

# The Sandbox Manual on the Payment Initiation Service

## Change log

<b>Date</b>	<b>Version</b>	<b>Description</b>
26.02.2020	5	Document and screen updates
21.05.2020	6	Added Standing Order (valid from 27.5.2020)
16.02.2023	7	Added update PISP v2

## Contents

1. Payment Initiation Service (PIS).....	4
2. Payment Initiation Service Calling API Sandbox .....	4
3. Issuing a Certificate.....	4
4. Error reporting .....	4
5. Accounts/Sandbox test data.....	5
6. Procedure for generating PISP key/token .....	5
7. Access to the application through the API console .....	8
“New Payment” PIS mock calling for the purpose of testing.....	8
“Payment status information” PIS mock calling for the purpose of testing .....	12
“Payment Authorization” PIS mock calling for the purpose of testing.....	15
Balance check” PIS mock calling for the purpose of testing .....	18
6. Access to the application through direct calling.....	19
New Payment – Payment Initiation (POST /my/payments) .....	19
New Payment – Payment Initiation report elements.....	21
New Payment – Payment Initiation response elements .....	31
Payment status information (GET /my/payments/{paymentId}/status) .....	33
Established/initiated Payment Status Report Elements .....	34
Payment detail information (GET /my/payments/{paymentId}).....	35
Payment authorisation Initiation – bank-specific (POST /my/payments/{paymentId}/sign/{signId }).....	38
Payment authorisation Initiation – Bank-Specific .....	39
Balance Check (POST /my/payments/balanceCheck) .....	40
MESSAGE ELEMENTS Query for Balance Check .....	41
MESSAGE ELEMENTS Response for Balance Check.....	41
Return codes for the parameter „response“ – Code set: .....	42

## 1. Payment Initiation Service (PIS)

Overview of implemented resources:

- Payment initiation
- Payment authorisation
- Established/initiated payment status
- Payment detail information

Unauthorised payment types for Sandbox:

- Standing payment orders
- Direct debit orders/mandates
- Payment orders batches
- Instant payments
- Cheque payments
- SEPA payments
- Foreign payments

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

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

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

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

## 2. Payment Initiation Service Calling API Sandbox

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

Any entities, not only the third parties with a PSD2 services licence, may access the API Sandbox. However, they must register at KB's API portal <https://api.kb.cz/portal/?tenant=api.kb.cz>. Failing this, they cannot utilize the Sandbox services. The procedure for registration is described in the document *API Sandbox Registration\_v1.doc*. Qualified PSD2 certificates issued by a qualified certification authority according to the EU QTSP list at <https://webgate.ec.europa.eu/tl-browser/>

## 3. Issuing a Certificate

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

## 4. Error reporting

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

PSD2 API: CZ, SK

Environment: Sandbox, Production

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

Request type

Date and time of the call

IP address

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

**Without the above values, it is not possible to solve the reported error.**

## 5. Accounts/Sandbox test data

BBAN	IBAN	Currency	Balance	Comments
900930427310227	CZ8501000900930427310227	CZK	9600,11	Only domestic payments
900930427430237	CZ0301000900930427430237	CZK	124001,01	Only domestic payments

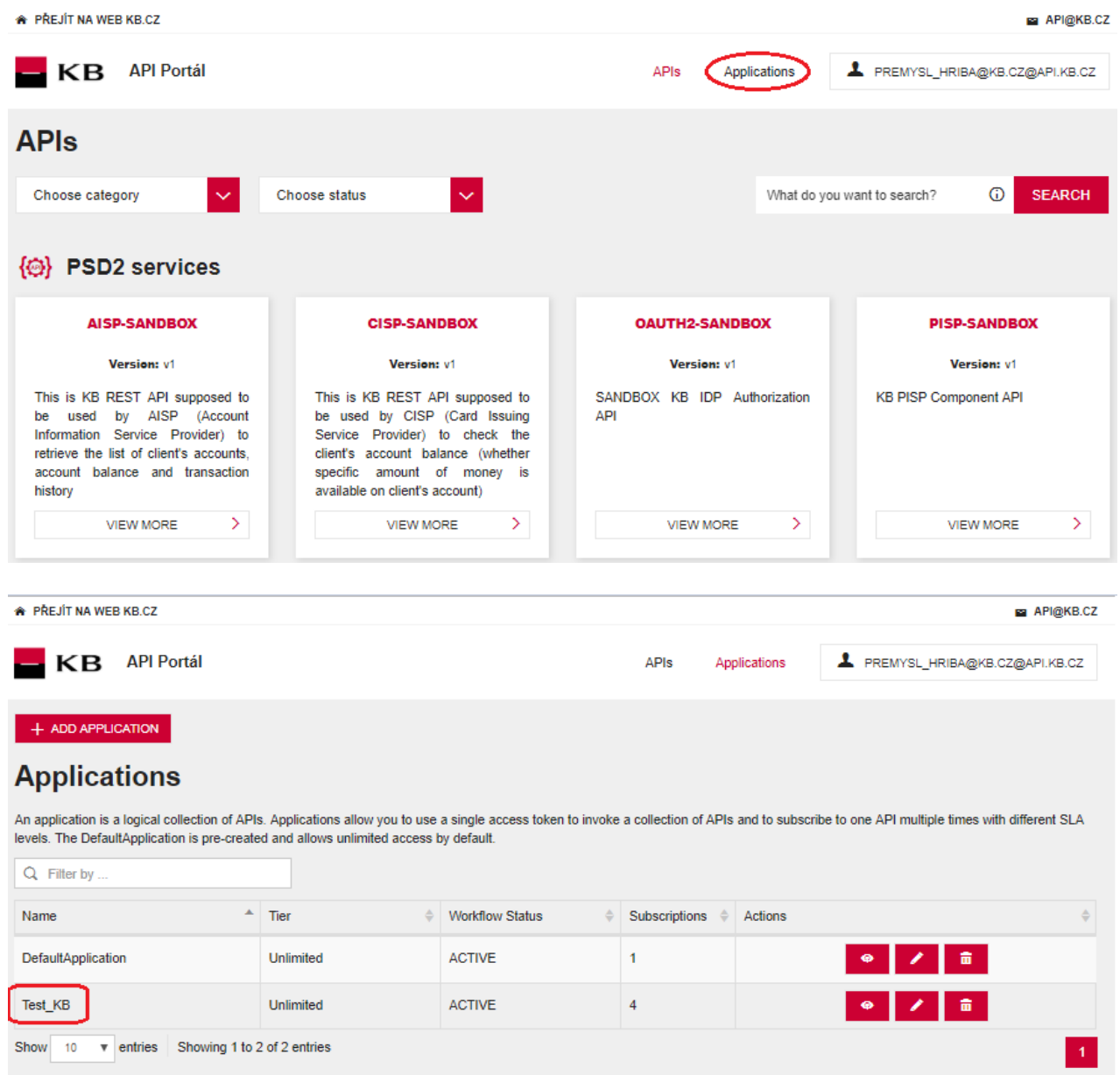
## 6. Procedure for generating PISP key/token

### Prerequisites for key/token generation

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

### Acces the application menu and select desired application







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



The screenshot shows the KB API Portal interface. The top navigation bar includes the KB logo, 'API Portál', and a user profile dropdown for 'PREMYSL\_HRIBA@KB.CZ@API.KB.CZ'. The 'Applications' menu item is highlighted with a red circle.

