

# 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)
03.06.2022	7	Added Batch payments

## Contents

1. Payment Initiation Service (PIS).....	4
2. Payment Initiation Service Calling API Sandbox .....	4
3. Issuing a Certificate.....	4
4. Definition of the Mock.....	4
5. Error reporting .....	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.....	17
“New Standing Order initiation” PIS mock calling for the purpose of testing .....	19
“Standing Orders Authorization - start” PIS mock calling for the purpose of testing .....	22
“Standing Order status” PIS mock calling for the purpose of testing .....	25
“Standing Order Detail” PIS mock calling for the purpose of testing .....	28
“New batch payment – instruction initiation” PIS mock calling for the purpose of testing .....	31
“Batch Authorization” PIS mock calling for the purpose of testing .....	33
“Batch payment status” PIS mock calling for the purpose of testing .....	35
6. Access to the application through direct calling.....	38
New Payment – Payment Initiation (POST /my/payments) .....	38
New Payment – Payment Initiation report elements.....	39
New Payment – Payment Initiation response elements .....	49
Established/Initiated Payment Status (GET /payments/{paymentId}/status) .....	51
Established/initiated Payment Status Report Elements .....	51
Step II – Payment Authorisation Initiation – bank-specific (POST /my/payments/{paymentId}/sign/{signId }).....	53
Step II – Payment Authorisation Initiation – Bank-Specific.....	54
Balance Check (POST /my/payments/balanceCheck) .....	55
MESSAGE ELEMENTS Query for Balance Check .....	56
MESSAGE ELEMENTS Response for Balance Check.....	56
Return codes for the parameter „response“ – Code set: .....	56
Standing order.....	57
New Payment – Payment Initiation (POST /my/payments) .....	57
New Payment – standing orders Initiation report elements .....	58
New Payment – Standing orders Initiation response elements .....	62
Standing order authorization – bank-specific (POST /my/standingorders/{transactionIdentification}/sign/{signId}) .....	63
Payment Authorisation initiation report elements .....	65
Standing order detail (GET /my/standingorders/{transactionIdentification}) .....	65
Elementy response zprávy Detail trvalého příkazu .....	66
Standing order status (GET /my/standingorders/{transactionIdentification})/status.....	70
Batch Payments – initiation (POST /my/batchpayments).....	72
Batch payment – initiation request elements.....	72
Batch payment – initiation response elements.....	72
Error codes defined for the batch payment initiation POST service .....	73
Batch payment authorization – initiation (POST /my/ batchpayments/{transactionIdentification}/sign/{signId}).....	74
Batch payment authorization - initiation request elements .....	75
Batch payment authorization – initiation response elements.....	75
Error codes defined for batch payment authorization POST service .....	76
Batch payment – status (GET/my/batchpayments/{transactionIdentification}/status).....	77
Batch payment status response elements.....	77
Error codes define for batch payment status GET service .....	78
7. PSD2 glossary – selected terms .....	78

## 1. Payment Initiation Service (PIS)

Overview of implemented resources:

- Payment initiation
  - o Step I. – authorised payment initiation
- Established/initiated payment status – it provides information about the current status of a payment while it is processed by KB
- Payment authorisation
  - o Step II – authorised payment initiation – this resource initiates the payment authorisation
- Standing order – initiation
- Standing order authorization
- Standing order detail and status
- Batch payment – initiation
- Batch payment authorization
- Batch payment status

Unauthorised payment types for Sanbox:

- Direct debit orders/mandates
- Instant payments
- Cheque 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. Definition of the Mock

Parameters of static calling are defined, which a third party cannot change within calling. Any change will result in the rejection of a request. Appropriate responses or, as the case may be, error codes are returned depending on the specimen requests used.

### 1. Payment initiation:

- Domestic remittance initiation (TPL) –click on the **Example Value** grey field to the right of the paymentRequest field to get the input value, i.e., the valid structure of the initiated payment. The Sandbox fills in these values into the payment entering field automatically.
- Domestic remittance payment initiation – invalid IBAN error 400 - AC02 - [InvalidDebtorAccountNumber], debtorAccount.identification.iban field
- Domestic remittance payment initiation – invalid amount error 400 - AM12 - [InvalidAmount], (amount.instructedAmount.value) field
- Domestic remittance payment initiation – invalid execution date error – the required date exceeds the actual date, 400 NARR - [Entered Execution date cannot exceed actual Business date], requestedExecutionDate field
- Domestic remittance payment initiation – invalid field formats errors – returns a collection of errors

### 2. Payment status:

- Payment status – **ZU000000LAP** must be filled in as an paymentID (transactionID) input value
- Payment status – non-existing paymentID (transactionID) error; if no ID is found, the following error is generated 404 \_NOT\_FOUND

## 5. 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.**

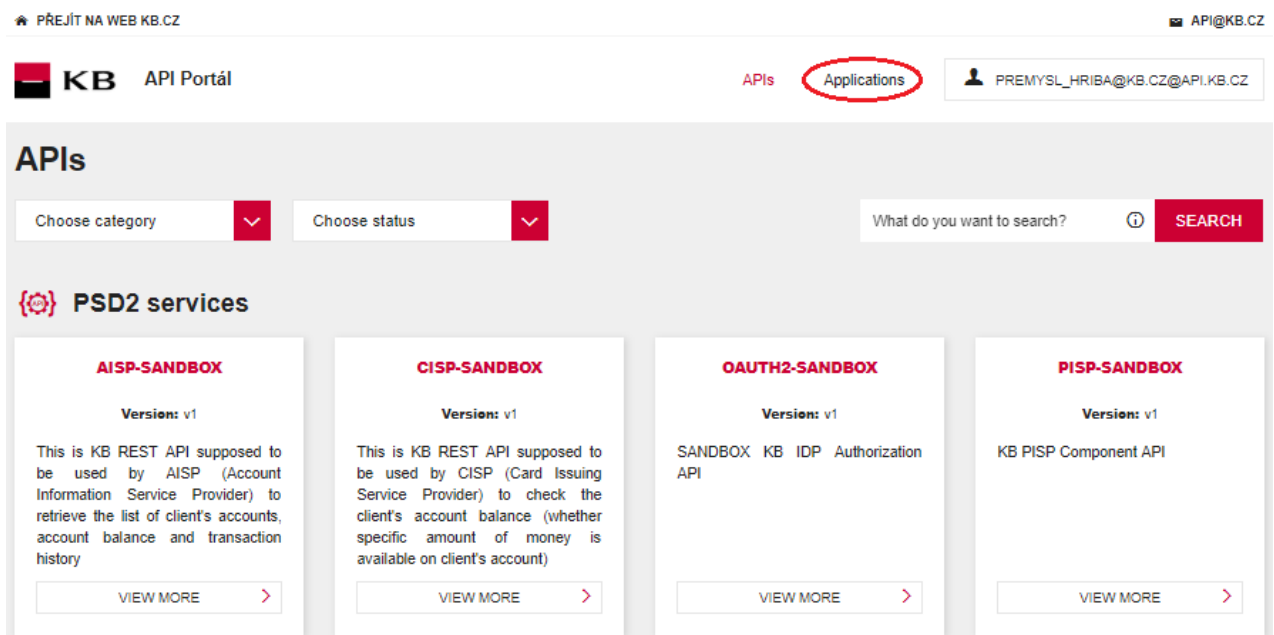
## 6. Procedure for generating PISP key/token

