
                }
              }
            },
               "additionalTransactionInformation": "POPLATEK ZA ODCHOZÍ
TRANSAKCÍ"
          }
        }
      },
      {
          "amount": {
            "value": 122.22,
            "currency": "CZK"
        },
          "status": "BOOK",
          "creditDebitIndicator": "CRDT",
          "bookingDate": {
            "date": "2016-09-05T00:00:00+01:00"
        },
          "valueDate": {
            "date": "2016-09-05T00:00:00+01:00"
        },
          "bankTransactionCode": {
            "proprietary": {
              "code": 9000020,
              "issuer": "CBA"
          }
        },
          "entryDetails": {
            "transactionDetails": {
              "amountDetails": {
                "instructedAmount": {
                  "amount": {
                    "value": 122.22,
                    "currency": "CZK"
```

```
                }
              }
            },
               "additionalTransactionInformation": "PŘIPSÁNÍ ÚROKU ZE
ZUSTATKU"
          }
        }
      },
      {
          "entryReference": "FP-4156489123",
          "amount": {
            "value": 2328262,
            "currency": "CZK"
        },
          "status": "BOOK",
          "creditDebitIndicator": "CRDT",
          "bookingDate": {
            "date": "2017-01-31T00:00:00.000+01"
        },
          "valueDate": {
            "date": "2017-01-31T00:00:00.000+01"
        },
          "bankTransactionCode": {
            "proprietary": {
              "code": 1000040,
              "issuer": "CBA"
          }
        },
          "entryDetails": {
            "transactionDetails": {
              "references": {
                "endToEndIdentification":
"VS0250117002/SS0000000000/KS0000"
            },
              "amountDetails": {
                "instructedAmount": {
                  "amount": {
                    "value": 2328262,
```

```
                "currency": "CZK"
            }
        },
            "counterValueAmount": {
            "amount": {
                "currency": "EUR",
                "value": 86200
            },
            "currencyExchange": {
                "sourceCurrency": "EUR",
                "targetCurrency": "CZK",
                "exchangeRate": 27.01
            }
        }
    },
        "relatedParties": {
        "debtor": {
            "name": "RENWORTH s.r.o",
            "identification": {
            "organisationIdentification": {
                "other": {
                "identification": "48135283",
                "schemeName": {
                    "code": "1.2.203.48135283",
                    "proprietary": "RENWORTH s.r.o"
                }
                }
            }
            }
        },
            "debtorAccount": {
            "identification": {
                "iban": "CZ1308001800640033122856"
            }
        }
    },
        "relatedAgents": {
```

```
                "debtorAgent": {
                  "financialInstitutionIdentification": {
                    "bic": "GIBACZPXXXX"
                }
              }
            },
                "purpose": {
                  "proprietary": "PLATBA ZA SLUŽBY"
            },
                "remittanceInformation": {
                  "structured": {
                    "creditorReferenceInformation": {
                      "reference": "VS:0250117002"
                }
              }
            },
                "additionalTransactionInformation": "8201701069595 BIC:
GIBACZPXXXX; #71A# SHA ZALOHA DLE SMLOUVY O DODAVKACH,zaloha dle
smlouvy o dodavkach c. 45678/2017,VS0250117002/SS0000000000/KS0000SEPA
převod"
          }
        }
      },
      {
            "amount": {
              "value": 105,
              "currency": "CZK"
        },
            "status": "BOOK",
            "creditDebitIndicator": "CRDT",
            "bookingDate": {
              "date": "2016-09-05T00:00:00+01:00"
        },
            "valueDate": {
              "date": "2016-09-05T00:00:00+01:00"
        },
            "bankTransactionCode": {
              "proprietary": {
```

```
                "code": 2000010,
                "issuer": "CBA"
            }
        }
    }
  ]
}
```

5.4.3 Example of response to wrong call

5.4.3.1 Error 404

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.4.3.2 Error 400

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
```