The main content area is divided into two sections:

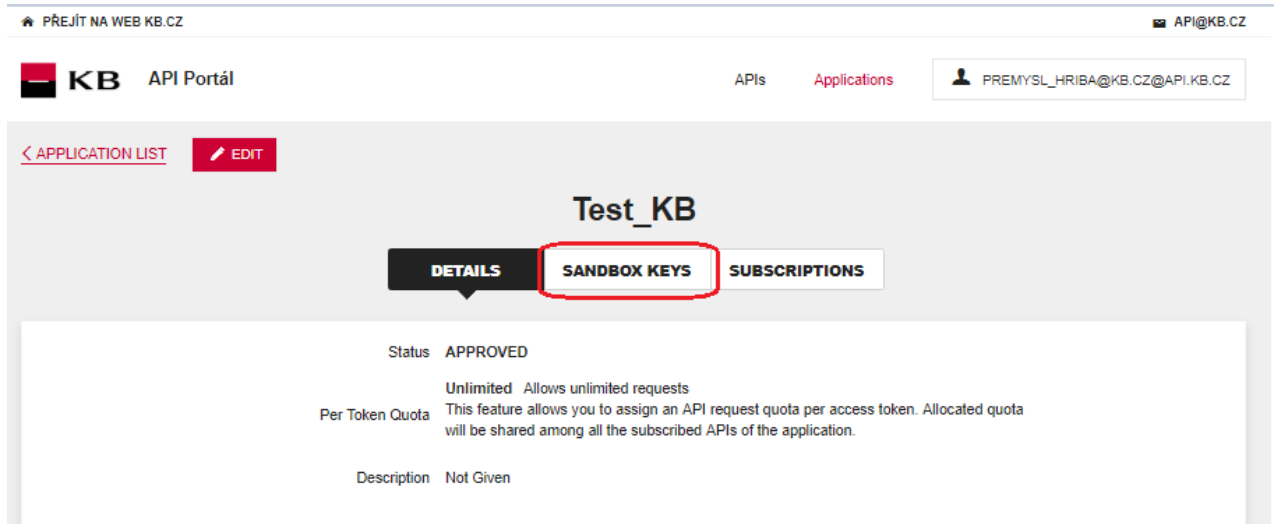
- APIs:** A search and filter interface for APIs. It includes dropdowns for 'Choose category' and 'Choose status', a search bar with the text 'What do you want to search?', and a 'SEARCH' button. Below this, there are four cards for PSD2 services:
  - AISP-SANDBOX:** Version: v1. Description: This is KB REST API supposed to be used by AISP (Account Information Service Provider) to retrieve the list of client's accounts, account balance and transaction history.
  - CISP-SANDBOX:** Version: v1. Description: This is KB REST API supposed to be used by CISP (Card Issuing Service Provider) to check the client's account balance (whether specific amount of money is available on client's account).
  - OAUTH2-SANDBOX:** Version: v1. Description: SANDBOX KB IDP Authorization API.
  - PISP-SANDBOX:** Version: v1. Description: KB PISP Component API.
- Applications:** A section with a '+ ADD APPLICATION' button. It contains a description: 'An application is a logical collection of APIs. Applications allow you to use a single access token to invoke a collection of APIs and to subscribe to one API multiple times with different SLA levels. The DefaultApplication is pre-created and allows unlimited access by default.' Below the description is a search filter 'Filter by ...' and a table of applications:
 

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

At the bottom of the Applications section, there is a 'Show 10 entries' dropdown and a 'Showing 1 to 2 of 2 entries' indicator. A red box highlights the 'Test\_KB' application in the table.

## Selection of application menu functionality

User selects „SANBOX KEYS“ in application menu



The screenshot shows the KB API Portal interface. At the top, there is a navigation bar with the KB logo, 'API Portál', and a user profile dropdown for 'PREMYSL\_HRIBA@KB.CZ@API.KB.CZ'. Below the navigation bar, there is a header for the application 'Test\_KB' with three menu items: 'DETAILS', 'SANDBOX KEYS' (highlighted with a red box), and 'SUBSCRIPTIONS'. The main content area displays the following details:

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

### **Generate certificate for PISP service**

The user Sandox may choose to generate a token for the PISP service, provided that the user is subscribed to the PISP service.

The user selects for key/token generation and generates the token using the „REGENERATE“ functionality.

[< APPLICATION LIST](#)
[EDIT](#)

## Test\_KB

[DETAILS](#)
[SANDBOX KEYS](#)
[SUBSCRIPTIONS](#)
[SHOW KEYS](#)

Consumer Key

.....

Consumer Secret

.....

Grant Types

The application can use the following grant types to generate Access Tokens. Based on the application requirement, you can enable or disable grant types for this application.

 Refresh Token

 SAML2

 Implicit

 Password

 IWA-NTLM

 Client Credential

 Code

Callback URL

https://www.kb.cz

[UPDATE](#)

Generating Access Tokens

The following cURL command shows how to generate an access token using the Password Grant type.

```
curl -k -d "grant_type=password&username=Username&password=Password" \
-H "Authorization: Basic Base64(consumer-key:consumer-secret)" \
https://api.kb.cz/token
```

In a similar manner, you can generate an access token using the Client Credential grant type with the following cURL command.

```
curl -k -d "grant_type=client_credentials" \
-H "Authorization: Basic Base64(consumer-key:consumer-secret)" \
https://api.kb.cz/token
```

Generate a Test Access Token

Access Token

.....

Above token has a validity period of 3600 seconds. If you want to regenerate this token, please select its scopes and validity period.

aisp : aisp.

**pisp : pisp.**

SELECT..

Validity period

3600

Seconds

[REGENERATE](#)

## 7. Access to the application through the API console

### “New Payment” PIS mock calling for the purpose of testing

The user chooses an operation he/she wishes to test. In this case, it is “New Payment”. A new payment can be established using this operation. The operation menu drops down after the user clicks on the “SHOW MORE” button.





[< GO BACK](#)

## PISP-Sandbox

Version: v1

Updated: 10/Dec/2019 13:08:04 PM CET

Status: PUBLISHED

KB PISP Component API

API CONSOLE

DOCUMENTATION

Try  Using

Set Request Header

Authorization : Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.