### Prerequisites for key/token generation

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

Access 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. 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'. A red circle highlights the 'Applications' link in the top navigation. Below the navigation bar, the main content area is titled 'APIs'. It features two dropdown menus for 'Choose category' and 'Choose status', and a search bar with the text 'What do you want to search?' and a 'SEARCH' button. Underneath, there is a section titled 'PSD2 services' with a gear icon. This section contains four service cards: 'AISP-SANDBOX', 'CISP-SANDBOX', 'OAUTH2-SANDBOX', and 'PISP-SANDBOX'. Each card displays its name, version (v1), a brief description of its function, and a 'VIEW MORE' button with a right-pointing arrow.

[+ ADD APPLICATION](#)

## Applications

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

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

Show  entries Showing 1 to 2 of 2 entries

1

### Selection of application menu functionality

User selects „SANBOX KEYS“ in application menu

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

## Test\_KB

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

Status APPROVED

Unlimited Allows unlimited requests

Per Token Quota This feature allows you to assign an API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.

Description Not Given

### **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.

PREJÍT NA WEB KB.CZ API@KB.CZ

**KB 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

POST New payment – payment initiation /payments SHOW LESS ^

**Required Scopes**

Key	Name
pisp	pisp

**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	<input type="text" value="(required)"/>	Transaction initiator name	header	string

paymentRequest

Parameter content type:

**New payment request**

body

Model **Example Value**

```

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

```

## “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": "ERR10",       "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 CONSOLEDOCUMENTATION

Try  ▼ Using  ▼

Set Request Header

Authorization : Bearer

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

#### PISP

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

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

**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:  ▼

**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   <b>Example Value</b> <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   <b>Example Value</b> <pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>	
403	Invalid certificate or token	Model   <b>Example Value</b> <pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>	
404	Id does not exist	Model   <b>Example Value</b> <pre>{   "errors": [     {       "error": "ID_NOT_FOUND",       "message": "Parameter paymentId not found"     }   ] }</pre>	
415	Invalid message charset	Model   <b>Example Value</b> <pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre>	
500	Unexpected error occurred	Model   <b>Example Value</b> <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

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

**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
pisip	pisip

**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
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.

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

### PISP

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

<span style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">POST</span>	New payment – payment initiation <code>/payments</code>	<a href="#">SHOW MORE</a> <span style="color: red;">▼</span>
<span style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">POST</span>	Payment Authorization – starting the particular authorization method <code>/payments/{paymentId}/sign/{signId}</code>	<a href="#">SHOW MORE</a> <span style="color: red;">▼</span>
<span style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">GET</span>	Payment status information <code>/payments/{paymentId}/status</code>	<a href="#">SHOW MORE</a> <span style="color: red;">▼</span>
<span style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">POST</span>	Balance check <code>/payments/balanceCheck</code>	<a href="#">SHOW MORE</a> <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

balanceCheckRequest



Model | Example Value

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

Parameter content type:



**“New Standing Order initiation” PIS mock calling for the purpose of testing**

The user chooses an operation he/she wishes to test. In this case, it is “New Standing Order initiation”. 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: 05/May/2020 09:11:54 AM CEST | Status: PUBLISHED

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try  Using  Key

Set Request Header

Authorization : Bearer  Swagger ( /swagger.json )

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

POST	New payment – payment initiation /payments	SHOW MORE ▾
POST	Payment Authorization – starting the particular authorization method /payments/{paymentId}/sign/{signId}	SHOW MORE ▾
GET	Payment status information /payments/{paymentId}/status	SHOW MORE ▾
POST	Balance check /payments/balanceCheck	SHOW MORE ▾
POST	<b>New Standing Order initiation</b> /standingorders	SHOW MORE ▾
POST	Standing Orders Authorization - start /standingorders/{transactionIdentification}/sign/{signId}	SHOW MORE ▾
GET	Standing Orders Status /standingorders/{transactionIdentification}/status	SHOW MORE ▾
GET	Standing Orders Detail /standingorders/{transactionIdentification}	SHOW MORE ▾

### Filling in the required fields of the “New Standing Order initiation” 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.

### Response Class (Status 200)

OK

Model | Example Value

```
{
  "standingOrderIdentification": {
    "instructionIdentification": 12312425,
    "transactionIdentification": 358745
  },
  "amount": {
    "instructedAmount": {
      "value": 10.08,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2022-02-19",
}
```

Response Content Type  ▼

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Payload	<input type="text" value="(required)"/>	Request Body	body	<div style="border: 2px solid red; padding: 5px;"> <p>Model   Example Value</p> <pre>{   "standingOrderIdentification": {     "instructionIdentification": 12312425   },   "amount": {     "instructedAmount": {       "value": 10.08,       "currency": "CZK"     }   }, }</pre> </div>
	Parameter content type: <input type="text" value="application/json"/> ▼			
x-request-id	<input type="text"/>	External Request ID	header	string
TPP-Name	<input type="text" value="(required)"/>	Transaction Initiator name	header	string

### “New Standing Order initiation” 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

### “Standing Orders Authorization - start” PIS mock calling for the purpose of testing

The user chooses an operation he/she wishes to test. In this case, it is “Standing Order Authorization - start”. 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.