Response **400 Bad Request** Body

```
{
  "errors": [
    {
      "error": "AM03",
      "scope": "currency"
    },
    {
      "error": "DT01",
      "parameters": {
        "DATE": "DATE_TO_OLD"
      },
      "scope": "fromDate"
    },
    {
      "error": "DT01",
      "parameters": {
        "DATE": "DATE_IN_FUTURE"
      },
      "scope": "toDate"
    }
  ]
}
```

5.5 Example of request and response after entering a new payment

Query URI used: POST /my/payments

5.5.1 Example of request Domestic payment

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
      "paymentIdentification": {
        "instructionIdentification": "NejakeID41785962314574"
  },
      "paymentTypeInformation": {
        "instructionPriority": "NORM"
  },
      "amount": {
        "instructedAmount": {
          "value": 1245.44,
          "currency": "CZK"
     }
  },
      "requestedExecutionDate": "2017-01-31",
      "debtorAccount": {
        "identification": {
          "iban": "CZ7508000000002108589434"
     },
        "currency": "CZK"
  },
```

```
      "creditorAccount": {
        "identification": {
          "iban": "CZ6330300000000000000123"
     },
        "currency": "CZK"
   },
      "remittanceInformation": {
        "unstructured": "VS/7418529630/SS/1234567890"
   }
}
```

5.5.2 Example of request SEPA payment

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
      "paymentIdentification": {
        "instructionIdentification": "SEPAFUL123",
        "endToEndIdentification": "``"
   },
      "paymentTypeInformation": {
        "instructionPriority": "HIGH"
   },
      "amount": {
        "instructedAmount": {
```

```
        "value": 45789.45,
        "currency": "EUR"
    }
  },
    "requestedExecutionDate": "2017-01-31",
    "ultimateDebtor": {
      "name": "Karel Novák",
      "postalAddress": {
        "streetName": "Vodičkova",
        "buildingNumber": "12",
        "postCode": "12000",
        "townName": "Praha 2",
        "country": "CZ"
    },
      "identification": {
        "privateIdentification": {
          "other": {
            "identification": "1245789528",
            "schemeName": {
              "proprietary": "ID CARD",
              "issuer": "OU Praha 2,CZ"
          }
        }
      }
    }
  },
    "debtorAccount": {
      "identification": {
        "iban": "CZ7508000000002108589434"
    },
      "currency": "CZK"
  },
    "creditorAgent": {
      "financialInstitutionIdentification": {
        "bic": "GIBAATWWXXX"
    }
  },
```

```
    "creditor": {
      "name": "1. wiena investment",
      "postalAddress": {
        "streetName": "Reisnerstraße",
        "buildingNumber": "20",
        "postCode": "1030",
        "townName": "Wiena",
        "country": "AT"
    }
  },
    "creditorAccount": {
      "identification": {
        "iban": "AT872011102000123456"
    }
  },
    "ultimateCreditor": {
      "name": "Franz Schubert",
      "postalAddress": {
        "streetName": "Dianagasse",
        "buildingNumber": "6",
        "postCode": "1030",
        "townName": "Wiena",
        "country": "AT"
    },
      "identification": {
        "privateIdentification": {
          "other": {
            "identification": "12356879131",
            "schemeName": {
              "proprietary": "passport",
              "issuer": "WIENA"
          }
        }
      }
    }
  },
    "purpose": {
```

```
        "proprietary": "sale of shares"
    }
}
```

5.5.3 Example of request Foreign payment within EEA

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
      "paymentIdentification": {
        "instructionIdentification": "MOJeID1234"
    },
      "paymentTypeInformation": {
        "instructionPriority": "NORM"
    },
      "amount": {
        "instructedAmount": {
          "value": 1245.44,
          "currency": "USD"
      }
    },
      "requestedExecutionDate": "2017-01-31",
      "debtorAccount": {
        "identification": {
          "iban": "CZ8601000000000161486937"
```