[Swagger \( /swagger.json \)](#)

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

### PISP

<b>POST</b>	New payment – payment initiation /payments	SHOW MORE ▾
<b>POST</b>	Payment Authorization – starting the particular authorization method /payments/{paymentId}/sign/{signId}	SHOW MORE ▾
<b>GET</b>	Payment status information /payments/{paymentId}/status	SHOW MORE ▾
<b>POST</b>	Balance check /payments/balanceCheck	SHOW MORE ▾

Applications

Tiers


SUBSCRIBE

## Filling in the required fields of the “New Payment” operation

The user wishing to initiate a new payment fills in all fields with values in an appropriate format. If everything is done properly and there is no other reason why the payment should not be made (e.g., insufficient account balance), he/she receives a report on the execution of the payment. If any of mandatory fields is not filled in, the report is not displayed and the blank fields are highlighted in red. For the “paymentRequest” field, just click on the displayed example to copy it to the appropriate field.

[PŘEJÍT NA WEB KB.CZ](#)
API@KB.CZ

---


**API Portál**
APIs
Applications
PREMYSL\_HRIBA@KB.CZ@API.KB.CZ

---

[GO BACK](#)

### PISP-Sandbox

Version: v1 | Updated: 10/Dec/2019 13:08:04 PM CET | Status: PUBLISHED

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try  Using

Set Request Header  

Swagger (/swagger.json)

**PISP** Show/Hide | List Operations | Expand Operations

POST New payment – payment initiation /payments SHOW LESS ^

**Required Scopes**

Key	Name
plsp	plsp

**Response Class (Status 200)**  
successful operation

Model | **Example Value**

```

{
  "paymentRequest": {
    "paymentIdentification": {
      "instructionIdentification": "NOTPROVIDED"
    },
    "paymentTypeInformation": {
      "instructionPriority": "NORM"
    },
    "amount": {
      "instructedAmount": {
        "value": "10.00",
        "currency": "CZK"
      }
    }
  }
}

```

Response Content Type

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
x-request-id	<input type="text"/>	External Request ID	header	string
TPP-Name	(required)	Transaction initiator name	header	string

paymentRequest  New payment request body

Parameter content type:

**Model | Example Value**

```

{
  "paymentIdentification": {
    "instructionIdentification": "NOTPROVIDED"
  },
  "paymentTypeInformation": {
    "instructionPriority": "NO
RM"
  },
  "amount": {
    "instructedAmount": {
      "value": "10.00",

```

## “New Payment” operation error message

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

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Input parameter is invalid	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_400",       "scope": "x-request-id",       "message": "Value of parameter x-request-id is wrong"     }   ] }</pre>	
401	Missing certificate or access token	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>	
403	Invalid certificate or token	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>	
415	Invalid message charset	Model   Example Value	
		<pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre>	
500	Unexpected error occurred	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>	


**TRY IT OUT**

## “Payment status information” PIS mock calling for the purpose of testing

The user chooses an operation he/she wishes to test. In this case, it is “Payment status information”. This operation will view the payment status. It is an established payment that has not yet been authorized by the client or has already been authorized and PISP queries its status (GET). The operation menu drops down after the user clicks on the "SHOW MORE" button.

[PŘEJÍT NA WEB KB.CZ](#)
API@KB.CZ

---


**API Portál**
APIs
Applications
PREMYSL\_HRIBA@KB.CZ@API.KB.CZ

---

[< GO BACK](#)

### PISP-Sandbox

Version: v1 | Updated: 10/Dec/2019 13:08:04 PM CET | Status: PUBLISHED

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try  Using

Set Request Header

Authorization : Bearer

[Swagger \( /swagger.json \)](#)

#### PISP

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

<b>POST</b>	New payment – payment initiation <a href="#">/payments</a>	<a href="#">SHOW MORE</a> ▾
<b>POST</b>	Payment Authorization – starting the particular authorization method <a href="#">/payments/{paymentId}/sign/{signId}</a>	<a href="#">SHOW MORE</a> ▾
<b>GET</b>	Payment status information <a href="#">/payments/{paymentId}/status</a>	<a href="#">SHOW MORE</a> ▾
<b>POST</b>	Balance check <a href="#">/payments/balanceCheck</a>	<a href="#">SHOW MORE</a> ▾

**Applications**  **Tiers**  SUBSCRIBE

## Filling in the required fields of the “Payment status information” operation

A user wishing to view the status of a particular payment fills in all fields with values in an appropriate format. If everything is done properly, information on the given payment is displayed. If any of mandatory fields is not filled in, the report is not displayed and the blank fields are highlighted in red.

GET

Payment status information  
/payments/{paymentId}/status

SHOW LESS ^

**Response Class (Status 200)**  
successful operation

Model | Example Value

```
{
  "instructionStatus": "ACTC",
  "errorInfo": {
    "error": "string",
    "parameters": {},
    "scope": "string",
    "message": "string"
  }
}
```

Response Content Type  v

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
x-request-id	<input type="text"/>	External Request ID	header	string
TPP-Name	<input type="text"/>	Transaction initiator name	header	string
paymentId	<input style="border: 1px solid red;" type="text" value="(required)"/>	Unique bank transaction identification • example: 'WU000024R20'	path	string

## “Payment status information” operation error message

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

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Input parameter is invalid	Model   Example Value <pre>{   "errors": [     {       "error": "ERR_CODE_400",       "scope": "x-request-id",       "message": "Value of parameter x-request-id is wrong"     }   ] }</pre>	
401	Missing certificate or access token	Model   Example Value <pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>	
403	Invalid certificate or token	Model   Example Value <pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>	
404	Id does not exist	Model   Example Value <pre>{   "errors": [     {       "error": "ID_NOT_FOUND",       "message": "Parameter paymentId not found"     }   ] }</pre>	
415	Invalid message charset	Model   Example Value <pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre>	
500	Unexpected error occurred	Model   Example Value <pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>	

**TRY IT OUT**

## “Payment Authorization” PIS mock calling for the purpose of testing

The user chooses an operation he/she wishes to test. In this case, it is “Payment Authorization”. This operation will start the specific authorization methods. The operation menu drops down after the user clicks on the "SHOW MORE" button.