[← GO BACK](#)

## PISP-Sandbox

Version: v1 | Updated: 05/May/2020 09:11:54 AM CEST | Status: PUBLISHED

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try  Using  Key

Set Request Header

Authorization : Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiU9.

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

### PISP

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

<b>POST</b>	New payment – payment initiation <code>/payments</code>	SHOW MORE ▾
<b>POST</b>	Payment Authorization – starting the particular authorization method <code>/payments/{paymentId}/sign/{signId}</code>	SHOW MORE ▾
<b>GET</b>	Payment status information <code>/payments/{paymentId}/status</code>	SHOW MORE ▾
<b>POST</b>	Balance check <code>/payments/balanceCheck</code>	SHOW MORE ▾
<b>POST</b>	New Standing Order initiation <code>/standingorders</code>	SHOW MORE ▾
<b>POST</b>	Standing Orders Authorization - start <code>/standingorders/{transactionIdentification}/sign/{signId}</code>	SHOW MORE ▾
<b>GET</b>	Standing Orders Status <code>/standingorders/{transactionIdentification}/status</code>	SHOW MORE ▾
<b>GET</b>	Standing Orders Detail <code>/standingorders/{transactionIdentification}</code>	SHOW MORE ▾

### Filling in the required fields of the “New Standing Order initiation” 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.

### Response Class (Status 200)


OK

Model | Example Value

```
{
  "standingOrderIdentification": {
    "instructionIdentification": 12312425,
    "transactionIdentification": 358745
  },
  "amount": {
    "instructedAmount": {
      "value": 10.08,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2022-02-19",
}
```

Response Content Type  

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Payload	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/json"/> 	Request Body	body	<div style="border: 2px solid red; padding: 5px;"> <p>Model   Example Value</p> <pre>{   "standingOrderIdentification": {     "instructionIdentification": 12312425   },   "amount": {     "instructedAmount": {       "value": 10.08,       "currency": "CZK"     }   }, }</pre> </div>
x-request-id	<input type="text"/>	External Request ID	header	string
TPP-Name	<input type="text" value="(required)"/>	Transaction Initiator name	header	string

### “New Standing Order initiation” 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

### “Standing Order status” PIS mock calling for the purpose of testing

The user chooses an operation he/she wishes to test. In this case, it is “Standing Order status”. This operation allows you to view standing order status information. The operation menu drops down after the user clicks on the "SHOW MORE" button.

# PISP-Sandbox

Version: v1

Updated: 05/May/2020 09:11:54 AM CEST

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 /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 ▾
<b>POST</b>	New Standing Order initiation /standingorders	SHOW MORE ▾
<b>POST</b>	Standing Orders Authorization - start /standingorders/{transactionIdentification}/sign/{signId}	SHOW MORE ▾
<b>GET</b>	Standing Orders Status /standingorders/{transactionIdentification}/status	SHOW MORE ▾
<b>GET</b>	Standing Orders Detail /standingorders/{transactionIdentification}	SHOW MORE ▾

### “Filling in the required fields of the Standing Order status”

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 Class (Status 200)


OK

Model | Example Value

```

{
  "unstructured": "avPoLe"
},
"paymentTypeInformation": {
  "serviceLevel": {
    "code": "DMCT"
  }
},
"signInfo": {
  "state": "OPEN",
  "signId": "string",
  "signInfo": "ACTC"
}

```

Response Content Type  

#### Parameters

Parameter	Value	Description	Parameter Type	Data Type
transactionIdentification	<input type="text" value="(required)"/>	Unique bank transaction identification example: '358745'	path	string
TPP-Name	<input type="text" value="(required)"/>		header	string
x-request-id	<input type="text"/>		header	string

### Response Class (Status 200)


successful operation

Model | Example Value

```

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

```

Response Content Type  

#### Parameters

Parameter	Value	Description	Parameter Type	Data Type
transactionIdentification	<input type="text" value="(required)"/>	Unique bank transaction identification example: '358745'	path	string
x-request-id	<input type="text"/>	External Request ID	header	string
TPP-Name	<input type="text" value="(required)"/>	Transaction Initiator name	header	string

### “Standing Order Status” 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 paymentsId not found"     }   ] }</pre>	
415	Invalid message charset	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR10",       "message": "InvalidCharacterSet"     }   ] }</pre>	
500	Unexpected error occurred	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>	

**TRY IT OUT** Hide Response

### “Standing Order Detail” PIS mock calling for the purpose of testing

The user chooses an operation he/she wishes to test. In this case, it is “Standing Order Details”. This operation allows you to view the details of a standing order. The operation menu drops down after the user clicks on the “SHOW MORE” button.

## PISP-Sandbox

Version: v1 | Updated: 05/May/2020 09:11:54 AM CEST | Status: PUBLISHED

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try PREMA\_TEST\_KB ▼ Using SANDBOX ▼ Key

Set Request Header

Authorization : Bearer eyJ0eXAI0iJKV1QilCJhbGciOiJSUzI1NiU9.

Swagger ( /swagger.json )

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

POST	New payment – payment initiation /payments	SHOW MORE <span style="color: red;">▼</span>
POST	Payment Authorization – starting the particular authorization method /payments/{paymentId}/sign/{signId}	SHOW MORE <span style="color: red;">▼</span>
GET	Payment status information /payments/{paymentId}/status	SHOW MORE <span style="color: red;">▼</span>
POST	Balance check /payments/balanceCheck	SHOW MORE <span style="color: red;">▼</span>
POST	New Standing Order initiation /standingorders	SHOW MORE <span style="color: red;">▼</span>
POST	Standing Orders Authorization - start /standingorders/{transactionIdentification}/sign/{signId}	SHOW MORE <span style="color: red;">▼</span>
GET	Standing Orders Status /standingorders/{transactionIdentification}/status	SHOW MORE <span style="color: red;">▼</span>
GET	Standing Orders Detail /standingorders/{transactionIdentification}	SHOW MORE <span style="color: red;">▼</span>

### Filling in the required fields of the “Standing Order Detail” 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.

**Response Class (Status 200)**


OK

Model | Example Value

```

{
  "unstructured": "avPoLe"
},
{
  "paymentTypeInformation": {
    "serviceLevel": {
      "code": "DMCT"
    }
  },
  "signInfo": {
    "state": "OPEN",
    "signId": "string",
    "signInfo": "ACTC"
  }
}

```

Response Content Type  

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
transactionIdentification	<input type="text" value="(required)"/>	Unique bank transaction identification • example: '358745'	path	string
TPP-Name	<input type="text" value="(required)"/>		header	string
x-request-id	<input type="text"/>		header	string

**“Standing Order status” 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** Hide Response

### “New batch payment – instruction initiation” PIS mock calling for the purpose of testing

The user chooses an operation he/she wishes to test. In this case, it is “New batch payment – instruction initiation”. This operation allows you to view the New batch payment – instruction initiation. The operation menu drops down after the user clicks on the "SHOW MORE" button.