```
    },
       "currency": "CZK"
  },
    "creditorAgent": {
      "financialInstitutionIdentification": {
        "bic": "GIBACZPX"
    }
  },
    "creditor": {
      "name": "Bohumil Hrabal",
      "postalAddress": {
        "streetName": "Na Hrázi",
        "buildingNumber": "326/24",
        "postCode": "18000",
        "townName": "Praha 8",
        "country": "CZ"
    }
  },
    "creditorAccount": {
      "identification": {
        "iban": "CZ3908000000000204533335",
        "other": {
          "identification": "123/0800"
      }
    }
  },
    "remittanceInformation": {
      "unstructured": "fa 123546897"
  }
}
```


5.5.4 Example of request Foreign payment outside EEA

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
      "paymentIdentification": {
        "instructionIdentification": "IDTPP45678911"
  },
      "paymentTypeInformation": {
        "instructionPriority": "NORM"
  },
      "amount": {
        "instructedAmount": {
          "value": 1245.44,
          "currency": "GBP"
    }
  },
      "requestedExecutionDate": "2017-01-31",
      "debtorAccount": {
        "identification": {
          "iban": "CZ6330300000000000000123"
    },
        "currency": "CZK"
  },
      "creditorAgent": {
        "financialInstitutionIdentification": {
          "bic": "ABNYUS33",
          "name": "New York Commercial Bank",
          "postalAddress": {
            "streetName": "Merrick Avenue615",
            "buildingNumber": "61511590-6644",
```

```
            "postCode": "11590-6644",
            "townName": "WESTBURY, NY",
            "country": "USA"
        }
      }
    },
      "creditor": {
        "name": "First Hudson boat Inc.",
        "postalAddress": {
          "streetName": "Grand St",
          "buildingNumber": "1256",
          "postCode": "NY 11211",
          "townName": "Brooklyn ,NY",
          "country": "USA"
      }
    },
      "creditorAccount": {
        "identification": {
          "other": {
            "identification": "123456789"
        }
      }
    }
}
```


5.5.5 Example of correct response

Response **200 OK** Headers


```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body


```
{
      "paymentIdentification": {
        "instructionIdentification": "NOTPROVIDED",
        "transactionIdentification": "048885570000001020045"
   },
      "paymentTypeInformation": {
        "instructionPriority": "NORM",
        "serviceLevel": {
          "code": "DOMESTIC"
     }
   },
      "amount": {
        "instructedAmount": {
          "value": 10050.15,
          "currency": "CZK"
     }
   },
      "requestedExecutionDate": "2017-02-20",
      "debtorAccount": {
        "identification": {
          "iban": "CZ0708000000001019382023"
     },
        "currency": "CZK"
   },
      "creditorAccount": {
        "identification": {
          "iban": "CZ0708000000001019540081"
     },
        "currency": "CZK"
   },
      "remittanceInformation": {
        "structured": {
          "creditorReferenceInformation": {
```

```
            "reference": [
              "VS:501",
              "KS:9",
              "SS:1005"
            ]
          }
        }
    },
        "signInfo": {
          "state": "OPEN",
          "signId": "164298331754922"
    },
        "instructionStatus": "ACTC",
        "statusChangeInfo": "TM01"
}
```

5.5.6 Example of response to wrong call

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
```

Response **400 Bad Request** Body

```
{
  "errors": [
    {
      "error": "AC01"
    },
    {
```

```
      "error": "AM06",
      "scope": "amount.instructedAmount.value"
    },
    {
      "error": "AM03",
      "parameters": {
        "CURRENCY": "JPY",
        "CURRENCY_ALLOWED": "CZK"
      },
      "scope": "debtorAccount.currency"
    }
  ]
}
```

5.6 Example of request and response for Query for status of entered/initiated payment

Query URI used: GET /payments/048885570000001020045/status

5.6.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body:

```
{
      "instructionStatus": "ACTC"
}
```

5.6.2 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}
```

5.7 Example of request and response for Query for info on entered/initiated payment

Query URI used: GET /payments/048885570000001020045

5.7.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body:

```
{
      "paymentIdentification": {
        "instructionIdentification": "NOTPROVIDED",
        "transactionIdentification": "048885570000001020045"
  },
      "paymentTypeInformation": {
        "instructionPriority": "NORM",
        "serviceLevel": {
          "code": "DOMESTIC"
    }
  },
      "amount": {
        "instructedAmount": {
          "value": 10050.15,
          "currency": "CZK"
    }
```