🏠 PŘEJÍT NA WEB KB.CZ ✉ API@KB.CZ

---

**API Portál** APIs Applications 👤 PREMYSL\_HRIBA@KB.CZ@API.KB.CZ

---

[← GO BACK](#)

### PISP-Sandbox

Version: v1 | Updated: 10/Dec/2019 13:08:04 PM CET | Status: PUBLISHED

KB PISP Component API

API CONSOLEDOCUMENTATION

Try  Using  Key

Set Request Header

Authorization : Bearer

[Swagger \( /swagger.json \)](#)

#### PISP

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

<span style="background-color: #f1c40f; padding: 2px 5px; font-weight: bold;">POST</span>	New payment – payment initiation <a href="#">/payments</a>	<a href="#">SHOW MORE</a> <span style="font-size: 0.7em;">▼</span>
<span style="background-color: #f1c40f; padding: 2px 5px; font-weight: bold;">POST</span>	Payment Authorization – starting the particular authorization method <a href="#">/payments/{paymentId}/sign/{signId}</a>	<a href="#">SHOW MORE</a> <span style="font-size: 0.7em;">▼</span>
<span style="background-color: #f1c40f; padding: 2px 5px; font-weight: bold;">GET</span>	Payment status information <a href="#">/payments/{paymentId}/status</a>	<a href="#">SHOW MORE</a> <span style="font-size: 0.7em;">▼</span>
<span style="background-color: #f1c40f; padding: 2px 5px; font-weight: bold;">POST</span>	Balance check <a href="#">/payments/balanceCheck</a>	<a href="#">SHOW MORE</a> <span style="font-size: 0.7em;">▼</span>

**Applications**

**Tiers**

SUBSCRIBE

## Filling in the required fields of the “Payment Authorization” operation“

A user wishing to start the authorization process fills in all fields with values in an appropriate format. If everything is done properly, an overview of the values necessary for the completion of the authorization will be displayed. If any of mandatory fields is not filled in, the report is not displayed and the blank fields are highlighted in red. For the “authInitiationRequest” field, just click on the displayed example to copy it to the appropriate field.

**Required Scopes**

Key	Name
plsp	plsp

**Response Class (Status 200)**  
successful operation

Model | **Example Value**

```
{
  "authorizationType": "USERAGENT_REDIRECT",
  "href": {
    "url": "string",
    "id": "string"
  },
  "method": "string",
  "formData": {
    "SAMLRequest": "string",
    "relayState": "string"
  },
  "signInfo": {
    .....
  }
}
```

Response Content Type

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
x-request-id	<input type="text"/>	External Request ID	header	string
TPP-Name	<input type="text"/>	Transaction initiator name	header	string
paymentId	<input type="text" value="(required)"/>	Unique bank transaction identification example: 'WU000024R20'	path	string
signId	<input type="text" value="(required)"/>	Identifier of the change	path	string
authInitiationRequest	<input type="text" value="(required)"/>	Initiation request - identifier of the authorization method and back URL	body	string

Parameter content type:

Model | **Example Value**

```
{
  "authorizationType": "USERAGENT_REDIRECT",
  "backUrl": "string"
}
```



## “Payment Authorization” operation error message

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

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Input parameter is invalid	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_400",       "scope": "x-request-id",       "message": "Value of parameter x-request-id is wrong"     }   ] }</pre>	
401	Missing certificate or access token	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>	
403	Invalid certificate or token	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>	
404	Id does not exist	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ID_NOT_FOUND",       "message": "Parameter paymentId not found"     }   ] }</pre>	
415	Invalid message charset	Model   Example Value	
		<pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre>	
500	Unexpected error occurred	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>	


TRY IT OUT

## Balance check” PIS mock calling for the purpose of testing

The user chooses an operation he/she wishes to test. In this case, it is “Balance check”. The operation allows to obtain information about the availability of funds on the client's accounts.

[PŘEJÍT NA WEB KB.CZ](#)
API@KB.CZ

---


**API Portál**
APIs
Applications
PREMYSL\_HRIBA@KB.CZ@API.KB.CZ

---

[< GO BACK](#)

### PISP-Sandbox

Version: v1 | Updated: 10/Dec/2019 13:08:04 PM CET | Status: PUBLISHED

KB PISP Component API

API CONSOLE

DOCUMENTATION

Try TEST\_KB ▼ Using SANDBOX ▼ Key

Set Request Header

Authorization : Bearer
eyJ0eXAI0iJKV1QiLCJhbGciOiJSUzI1NiJ9.

[Swagger \(/swagger.json\)](#)

#### PISP

Show/Hide | List Operations | Expand Operations

<span style="background-color: #dc3545; color: white; padding: 2px 5px; font-size: 0.7em;">POST</span>	New payment – payment initiation <code>/payments</code>	SHOW MORE <span style="color: red;">▼</span>
<span style="background-color: #dc3545; color: white; padding: 2px 5px; font-size: 0.7em;">POST</span>	Payment Authorization – starting the particular authorization method <code>/payments/{paymentId}/sign/{signId}</code>	SHOW MORE <span style="color: red;">▼</span>
<span style="background-color: #dc3545; color: white; padding: 2px 5px; font-size: 0.7em;">GET</span>	Payment status information <code>/payments/{paymentId}/status</code>	SHOW MORE <span style="color: red;">▼</span>
<span style="background-color: #dc3545; color: white; padding: 2px 5px; font-size: 0.7em;">POST</span>	Balance check <code>/payments/balanceCheck</code>	SHOW MORE <span style="color: red;">▼</span>

**Applications**

DEFAULTAPPLICATION ▼

**Tiers**

UNLIMITED ▼

SUBSCRIBE

## Filling in the required fields of the “Balance Check” operation

### Response Class (Status 200)

successful operation

Model Example Value

```
{
  "responseIdentification": 987654,
  "exchangeIdentification": "103149078",
  "response": "APPR"
}
```

Response Content Type

### Parameters


Parameter	Value	Description	Parameter Type	Data Type
x-request-id	<input type="text"/>	External Request ID	header	string
TPP-Name	<input type="text" value="(required)"/>	Transaction initiator name	header	string

Parameter	Value	Description	Parameter Type	Data Type
balanceCheckRequest	<input type="text" value="(required)"/>	Query for Balance Check	body	string

Parameter content type:

Model Example Value

```
{
  "exchangeIdentification": "103149078",
  "card": {
    "cardHolderName": "Jan Novak",
    "maskedPAN": "1234*****6789"
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ3601009009300"
    }
  }
}
```



## 6. Access to the application through direct calling

### New Payment – Payment Initiation (POST /my/payments)

Resource for establishing a new payment.

#### Resource characteristics

- URI:** /my/payments
- HTTP Method:** POST
- Request URL:** <https://api.kb.cz/sandbox/pisp/v2/my/payments>
- Authorization:** the request **requires** an authorisation by the user/client as part of the API call.
- Certification:** the request **requires** the use of the third party qualified certificate as part of establishing two-way TSL communication with the server. The third party is identified by verifying the validity and content of this certificate.

**Pagination:** no  
**Sorting:** no  
**Filtering:** no

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

Request header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
<b>Content-Type</b>	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
<b>Authorisation</b>	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
<b>Date</b>	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
<b>X-request-id</b>	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
<b>User-involved</b>	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
<b>TPP-Name</b>	Text	Yes	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'. In this field, only characters with no diacritics are supported.
<b>TPP-Identification</b>	Text	No	The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'

Response header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
<b>Content-Type</b>	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the <b>application/json</b> format is primarily supported.

For the content of the request and response call POST see Chapter 1.1 New Payment – Payment Initiation REPORT ELEMENTS.