<b>POST</b>	Standing Orders Authorization - start /standingorders/{transactionIdentification}/sign/{signId}	SHOW MORE ▾
<b>GET</b>	Standing Orders Status /standingorders/{transactionIdentification}/status	SHOW MORE ▾
<b>GET</b>	Standing Orders Detail /standingorders/{transactionIdentification}	SHOW MORE ▾
<b>POST</b>	New batch payment - instruction initiation /batchpayments	SHOW MORE ▾
<b>POST</b>	Batch Authorization – starting the particular authorization method /batchpayments/{transactionIdentification}/sign/{signId}	SHOW MORE ▾
<b>GET</b>	Batch payment status /batchpayments/{transactionIdentification}/status	SHOW MORE ▾

### Filling in the required fields of the “New batch payment – instruction initiation”

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.

```
{
  "transactionIdentification": "001RE6YMM",
  "signInfo": {
    "state": "OPEN",
    "signId": "048885570000001020045"
  },
  "instructionStatus": "ACTC",
  "batchDigest": "AAF21A859D225FD1D1889B2F08DA3E4A9AD6F7CC"
}
```

Response Content Type  


**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
x-request-id	<input type="text"/>	External Request ID	header	string

paymentRequest

New batch payment request

body

Parameter content type:  

Model	Example Value
	<pre>{   "exchangeIdentification": "103149078",   "instructionName": "Mzdy březen 2021 z KB účtu.",   "payments": [     "ZU00000KJSE",     "ZU00000KJSE"   ] }</pre>



**“New batch payment – instruction initiation” 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>	

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

The user chooses an operation he/she wishes to test. In this case, it is “Batch authorization”. This operation allows you to view the Batch Authorization. The operation menu drops down after the user clicks on the "SHOW MORE" button.

<b>POST</b>	Standing Orders Authorization - start /standingorders/{transactionIdentification}/sign/{signId}	SHOW MORE ▾
<b>GET</b>	Standing Orders Status /standingorders/{transactionIdentification}/status	SHOW MORE ▾
<b>GET</b>	Standing Orders Detail /standingorders/{transactionIdentification}	SHOW MORE ▾
<b>POST</b>	New batch payment - instruction initiation /batchpayments	SHOW MORE ▾
<b>POST</b>	Batch Authorization – starting the particular authorization method /batchpayments/{transactionIdentification}/sign/{signId}	SHOW MORE ▾
<b>GET</b>	Batch payment status /batchpayments/{transactionIdentification}/status	SHOW MORE ▾

### Filling in the required fields of the “Batch Authorization”

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.

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
x-request-id	<input type="text"/>	External Request ID	header	string
transactionIdentification	(required) <input type="text"/>	Unique bank transaction identification example: '358745'	path	string
signId	(required) <input type="text"/>	Identifier of the change	path	string
authInitiationRequest	(required) <input type="text"/>  Parameter content type: application/json <input type="button" value="v"/>	Initiation request - identifier of the batch authorization method and back URL	body	string

Model	Example Value
	<pre>{   "authorizationType": "USERAGENT_REDIRECT",   "redirectUrl": "http://kb.cz",   "batchDigest": "AAF21A8590225FD1D1889B2F08DA3E4A9AD6F7CC" }</pre>

**“Batch 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.

HTTP Status Code	Reason	Response Model	Header				
400	Input parameter is invalid	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "ERR_CODE_400",       "scope": "x-request-id",       "message": "Value of parameter x-request-id is wrong"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "ERR_CODE_400",       "scope": "x-request-id",       "message": "Value of parameter x-request-id is wrong"     }   ] }</pre>	
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	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>						
403	Invalid certificate or token	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>						
404	Id does not exist	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "ID_NOT_FOUND",       "message": "Parameter paymentId not found"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "ID_NOT_FOUND",       "message": "Parameter paymentId not found"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "ID_NOT_FOUND",       "message": "Parameter paymentId not found"     }   ] }</pre>						
415	Invalid message charset	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre>						
500	Unexpected error occurred	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>						

### “Batch payment status” PIS mock calling for the purpose of testing

The user chooses an operation he/she wishes to test. In this case, it is “Batch payment status”. This operation allows you to view the Batch payment status. The operation menu drops down after the user clicks on the "SHOW MORE" button.

<b>POST</b>	Standing Orders Authorization - start /standingorders/{transactionIdentification}/sign/{signId}	SHOW MORE ▾
<b>GET</b>	Standing Orders Status /standingorders/{transactionIdentification}/status	SHOW MORE ▾
<b>GET</b>	Standing Orders Detail /standingorders/{transactionIdentification}	SHOW MORE ▾
<b>POST</b>	New batch payment - instruction initiation /batchpayments	SHOW MORE ▾
<b>POST</b>	Batch Authorization – starting the particular authorization method /batchpayments/{transactionIdentification}/sign/{signId}	SHOW MORE ▾
<b>GET</b>	Batch payment status /batchpayments/{transactionIdentification}/status	SHOW MORE ▾

### Filling in the required fields of the “Batch payment status”

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.

Response Content Type  ▾

#### Parameters

Parameter	Value	Description	Parameter Type	Data Type
transactionIdentification	<input type="text" value="(required)"/>	Unique bank transaction identification example: '358745'	path	string
x-request-id	<input type="text"/>	External Request ID	header	string

### “Batch payment status” 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	<b>Model</b>   <b>Example Value</b> <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	<b>Model</b>   <b>Example Value</b> <pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>	
403	Invalid certificate or token	<b>Model</b>   <b>Example Value</b> <pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>	
500	Unexpected error occurred	<b>Model</b>   <b>Example Value</b> <pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>	

## 6. Access to the application through direct calling

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

Resource for establishing a new payment.