```
  },
    "requestedExecutionDate": "2017-02-20",
    "debtorAccount": {
      "identification": {
        "iban": "CZ0708000000001019382023"
    },
      "currency": "CZK"
  },
    "creditorAccount": {
      "identification": {
        "iban": "CZ0708000000001019540081"
    },
      "currency": "CZK"
  },
    "remittanceInformation": {
      "structured": {
        "creditorReferenceInformation": {
          "reference": [
         "VS:501",
         "KS:9",
         "SS:1005"
        ]
      }
    }
  },
    "signInfo": {
      "state": "OPEN",
      "signId": "164298331754922"
  },
    "instructionStatus": "ACTC",
    "statusChangeInfo": "TM01"
}
```

5.7.2 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}
```

5.8 Example of request and response to delete unauthorized payments

Query URI used: DELETE /payments/048885570000001020045

5.8.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Response **201 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

5.8.2 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}
```

5.9 Example of request and response to generate payment authorization ID

Query URI used: POST /payments/048885570000001020045/sign

5.9.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

```
```

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body:

```
{
      "scenarios": [
     [
       "CODE of The One of the methods ..."
     ],
     [
       "CODE of The second method ..."
     ]
  ],
      "signInfo": {
        "state": "OPEN",
        "signId": "164298331754922"
  }
}
```

5.9.2 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
```

```

```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}
```

5.10 Example of request and response for Payment Authorization - Step I. Payment Authorization Detail

Query URI used: GET /payments/04888557000000102004 5/sign/164298331754922

5.10.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body:


```
{
      "scenarios": [
    [
      "CODE of The One of the methods ..."
    ],
    [
      "CODE of The second method ..."
    ]
  ],
      "signInfo": {
        "state": "OPEN",
        "signId": "164298331754922"
  }
}
```


5.10.2 Example of response to wrong call

5.10.2.1 Error 404

Response **404 Not Found** Headers


```
HTTP/1.1 404 Not Found
Content-Type: application/json
```


Response **404 Not Found** Body


```
{
  "errors": [
    {
```

```
      "error": "ID_NOT_FOUND",
      "scope": "signId"
    },
    {
      "error": "ID_NOT_FOUND",
      "scope": "paymentId"
    }
  ]
}
```

5.10.2.2 Error 400

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
```

Response **400 Bad Request** Body

```
{
  "errors": [
    {
      "error": "AUTH_LIMIT_EXCEEDED"
    }
  ]
}
```

5.11 Example of request and response for Payment Authorization - Step II. Payment Authorization Initiation - specific for each bank

Query URI used: POST /payments/04888557000000102004 5/sign/164298331754922

5.11.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
      "authorizationType": "SMS"
}
```

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body:

```
{
      "authorizationType": "USERAGENT_REDIRECT",
      "href": {
        "url": "http://www.bank.cz/authorization/164298331754922",
        "id": "164298331754922"
   },
      "method": "GET",
```

```
      "formData": {
        "SAMLRequest": "45sa4d4e4e4fds5f6s4df4sd6f",
        "relayState": "token"
    },
      "signInfo": {
        "state": "OPEN",
        "signId": "164298331754922"
    }
}
```


5.11.2 Example of response to wrong call

5.11.2.1 Error 404

Response **404 Not Found** Headers


```
HTTP/1.1 404 Not Found
Content-Type: application/json
```


Response **404 Not Found** Body


```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.11.2.2 Error 400

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
```

Response **400 Bad Request** Body

```
{
  "errors": [
    {
      "error": "AUTH_LIMIT_EXCEEDED"
    }
  ]
}
```

5.12 Example of request and response for Payment Authorization - Step III. Payment Authorization Finalization - specific for each bank

Query URI used: PUT /payments/048885570000001020045/sign/164298331754922

5.12.1 Example of request

Request headers:

```
Content-Type: application/json
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
      "authorizationType": "SMS",
      "oneTimePassword": "12345"
}
```

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body:

```
{
      "state": "DONE",
      "pollInterval": 5000
}
```

5.12.2 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
```

Response **404 Not Found** Body


```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```