Table – CBA-standard defined error codes for the payment initiation POST service

HTTP STATUS CODE	ERROR CODE	PURPOSE
<b>401</b>	UNAUTHORISED	Missing certificate.
<b>403</b>	FORBIDDEN	Calling of a method that does not correspond to the licence, or invalid certificate.
<b>400</b>	FIELD_MISSING	Missing mandatory field in the request .
<b>400</b>	FIELD_INVALID	The field value is not valid.
<b>400</b>	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
<b>400</b>	AC03	[InvalidCreditorAccountNumber] – the creditor account number is closed/blocked, or credit transactions are not allowed for the given account type, or the creditor account number is given in an invalid format (note: validated for in-house payments only).

400	AC10	[InvalidDebtorAccountCurrency] – the declared account currency does not correspond to the currency in which the client's account held with the bank under the given number is denominated (the account currency is optional; however, it should be specified in the case of multicurrency accounts – Raiffeisenbank).
400	AC12	[InvalidAccountType] – the account type does not match allowed account types (e.g., a non-paying account).
403	AG01	[TransactionForbidden] – an absent consent to access to the account balance check.
400	AM05	[Duplication] – a duplication occurred. A universal code for a CISP duplicate query (validated e.g. by ČSOB) or a duplicate payment via PISP (non-unique payment reference).
400	AM11	[InvalidTransactionCurrency] – the request contains a currency that is not traded/supported.
400	AM12	[InvalidAmount] – a wrong amount, e.g., too low or high amount or a wrong number format in terms of the number of decimal places according to ISO 4217.
400	FF01	[Invalid File Format] – an invalid JSON format or other technical problem with the query processing.
400	BE19	[InvalidChargeBearerCode] – an invalid charge type for the given transaction type.
400	DT01	[InvalidDate] – "Invalid Date" – see below *
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RC07	[InvalidCreditorBICIdentifier] – an invalid SWIFT / BIC code of the creditor's bank.
400	RC10	[InvalidCreditorClearingSystemMemberIdentifier] – an invalid creditor's bank code identification.
400	RR03	[MissingCreditorNameOrAddress] – required data concerning the creditor's name or address is missing fully or partly in the field. If the data is given in a wrong format, the FIELD_INVALID error code is used.
400	RR10	[InvalidCharacterSet] – an invalid character set in the request.

#### New Payment – Payment Initiation report elements

##### Considered payment types

PAYMENT CODE	SERVICE LEVEL CODE	DESCRIPTION
TUZEM	DMCT	Domestic payment
SEPA	ESCT	SEPA payment
ZPL	XBCT	Cross-border payment within the EEA, Cross-border payment outside the EEA

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	paymentIdentification	[1..1]	ALL	PaymentIdentification1	Payment identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification

++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA ZPL	Max35Text	End To End identification
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInformation	[0..1]	ALL	PaymentTypeInformation19	Information about the payment type
++	instructionPriority	[0..1]	ALL	Priority2Code	Priority of the instruction
++	serviceLevel	[0..0]	ALL	ServiceLevel8CZ	Service level
+++	code	[0..0]	ALL	ExternalServiceLevel1Code	Service level code
++	categoryPurpose	[0..0]	ALL	CategoryPurpose1Choice	Payment purpose category
+++	code	[0..0]	ALL	ExternalCategoryPurpose1Code	Payment purpose category code
+++	proprietary	[0..0]	ALL	Max35Text	Payment purpose category in the free format
+	amount	[1..1]	ALL	TUZEM-AmountType3CZ SEPA- AmountType3CZ EHP-AmountType3Choice NONEHP-AmountType3Choice	Amount
++	instructedAmount	[1..1]	ALL	CurrencyAndAmount	Instruction currency and amount
+++	value	[1..1]	ALL	Amount	Transfer amount
+++	currency	[1..1]	ALL	CurrencyCode	Transfer currency
++	equivalentAmount	[0..0]	ALL	CurrencyAndAmount	Equivalent currency and amount
+++	value	[0..0]	ALL	Amount	Equivalent transaction amount
+++	currency	[0..0]	ALL	CurrencyCode	Currency of the equivalent transaction amount
+	requestedExecutionDate	[0..1]	ALL	ISODate	Requested execution date of the payment
+	exchangeRateInformation	[0..0]	ALL	ExchangeRateInformation1	Contractual rate
++	exchangeRate	[0..0]	ALL	BaseOneRate	Agreed exchange rate
++	rateType	[0..0]	ALL	ExchangeRateType1Code	Type of the agreed exchange rate
++	contractIdentification	[0..0]	ALL	Max35Text	Identifier of the use of the agreed exchange rate
+	chargeBearer	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	ChargeBearerType1Code	Charge bearer
+	chargesAccount	[0..0]	ALL	CashAccount16CZ	Charges account
++	identification	[0..0]	ALL	AccountIdentification4ChoiceCZ	Charges account number identification
+++	iban	[0..0]	ALL	IBAN2007Identifier	Charges account IBAN number
++	currency	[0..0]	ALL	CurrencyCode ISO 4217	Charges account currency
+	ultimateDebtor	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PartyIdentification32CZ1	Ultimate debtor

++	name	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max70Text	Ultimate debtor's name
++	postalAddress	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PostalAddress6CZ	Ultimate debtor's postal address
+++	streetName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max70Text	Ultimate debtor's street
+++	buildingNumber	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max16Text	Ultimate debtor's building number
+++	postCode	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max16Text	Ultimate debtor's Postal Code
+++	townName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Ultimate debtor's town/city
+++	country	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	CountryCode ISO3166	Ultimate debtor's country
+++	addressLine	[0..0] [0..2] [0..0]	TUZEM SEPA ZPL	Max70Text	Unstructured record of the ultimate debtor's address
++	identification	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Party6Choice	Ultimate debtor's identification
+++	organisationIdentification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	OrganisationIdentification 4CZ	Unique identification of the ultimate debtor as an organization/ legal person. Either organisationIdentification or privateIdentification
++++	bicOrBei	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	BICIdentifier	Identification of the ultimate debtor as an organization/legal person in the form of the BIC or BEI code.
++++	other	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	GenericOrganisationIdentification1	Other identification of the ultimate debtor as an organization/legal person.