#### Resource characteristics

**URI:** /payments  
**HTTP Method:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/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 <b>application/json</b> format is primarily supported.
<b>API-key</b>	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as the configuration element of communication.
<b>Authorization</b>	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.

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
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of a method that does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request .
400	FIELD_INVALID	The field value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.

400	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
+	paymentTypeInfoInformation	[0..1]	ALL	PaymentTypeInfoInformation19	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	AccountIdentification4ChoiceCZ	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	GenericAccountIdentification1CZ:	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),	Unstructured report for the creditor (see below*)



				including supported special characters	
++	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

## Established/Initiated Payment Status (GET /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/v1/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 <b>application/json</b> format is primarily supported.
<b>API-key</b>	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as the configuration element of communication.

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
<b>404</b>	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

## Step II – 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:** /payments/{paymentId}/sign/{signId}  
**HTTP Method:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/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	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.
<b>API-key</b>	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as the configuration element of communication.
<b>Authorization</b>	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.

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

## Step II – Payment Authorisation Initiation – Bank-Specific

### Request parameters:

LEVEL	MESSAGE ELEMENT	OCCUR- RENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	PISP ALL	Text	The code of the required authorisation (from authorisation scenarios)

### Response parameters:

LEVEL	MESSAGE ELEMENT	OCCUR- RENCE	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

<b>URI:</b>	<b>/my/payments/balanceCheck</b>
<b>HTTP Method:</b>	POST
<b>Request URL:</b>	<a href="https://api.kb.cz/sandbox/pisp/v1/payments/balanceCheck">https://api.kb.cz/sandbox/pisp/v1/payments/balanceCheck</a>
<b>Authorization:</b>	request <b>requires</b> the authorization of user/client as part of API calling
<b>Use certificate:</b>	request <b>requires</b> the use of the qualified third-party certificate
<b>Paging:</b>	no
<b>Sorting:</b>	no
<b>Filtering:</b>	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>API-key</b>	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.
<b>Authorization</b>	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type
<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
<b>401</b>	UNAUTHORISED	Missing certificate.
<b>403</b>	FORBIDDEN	Calling of the method which 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	FIELD value is not valid.
<b>400</b>	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
<b>400</b>	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
<b>400</b>	AC12	[InvalidAccountType] - account type does not match allowed account types (e.g., a non-paying account).
<b>403</b>	AG01	[TransactionForbidden] – absent consent to access to balance check at the account.
<b>400</b>	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
<b>400</b>	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:

CODE	DESCRIPTION
APPR	Enough funds on this account



DECL	Uninsufficient funds on this account
------	--------------------------------------

### Standing order

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

Resource for entering a new standingorder.

#### Resource characteristics

**URI:** /standingorders  
**HTTP Method:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/standingorders/>  
**Authorisation:** 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:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>API-key</b>	Text	Ne	Volitelný řetězec vydaný komunikující třetí straně jako identifikátor volání této strany primárně sloužící jako konfigurační prvek komunikace.
<b>Authorization</b>	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.
<b>TPP-Name</b>	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“ V tomto poli jsou podporovány pouze znaky bez diakritiky.
<b>TPP-Identification</b>	Text	Ne	Identifikace původního TPP, které request vytvořilo. Např.: „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 **Chyba! Nenalezen zdroj odkazů**. New Payment – Payment Initiation REPORT ELEMENTS.

Error codes defined for the payment initiation POST service

HTTP STATUS CODE	ERROR CODE	ÚČEL
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
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	AC03	[InvalidCreditorAccountNumber] - recipient account number is closed, blocked, credit card are disabled for the account type, or the recipient's account number is in invalid format (note: only valid for in-house payments). Není používáno.
400	AC09	[InvalidAccountCurrency] – the specified payer account currency does not correspond to the client's account currency for the account number held in the bank (the account currency is optional, in the case of multicurrency accounts, the account currency must be specified - Raiffeisenbank). Used instead of AC10.
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. Není využíváno
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 format or other technical problem with the query processing.
400	BE19	[InvalidChargeBearerCode] - invalid fee type for the given transaction type. Not supported
400	DT01	[InvalidDate] - non-existent date value or format. Maturity in the future or in the past. Transactions on a non-business day of a bank (unless the bank receives such orders).
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RC07	[InvalidCreditorBICIdentifier] – invalid recipient SWIFT / BIC code. Not supported
400	RC10	[InvalidCreditorClearingSystemMemberIdentifier] - invalid recipient bank code identification. Not supported
400	RR03	[MissingCreditorNameOrAddress] - the field is missing the required data about the recipient's name or address (or its part). If the data is in bad format, the error code FIELD_INVALID is used. Not supported
400	RR10	[InvalidCharacterSet] – invalid character set in the request.
400	REC_SEND	[RECEIVER_SAME_AS_SENDER] – Payer's account is the same as the payee's account (Applies to all types of payments)

### New Payment – standing orders Initiation report elements

#### Considered standing orders types

Standing orders may only be placed as domestic payments in the Czech Republic.

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	standingOrderIdentification	[1..1]	ALL	Max35Text	Standing order identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification
++	endToEndIdentification	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	End to end identification. SEPA – Not supported
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification

+	paymentTypeInformation	[0..1]	ALL	The same object as to initiate payments chapter ...	
+	amount	[1..1]	ALL	STDO - AmountType3CZ ESST - AmountType3CZ XBST AmountType3Choice	Payment amount
++	instructedAmount	[1..1]	ALL	CurrencyAndAmount	Amount and currency in the instruction
+++	value	[0..0]	ALL	Amount	Amount of the transfer. Not supported
+++	currency	[0..0]	ALL	CurrencyCode	Transfer Currency. Not supported.
+	requestedExecutionDate	[1..1]	ALL	ISODate	Requested date of payment
+	standingOrder	[1..1]	ALL	Structure	Structure describing the parameters of the standing orders.
++	alias	[0..1]	ALL	Max250Text	Description resp. the user-defined standing order payment name. This field is not supported on input. For alias in KB is used the value Remittanceinformation from which the information is taken into the alias field
++	execution	[1..1]	ALL		Features of execution a standing order for payment.
+++	mode	[1..1]	ALL	Max35Text	The execution mode defines when or how standing order will be cancelled, processed the last time. Possible values: UNTIL_DATE (standing order is valid until specific date - field lastExecutionDate), UNTIL_CANCELLATION (standing order is valid forever and must be cancelled by client), AFTER_MAX_ITERATION_EXCEEDED (certain count of executions is specified - field maxIterations) In KB the date of the last payment is determined by the number of repetitions and added or MAX_AMOUNT_EXCEEDED (maximum amount which can be transferred for this order is specified, if next iteration would exceed this amount it is not executed - field maxAmount). Not supported
+++	modeDue	[1..1]	ALL	Max35Text	The execution due mode defines how the date when order should be

					<p>executed is specified. Possible values: DUE_DAY_OF_MONTH (specific number of day in the month is defined) or DUE_OR_BEFORE_DAY_OF_MONTH (is defined as a specific day of the month or the previous day if the day falls on a non-banking day) or DUE_OR_NEXT_DAY_OF_MONTH (is defined as a specific day of the month or the next day if the day falls on a non-banking day) or DUE_LAST_DAY_OF_MONTH (order is executed on last day of particular month). Not supported</p>
+++	interval	[1..1]	ALL	Max10Text	<p>Execution interval defines how often order is executed. Possible values: DAILY, WEEKLY, BI_WEEKLY, MONTHLY, BI_MONTHLY, QUARTERLY, HALFYEARLY, YEARLY, <del>IRREGULAR</del> Not supported.</p>
+++	intervalDue	[0..0]	ALL	Number	<p>Value represents order number of the day within particular period when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12 for YEARLY. Not supported</p>
++	validity	[0..1]	ALL		<p>Elements restricting the validity of the standing order.</p>
+++	lastExecutionDate	[0..1]	ALL	ISODate	<p>Date when the last order will be processed. Null value responds to ending by the user.</p>
+++	maxIterations	[0..1]	ALL	Number	<p>Maximum number of iterations - processing of the standing order. Only applicable in combination with executionMode AFTER_MAX_ITERATION_EXCEEDED.</p>
+++	maxAmount	[0..0]	ALL	Amount	<p>Maximum amount to be transferred using the standing order. Only applicable in combination with executionMode</p>

					AFTER_MAX_AMOUNT_EXCEEDED. Not supported
++++	value	[0..0]	ALL	Amount	Maximum amount of the transfer. Not supported
++++	currency	[0..0]	ALL	CurrencyCode	Transfer Currency of the maximum amount. Not supported
++	exceptions	[0..0]	ALL		Elements defining exceptions to executing a standing order. Not supported.
+++	stoppages	[0..0]	ALL	Array of Max20Text	List of months where there is no payment. Possible values: JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER. Not supported
+++	breaks	[0..0]	ALL	Array of Intervals	Break periods. Not supported.
++++	validFromDate	[0..0]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date. Not supported.
++++	validToDate	[0..0]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date. Not supported.
+	exchangeRateInformation	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	chargeBearer	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	ChargeBearerType1Code	Fee payer. EHP; NONEHP – Not supported
+	chargesAccount	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	ultimateDebtor	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	SEPA – Not supported
+	debtor	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	debtorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	intermediaryAgent1	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	creditorAgent	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	SEPA; EHP; NONEHP – Not supported
+	creditor	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	SEPA; EHP; NONEHP – Not supported

+	creditorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	ultimateCreditor	[0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	SEPA – Not supported
+	Purpose	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instructions for the next bank
+	remittanceInformation	[0..1]	ALL	The same object as to initiate payments chapter ...	

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 – Standing orders Initiation response elements**

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

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	standingOrderIdentification	[1..1]	ALL	Max35Text
++	transactionIdentification	[1..1]	Max35Text	Identifier of established standing order instruction.
+	paymentTypeInfoInformation	[1..1]	ALL	
++	serviceLevel	[1..1]	±	Service placement (within type of payment).
+++	code	[1..1]	Text	Type of entered payment. Is only supported DMCT - DomesticPayment
+	signInfo	[1..1]	±	Status information and id of unauthorized transactions
++	state	[1..1]	StateCode	Transaction authorization status In KB is always returned status OPEN
++	signId	[1..1]	Text	Identifier of the authorization process of a particular transaction.
+	instructionStatus	[1..1]	StatusCode	Status of entered standing order

				In KB is always returned states ACWC or ACTC
+	statusChangeInfo	[0..1]	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC".

#### Values of serviceLevel.code - The type of entered payment

KÓD	POPIS
DMCT	[DoMestic Credit Transfer] Domestic payment
ESCT	[SEPA Credit Transfer] – SEPA payment. Not supported.
XBCT	[Cross-Border Credit Transfer] – Cross border payment. Not supported
EXCT	[EEA X-border Credit Transfers] – Cross border payment within the EEP. Not supported
NXCT	[Non-EEA X-border Credit Transfers] – Cross border payment outside the EEP. Not supported.

#### Status codes of standing order – 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 sent

#### Standing order authorization – bank-specific (POST /my/standingorders/{transactionIdentification}/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.

**The standing orders authorisation request should be sent within 5 minutes from the moment the payment initiation has been sent by a third party.**

#### Resource characteristics

**URI:** /standingorders/{transactionIdentification}/sign/{signId}  
**HTTP Method:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/standingorders/{transactionIdentification}/sign/{signId}>  
**Authorisation:** 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:

<i>PARAMETER</i>	<i>TYPE</i>	<i>MANDA-TORY</i>	<i>PURPOSE</i>
<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.
<b>API-key</b>	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as the configuration element of communication.
<b>Authorisation</b>	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
<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:

<i>PARAMETER</i>	<i>TYPE</i>	<i>MANDA-TORY</i>	<i>PURPOSE</i>
<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 0 Payment Authorisation initiation – Bank-Specific, Step II, REPORT ELEMENTS



Error codes defined for the payment authorisation initiation POST service:

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

### Payment Authorisation initiation report elements

Parameters of the request:

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	authorizationType	[1..1]	±	Code of required <i>authorization</i> (from <i>authorization scenarios</i> )
+	Href	[1..1]	±	Reference to call federated authorization
++	url	[1..1]	Text	URL link or package of federated authorization. Always returned
++	Id	[1..1]	Text	Possible id for calling federated authorization. Always returned
+	method	[1..1]	Text	Method to use href link and federated authorization. Always returned.
+	formData	[0..1]	±	Optional element. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains data for sending in redirecting to federated authorization.
++	SAMLRequest	[0..1]	Text	Optional parameter. For the method POST federated authorization. (authorizationType=USERAGENT_REDIRECT) element Contains data of SAML request. KB only uses GET.
++	relayState	[0..1]	Text	Optional parameter. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains relayState for returnable value. KB only uses GET.
+	signInfo	[1..1]	±	Information on instruction author
++	state	[1..1]	Text	Status of transaction authorization in a format supported by the bank
++	signId	[1..1]	Text	Unique identifier for current transaction authorization

### Standing order detail (GET /my/standingorders/{transactionIdentification})

Resource for get detail about the standing order.