+++++	identification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Other identification of the ultimate debtor as an organization/legal person in the unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	OrganisationIdentificationSchemeName1CZ	Type of the document used for the identification of the ultimate debtor as an organization/legal person.
+++++	proprietary	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Type of the document used for the identification of the ultimate debtor as an organization/legal person in the free text format.
+++++	issuer	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Issuer of the document used for the identification of the ultimate debtor as an organization/legal person.
+++	privateIdentification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	PersonIdentification5CZ	Unique identification of the ultimate debtor as a natural person. Either organisationIdentification or privateIdentification
++++	other	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	GenericPersonIdentification1	Other identification of the ultimate debtor as a natural person in the unstructured form.
+++++	identification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Other identification of the ultimate debtor as a natural person in the unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PersonIdentificationSchemeName1Choice	Type of the document used for the identification of the ultimate debtor as a natural person.
+++++	proprietary	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Type of the document used for the identification of the ultimate debtor as a natural person in the free text format.
+++++	issuer	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Issuer of the document used for the identification of the ultimate debtor as a natural person.
+	debtor	[0..0]	ALL	PartyIdentification32CZ2	Debtor
++	name	[0..0]	ALL	Max70Text	Debtor's name
++	postalAddress	[0..0]	ALL	PostalAddress6CZ	Debtor's postal address
+++	streetName	[0..0]	ALL	Max70Text	Street name used in the debtor's postal address.
+++	buildingNumber	[0..0]	ALL	Max16Text	Building number used in the debtor's postal address.
+++	postCode	[0..0]	ALL	Max16Text	Postal code used in the debtor's postal address.



+++	townName	[0..0]	ALL	Max35Text	Town name used in the debtor's postal address.
+++	country	[0..0]	ALL	CountryCode ISO3166	Country name used in the debtor's postal address.
+++	addressLine	[0..0]	ALL	Max70Text	Unstructured record of the debtor's postal address.
+	debtorAccount	[1..1]	ALL	CashAccount16CZ	Debtor's account
++	identification	[1..1]	ALL	AccountIdentification4ChoiceCZ	Debtor's account identification
+++	iban	[1..1]	ALL	IBAN2007Identifier	Debtor's account number in the IBAN format
+++	other	[0..0]	ALL	GenericAccountIdentification1CZ	Debtor's account number in other format
++++	identification	[0..0]	ALL	Max34Text	Debtor's account number in the local BBAN format
++	currency	[0..1]	ALL	CurrencyCode ISO 4217	Debtor's account currency
+	intermediaryAgent1	[0..0]	ALL	BranchAndFinancialInstitutionIdentification4CZ	Intermediary bank 1
++	financialInstitutionIdentification	[0..0]	ALL	FinancialInstitutionIdentification7CZ	Financial institution identification
+++	bic	[0..0]	ALL	BICIdentifier	BIC / SWIFT bank code
+++	clearingSystemMemberIdentification	[0..0]	ALL	ClearingSystemMemberIdentification2	Clearing system member identification
++++	clearingSystemIdentification	[0..0]	ALL	ClearingSystemIdentification2Choice	Clearing system identification
+++++	code	[0..0]	ALL	ClearingSystemIdentification1Code	Code
+++++	proprietary	[0..0]	ALL	Max35Text	Free format
++++	memberIdentification	[0..0]	ALL	Max35Text	Member's clearing code
+++	name	[0..0]	ALL	Max70Text	Name
+++	postalAddress	[0..0]	ALL	PostalAddress6CZ	Postal address
++++	streetName	[0..0]	ALL	Max70Text	Street
++++	buildingNumber	[0..0]	ALL	Max16Text	Building number
++++	postCode	[0..0]	ALL	Max16Text	Postal Code
++++	townName	[0..0]	ALL	Max35Text	Town/City
++++	country	[0..0]	ALL	CountryCode ISO3166	Country
++++	addressLine	[0..0]	ALL	Max70Text	Unstructured record of the address
+++	other	[0..0]	ALL	GenericFinancialIdentification1CZ	Other identification of the bank
++++	identification	[0..0]	ALL	Max35Text	Bank's local code
+	creditorAgent	[0..0] [0..1] [1..1]	TUZEM SEPA ZPL	BranchAndFinancialInstitutionIdentification4CZ	Creditor's bank
++	financialInstitutionIdentification	[0..0] [0..1] [1..1]	TUZEM SEPA ZPL	FinancialInstitutionIdentification7CZ	Identification of the financial institution
+++	bic	[0..0] [1..1] [0..1]	TUZEM SEPA ZPL	BICIdentifier	BIC / SWIFT bank code

+++	clearingSystemMemberIdentification	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	ClearingSystemMemberIdentification2	Clearing system member identification
++++	clearingSystemIdentification	[0..0] [0..0] [0..1] ... buď anebo některá z jiných variant identifikace banky příjemce	TUZEM SEPA ZPL	ClearingSystemIdentification2Choice	Clearing system identification
+++++	code	[0..0] [0..0] [1..1]... buď code anebo proprietary	TUZEM SEPA ZPL	ExternalClearingSystemIdentification1Code	Code
+++++	proprietary	[0..0] [0..0] [1..1]... buď code anebo proprietary	TUZEM SEPA ZPL	Max35Text	Free format
++++	memberIdentification	[0..0] [0..0] [1..1]	TUZEM SEPA ZPL	Max35Text	Member's clearing code
+++	name	[0..0] [0..0] [0..1]... buď anebo některá z jiných variant identifikace banky příjemce	TUZEM SEPA ZPL	Max70Text	Name
+++	postalAddress	[0..0] [0..0] [0..1]... buď anebo některá z jiných variant identifikace banky příjemce	TUZEM SEPA ZPL	PostalAddress6CZ	Postal address
++++	streetName	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max70Text	Street

++++	buildingNumber	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max16Text	Building number
++++	postCode	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max16Text	Postal Code
++++	townName	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max35Text	Town/City
++++	country	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	CountryCode ISO3166	Country
++++	addressLine	[0..0] [0..0] [0..2]	TUZEM SEPA ZPL	Max70Text	Unstructured record of the address
+++	other	[0..0] [0..0] [0..1]... buď anebo některá z jiných variant identifika ce banky příjemce	TUZEM SEPA ZPL	GenericFinancialIdentific ation1C	Other identification of the bank
++++	identification	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max35Text	Bank's local code
+	creditor	[0..0] [1..1] [1..1]	TUZEM SEPA ZPL	PartyIdentification32CZ2	Creditor
++	name	[0..0] [1..1] [1..1]	TUZEM SEPA ZPL	Max70Text	Creditor's name
++	postalAddress	[0..0] [0..1] [1..1]	TUZEM SEPA ZPL	PostalAddress6CZ	Postal address
+++	streetName	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	Max70Text	Street

+++	buildingNumber	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	Max16Text	Building number
+++	postCode	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	Max16Text	Postal Code
+++	townName	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	Max35Text	Town/City
+++	country	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	CountryCode ISO3166	Country
+++	addressLine	[0..0] [0..2] [0..2]	TUZEM SEPA ZPL	Max70Text	Unstructured record of the address
+	creditorAccount	[1..1]	ALL	CashAccount16CZ	Creditor's account
++	identification	[1..1]	ALL	AccountIdentification4Ch oiceCZ	Creditor's account identification
+++	iban	[1..1]	ALL	IBAN2007Identifier	Account number in the IBAN format
+++	other	[0..0] [0..0] [1..1]	TUZEM SEPA ZPL	GenericAccountIdentificat ion1CZ:	Account number in other format
++++	identification	[0..0] [0..0] [1..1]	TUZEM SEPA ZPL	Max34Text	Account number in the local BBAN format
++	currency	[0..1] [0..0] [0..0]	TUZEM SEPA ZPL	CurrencyCode ISO4217	Creditor's account currency
+	ultimateCreditor	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PartyIdentification32CZ1	Ultimate creditor
++	name	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max70Text	Name
++	postalAddress	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PostalAddress6CZ	Postal address