It is at the choice of each bank whether to return details of any client's standing order or just those that the client has established through a particular TPP.

#### Resource characteristics

**URI:** /standingorders/{transactionIdentification}  
**HTTP Metoda:** GET  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/standingorders/{transactionIdentification}>  
**Authorisation:** 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.

**Paging:** ne

Sorting: ne  
 Filtration: ne

Query parameters of the request:

PARAMET	TYP	POVINNÝ	ÚČEL
transactionIdentification	Max35Text	Yes	Identifier of established standing order instruction

Request header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the <b>application/json</b> 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 the configuration element of communication.
Authorisation	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
TPP-Name	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.
TPP-Identification	Text	No	The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'

Response header parameters:

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

Error codes defined for the service GET info on entered/initiated standing order:

HTTP STATUS CODE	KÓD CHYBY	ÚČEL
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

Elementy response zprávy Detail trvalého příkazu

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	standingOrderIdentification	[1..1]	ALL	Max35Text	Standing order identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification
++	endToEndIdentification	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	End to end identification SEPA – Not supported
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInformation	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	amount	[1..1]	ALL	STDO - AmountType3CZ ESST - AmountType3CZ XBST AmountType3Choice	Payment amount
++	instructedAmount	[1..1]	ALL	CurrencyAndAmount	Amount and currency in the instruction
+++	value	[1..1]	ALL	Amount	Amount of the transfer
+++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency
+	requestedExecutionDate	[1..1]	ALL	ISODate	Requested date of payment
+	standingOrder	[1..1]	ALL	Structure	Structure describing the parameters of the standing orders.
++	alias	[0..1]	ALL	Max250Text	Description resp. the user-defined standing order payment name. In the alias field always display information from the remittance information field.
++	execution	[1..1]	ALL		Features of execution a standing order for payment.
+++	mode	[1..1]	ALL	Max35Text	The execution mode defines when or how standing order will be cancelled, processed the last time. Possible values: UNTIL_DATE (standing order is valid until specific date - field lastExecutionDate), UNTIL_CANCELLATION (standing order is valid forever and must be cancelled by client), AFTER_MAX_ITERATION_EXCEEDED (certain count of executions is specified - field maxIterations) In KB the date of the last payment is determined by the number of repetitions and added or MAX_AMOUNT_EXCEEDED (maximum amount which can be transferred for this order is specified, if next iteration would exceed this amount it is

					not executed - field maxAmount). Not supported
+++	modeDue	[1..1]	ALL	Max35Text	The execution due mode defines how the date when order should be executed is specified. Possible values: DUE_DAY_OF_MONTH (specific number of day in the month is defined) or DUE_OR_BEFORE_DAY_OF_MONTH (is defined as a specific day of the month or the previous day if the day falls on a non-banking day) or DUE_OR_NEXT_DAY_OF_MONTH (is defined as a specific day of the month or the next day if the day falls on a non-banking day) or DUE_LAST_DAY_OF_MONTH (order is executed on last day of particular month). Not supported
+++	interval	[1..1]	ALL	Max10Text	Execution interval defines how often order is executed. Possible values: DAILY, WEEKLY, BI-WEEKLY, MONTHLY, BI_MONTHLY, QUARTERLY, HALFYEARLY, YEARLY, IRREGULAR — Not supported.
+++	intervalDue	[0..0]	ALL	Number	Value represents order number of the day within particular period when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12 for YEARLY. Not supported.
++	validity	[1..1]	ALL		Elements restricting the validity of the standing order. If the final due date is not filled in, an empty field is always returned.
+++	lastExecutionDate	[0..1]	ALL	ISODate	Date when the last order will be processed. Null value responds to ending by the user.
+++	maxIterations	[0..0]	ALL	Number	Maximum number of iterations - processing of the standing order. Only applicable in combination with executionMode

					AFTER_MAX_ITERATION_EXCEEDED. Not supported
+++	maxAmount	[0..0]	ALL	Amount	Maximum amount to be transferred using the standing order. Only applicable in combination with executionMode AFTER_MAX_AMOUNT_EXCEEDED. Not supported.
++++	value	[1..1]	ALL	Amount	Maximum amount of the transfer
++++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency of the maximum amount
++	exceptions	[0..1]	ALL		Elements defining exceptions to executing a standing order.
+++	stoppages	[0..0]	ALL	Array of Max20Text	List of months where there is no payment (only applicable with interval IRREGULAR). Possible values: JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER. Not supported
+++	breaks	[0..1]	ALL	Array of Intervals	Break periods.
++++	validFromDate	[0..1]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date.
++++	validToDate	[0..1]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date.
+	exchangeRateInformation	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	chargeBearer	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	ChargeBearerType1Code	Fee payer.  EHP - Not supported
+	chargesAccount	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	ultimateDebtor	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	SEPA - Not supported
+	debtor	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	debtorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	intermediaryAgent1	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	creditorAgent	[0..0] [0..0]	TUZEM SEPA	The same object as to initiate payments chapter ...	SEPA, EHP, NONEHP - Not supported

		[0..0] [0..0]	EHP NONEHP		
+	creditor	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	SEPA, EHP, NONEHP - Not supported
+	creditorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	ultimateCreditor	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	SEPA - Not supported
+	purpose	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instructions for the next bank
+	remittanceInformation	[1..1]	ALL	The same object as to initiate payments chapter ...	It is always returned even if nothing has been entered.
+	signInfo	[1..1]	ALL	±	Status information and id of unauthorized transactions
++	state	[1..1]	ALL	StateCode	Transaction authorization status
++	signId	[1..1]	ALL	Text	Identifier of the authorization process of a particular transaction. Always returned
+	instructionStatus	[1..1]	ALL	Status Code set	Transaction status identifier
+	statusChangeInfo	[0..1]	ALL	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC". Not supported.