+++	streetName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max70Text	Street
+++	buildingNumber	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max16Text	Building number
+++	postCode	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max16Text	Postal Code
+++	townName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Town/City
+++	country	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	CountryCode ISO3166	Country
+++	addressLine	[0..0] [0..2] [0..0]	TUZEM SEPA ZPL	Max70Text	Unstructured record of the address
++	identification	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Party6Choice	Ultimate creditor's identification
+++	organisationIdentification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	OrganisationIdentification 4CZ	Unique identification of the ultimate creditor as an organization/ legal person. Either organisationIdentification or privateIdentification
++++	bicOrBei	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	BICIdentifier	Identification of the ultimate creditor as an organization/ legal person in the form of the BIC or BEI code.
++++	other	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	GenericOrganisationIdentification1	Other identification of the ultimate creditor as an organization/ legal person.
+++++	identification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Other identification of the ultimate creditor as an organization/ legal person in the unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	OrganisationIdentification SchemeName1CZ	Type of the document used for the identification of the ultimate creditor as an organization/legal person.

+++++	proprietary	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Type of the document used for the identification of the ultimate creditor as an organization/legal person in the free text format.
++++	issuer	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Issuer of the document used for the identification of the ultimate creditor as an organization/legal person.
+++	privateIdentification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	PersonIdentification5CZ	Unique identification of the ultimate creditor as a natural person. Either organisationIdentification or privateIdentification
++++	other	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	GenericPersonIdentification1	Other identification of the ultimate creditor as a natural person in the unstructured form.
+++++	identification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Other identification of the ultimate creditor as a natural person in the unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PersonIdentificationSchemeName1Choice	Type of the document used for the identification of the ultimate creditor as a natural person.
+++++	proprietary	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Type of the document used for the identification of the ultimate creditor as a natural person in the free text format.
+++++	issuer	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Issuer of the document used for the identification of the ultimate creditor as a natural person.
+	purpose	[0..0]	ALL	Purpose2Choice	Payment purpose
++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code
++	proprietary	[0..0]	ALL	Max35Text	Payment purpose in the free format
+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instruction for the next bank
+	remittanceInformation	[0..1]	ALL	RemittanceInformation5CZ	Information about the payment
++	unstructured	[0..1]	ALL	Max140Text, consisting of alphanumeric characters supported by CERTIS (CNB clearing), including supported special characters	Unstructured report for the creditor (see below*)

++	structured	[0..1] [0..0] [0..0]	TUZEM SEPA ZPL	StructuredRemittanceInformation7CZ	Structured message for the creditor – variable, specific, and constant symbol
+++	creditorReferenceInformation	[0..1] [0..0] [0..0]	TUZEM SEPA ZPL	CreditorReferenceInformation2CZ	Creditor reference information
++++	reference	[0..3] [0..0] [0..0]	TUZEM SEPA ZPL	CreditorReferenceInformation2CZ	VS, SS & KS values

TUZEM = domestic payment  
ZPL = cross-border payment  
EHP = EEA

\* If the field **remittanceInformation.structured.creditorReferenceInformation.reference** contains a variable, constant or specific symbol, they will be identified and stored separately in the respective fields in the PISP model.

- The **variable symbol** value is recorded as VS:max.10 digits (e.g. VS:3451859072).
- The **constant symbol** value is recorded as KS:max.10 characters (e.g. KS:0308).
- The **specific symbol** value is recorded as SS:max.10 digits (e.g. SS:8451201274).

JSON – example of an element:

"reference": "VS:123456\","KS:456789\","SS:879213546"

Note concerning the *remittanceInformation.unstructured* field: According to the Banking Association standard, this field may also contain information about VS, KS and SS; however, we treat any and all information contained therein as if it were a simple description of the payment (information for the creditor). Hence, no symbol parsing will take place here even if they occur here.

### New Payment – Payment Initiation response elements

The table only contains the elements that occur exclusively in the message response.

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	transactionIdentification	[1..1]	Max35Text	Established transaction identifier
+	serviceLevel	[1..1]	±	Service level (within the payment type)
++	code	[1..1]	Text	Type of the submitted payment
+	signInfo	[1..1]	±	Information about the status and id of the unauthorised transaction
++	state	[1..1]	StateCode	Information about the status of the transaction authorisation
++	signId	[0..1]	Text	Identifier of the authorising process of the particular transaction.
++	signInfo	[1..1]	Status Code set	Transaction status identifier.

**serviceLevel.code** element values – initiated payment type

CODE	DESCRIPTION
<b>DMCT</b>	[DoMestic Creidt Transfer] Domestic payment
<b>ESCT</b>	[SEPA Credit Transfer] – SEPA payment
<b>XBCT</b>	[Cross-Border Credit Transfer] – Cross border payment

**Payment status codes – StatusCode**

HTTP STATUS CODE	STATUS CODE	PURPOSE
<b>200</b>	ACTC	[AcceptedTechnicalValidation] – The authentication and syntactical and semantical validation are successful
<b>200</b>	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
<b>200</b>	ACWC	[AcceptedWithChange] – An instruction is accepted but a change will be made, such as date or remittance not sent



## Payment status information (GET /my/payments/{paymentId}/status)

A resource for viewing the payment status. It is an established payment that has not yet been authorised by the client or has been authorised and PISP sends a query about its status (GET).

The resource only returns information about transactions established through the mediation of a specific provider. Information on the provider is taken from the certificate, or from the licence type information.

The user authorisation of this resource is optional. Primarily, only a provider's valid certificate is required.

### Resource characteristics

**URI:** /payments/{paymentId}/status  
**HTTP Method:** GET  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v2/my/payments/{paymentId}/status>  
**Authorization:** the request **does not require** any authorisation by the user/client as part of the API call.  
**Certification:** the request **requires** the use of the third party qualified certificate as part of establishing two-way TSL communication with the server. The third party is identified by verifying the validity and content of this certificate.

**Pagination:** no  
**Sorting:** no  
**Filtering:** no

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

Request header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
<b>Content-Type</b>	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
<b>Date</b>	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
<b>X-request-id</b>	text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
<b>User-involved</b>	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
<b>TPP-Name</b>	Text	Yes	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'. In this field, only characters with no diacritics are supported.
<b>TPP-Identification</b>	Text	No	The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'

Path parameter request:

PARAMETER	TYPE	MANDATORY	PURPOSE
<b>paymentId</b>	Text	Yes	Identifier of the established payment.

Response header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
<b>Content-Type</b>	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the <b>application/json</b> format is primarily supported.

For the content of the request and response call POST see Chapter 2.1 Established/Initiated Payment Status REPORT ELEMENTS

CBA-standard defined error codes for the GET Status service of the Established/Initiated Payment

HTTP STATUS CODE	ERROR CODE	PURPOSE
<b>401</b>	UNAUTHORISED	Invalid/missing certificate = the provider has not been authenticated

404	TRANSACTION_MISSING	Calling of a method that does not match with the licence, or invalid certificate.
-----	---------------------	---

**Established/initiated Payment Status Report Elements**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	instructionStatus	[1..1]	PISP ALL	StatusCode	Established payment status

**Payment codes status – StatusCode**

HTTP STATUS CODE	STATUS CODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] – The 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	[AcceptedSettlementInProgress] – All preceding checks such as technical validation and customer profile 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 first agent to report to the debtor that the transaction has been completed. Warning: this status is provided for transaction status reasons, not for financial information. It can only be used after bilateral agreement
200	ACWC	[AcceptedWithChange] – An instruction is accepted but a change will be made, such as date or remittance not change

## Payment detail information (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  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v2/my/payments/{paymentId}>  
**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
<b>Content-Type</b>	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, <b>application/json</b> format is primarily supported.
<b>Authorization</b>	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.
<b>Date</b>	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
<b>X-request-id</b>	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
<b>User-involved</b>	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
<b>TPP-Name</b>	Text	Yes	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'. In this field, only characters with no diacritics are supported.
<b>TPP-Identification</b>	Text	No	The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15

Path parameter request:

PARAMETER	TYPE	MANDATORY	PURPOSE
paymentId	Text	Yes	Identifier of the established payment.

Parameters of the response header:

PARAMETER	TYPE	MANDATORY	PURPOSE
-----------	------	-----------	---------

<b>Content-Type</b>	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, <b>application/json</b> format is primarily supported.
---------------------	------	-----	--

Error codes defined for the service GET 6. Info on entered/initiated payment:

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



## Payment authorisation Initiation – bank-specific (POST /my/payments/{paymentId}/sign/{signId })

This resource serves for **starting a specific authorisation method** from a selected scenario.

The input is a JSON object containing the required authorisation method type - **CODE** and all elements specific for this step.

The output of this resource is an overview of values necessary for completing the authorisation.

**E.g., the response to the CODE corresponding to the federated authorisation will be URL and parameters for the redirection to the federated authorisation page.**

Further, e.g. the response to the CODE corresponding to the authorisation through the OTP code sent via SMS will only be a confirmation of the code sending. The sending itself is initiated by the bank.

### Resource characteristics

**URI:** my//payments/{paymentId}/sign/{signId}  
**HTTP Method:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v2/my/payments/{paymentId}/sign/{signId}>  
**Authorization:** the request **requires** an authorisation by the user/client as part of the API call.  
**Certification:** the request **requires** the use of the third party qualified certificate as part of establishing two-way TSL communication with the server. The third party is identified by verifying the validity and content of this certificate.

**Pagination:** no  
**Sorting:** no  
**Filtering:** no

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

Request header parameters:

PARAMETER	TYPE	MANDA-TORY	PURPOSE
<b>Content-Type</b>	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
<b>Authorisation</b>	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
<b>Date</b>	DateT ime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
<b>X-request-id</b>	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
<b>User-involved</b>	Boole an	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
<b>TPP-Name</b>	Text	Yes	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'. In this field, only characters with no diacritics are supported.
<b>TPP-Identification</b>	Text	No	The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'

Path parameter requestu:

PARAMETER	TYPE	MANDA-TORY	PURPOSE
<b>paymentId</b>	Text	Yes	Identifier of the established payment.
<b>signId</b>	Text	Yes	The unique identifier of the current transaction authorization

Response header parameters:

PARAMETER	TYPE	MANDA-TORY	PURPOSE
<b>Content-Type</b>	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the <b>application/json</b> format is primarily supported.

For the content of the request and response call POST see Chapter 3.1 Step II - Payment Authorisation Initiation – Bank-Specific – REPORT ELEMENTS

CBA-standard defined error codes for the payment authorisation initiation POST service:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = the user has not been authenticated
403	FORBIDDEN	Invalid/missing certificate = the provider has not been authenticated
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	This resource cannot be authorised in this manner

### Payment authorisation Initiation – Bank-Specific

#### Request parameters:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	PISP ALL	Text	The code of the required authorisation (from authorisation scenarios)
+	redirectURL	[1..1]	PISP ALL	Text	The URL link for the return of a response concerning the executed authorisation.

#### Response parameters:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	PISP ALL	±	The code of the required authorisation ( <i>from authorisation scenarios</i> )
+	href	[0..1]	PISP ALL	±	The reference for calling the federated authorisation
++	url	[1..1]	PISP ALL	Text	The URL link or package federated authorisation
++	id	[0..1]	PISP ALL	Text	The potential id for calling the federated authorisation
+	method	[0..1]	PISP ALL	Text	The method of the use of the href link for the federated authorisation.
+	formData	[0..1]	PISP ALL	±	An optional element. In the case of the POST method of the federated authorisation (authorisationType=USERAGENT_REDIRECT), the element contains the data for sending in the redirection to the federated authorisation.
++	SAMLRequest	[0..1]	PISP ALL	Text	An optional parameter. In the case of the POST method of the federated authorisation (authorisationType=USERAGENT_REDIRECT), the element contains the SAML request data.
++	relayState	[0..1]	PISP ALL	Text	An optional parameter. In the case of the POST method of the federated authorisation (authorisationType=USERAGENT_REDIRECT), the element contains the relay State for the return value.
+	signInfo	[1..1]	PISP ALL	±	Information about the instruction authorisation.
++	state	[1..1]	PISP ALL	Text	A status of the transaction authorisation in a format supported by the bank.
++	signId	[1..1]	PISP ALL	Text	A unique identifier of the current transaction authorisation.

## 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  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v2/my/payments/balanceCheck>  
**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
<b>Content-Type</b>	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.
<b>Authorization</b>	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type
<b>Date</b>	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
<b>X-request-id</b>	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
<b>User-involved</b>	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
<b>TPP-Name</b>	Text	Yes	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'. In this field, only characters with no diacritics are supported.
<b>TPP-Identification</b>	Text	No	The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'

Parameters of the response header:

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

The content of POST request and response for calling, please see Chapter 7.1. 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.
400	AC12	[InvalidAccountType] - account type does not match allowed account types (e.g., a non-paying 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.

#### 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
+	debtorAccount	[1..1]	±	Payer account
++	identification	[1..1]	±	Payer account identification
+++	iban	[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 (2 alphanumeric characters code version)	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

#### 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]	Max18Text	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:

<i>CODE</i>	<i>DESCRIPTION</i>
<b>APPR</b>	Enough funds on this account
<b>DECL</b>	Unsufficient funds on this account