### Standing order status (GET /my/standingorders/{transactionIdentification}/status)

Resource for getting status of the standing order.

#### Resource characteristics

**URI:** /standingorders/{transactionIdentification}/stauts  
**HTTP Metoda:** GET  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/standingorders/{transactionIdentification}/status>  
**Authorisation:** the request **requires** no 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.

**Stránkování:** ne  
**Třídění:** ne  
**Filtrování:** ne

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 <b>application/json</b> format is primarily supported.

<b>API-key</b>	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as the configuration element of communication.
<b>Authorisation</b>	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
<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:

<i>PARAMETER</i>	<i>TYPE</i>	<i>MANDATORY</i>	<i>PURPOSE</i>
<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.

Error codes defined for the service GET info on entered/initiated standing order:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
<b>401</b>	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
<b>404</b>	TRANSACTION_MISSING	The requested transaction ID does not exist.

#### 12.2. MESSAGE ELEMENTS Status of entered/initiated standing order

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURRENCE</i>	<i>FORMAT TYP</i>	<i>PRESENTATION</i>
+	instructionStatus	[1..1]	StatusCode	Status of entered standing order
+	realizedPayments	[0..0] conditional if it is an authorized standing order	Array	Array of transaction identifiers. Not supported
++	transactionIdentification	[1..1]	Max35Text	Identifier of established transaction
++	date	[0..1]	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.
+	errorinfo	[0..1]	Array	
++	error	[0..1]	String	
++	parameters	[0..1]	Array	
++	message	[0..1]	String	

#### Status codes of standing order – StatusCode

<i>HTTP STATUS CODE</i>	<i>STATUSCODE</i>	<i>PURPOSE</i>
<b>200</b>	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	[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] – Standing order on the debtor's account has been completed

### Batch Payments – initiation (POST /my/batchpayments)

Resource for establishing a batch payment.

#### Additional information:

- A batch payment can separately include transactions of the same type of domestic and SEPA payments (for example: SEPA only, domestic only).
- Instant and foreign payments are not supported.

#### Resource characteristics

**URI:** /batchpayments  
**HTTP Method:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/batchpayments>  
**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>Authorization</b>	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
<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>X-Request-ID</b>	Text	Yes	Unique identifier of the TPP application request.

### Batch payment – initiation request elements

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	exchangeIdentification	[1..1]	Max14Text	Batch identification. Clear query identification.
+	instructionName	[0..1]	Max35Text	An entry field for an optional batch name
+	payments	[1..1]	Max35Text	Collection of payments

### Batch payment – initiation response elements



LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	transactionIdentification	[1..1]	Max35Text	Batch transaction identification.
+	signInfo	[1..1]	±	Information about the status and id of the unauthorised transaction.
++	state	[1..1]	StateCode	Information about the status of the batch transaction authorisation.
++	signId	[1..1]	Text	Identifier of the authorising process of the particular batch transaction.
+	instructionStatus	[1..1]	StatusCode	Batch transaction status identifier.
+	statusChangeInfo	[0..1]	Text	Specifies the change that was made to the instruction. Used with instructionStatus "ACWC".
+	batchDigest	[1..1]	Max50Text	A unique ID, wrapped in a SHA HASH batch, to help check that there has been no change between batch initiation and authorization.

#### 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 sent

#### Error codes defined for the batch payment initiation POST service

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Missing certificate.

403	FORBIDDEN	Calling of the method that does not correspond to the licence, or invalid certificate.
404	[ID_NOT_FOUND]	If the field is filled in but is invalid.
404	NOT_FOUND	If the parameter is not filled in or the field does not correspond to the maximum length.
400	TYPE_DIFFERENT	Payments can be one type of Domestic or SEPA.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC12	[InvalidAccountType] The account type does not match allowed account types (e.g., a non-paying account).
400	NARR	[Exceeded number of orders in a single batch]
400	NARR	[Batch payment is empty]
400	NARR	[ID is not unique]
400	NARR	[AccessDenied - Unauthorized access to account.]
400	NARR	[AccessDenied - User is not in active state.]
400	NARR	[AccessDenied - Client is not in active state.]
400	NARR	[AccessDenied - Debtor payment account is not in active state.]
400	NARR	[AccessDenied - Debtor payment account is not available in internet banking]
400	NARR	[Foreign payment - unsupported payment type.]
400	NARR	[Transaction has already been authorized or rejected.]
400	NARR	[Payment has already been incorporated into another batch payment.]
400	NARR	[Invalid Originator TPP]
400	NARR	[Payments initiated by different applications]
400	NARR	[Daily limit of transactions in batches exhausted.]

### Batch payment authorization – initiation (POST /my/ batchpayments/{transactionIdentification}/sign/{signId})

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

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

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

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

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

**Important:** The payment authorization request should be sent within 5 minutes from the moment the payment initiation has been sent by a third party.

#### Resource characteristics

**URI:** </batchpayments/{transactionIdentification}/sign/{signId}>  
**HTTP Method:** POST  
**RequestURL:** <https://api.kb.cz/sandbox/pisp/v1/batchpayments/{transactionIdentification}/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	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>Authorization</b>	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
<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>X-Request-ID</b>	Text	Yes	Unique identifier of the TPP application request.

Path parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
transactionIdentification	Text	Yes	Batch transaction identification.
signId	Text	Yes	Identifier of the particular batch transaction.

#### Batch payment authorization - initiation request elements

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	Text	The code of the required authorisation (from authorisation scenarios) Allow value : "USERAGENT_REDIRECT",
+	redirectUrl	[0..1]	Text	URL link or package of federated authorization call back address.  This address is used by the federated bank authorization to redirect back to the TPP application after authorization is complete.
+	batchDigest	[0..1]	Max50Text	A unique ID, wrapped in a SHA HASH batch, to help check that there has been no change between batch initiation and authorization. We recommend its use from the point of view of security.

#### Batch payment authorization – initiation response elements

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	±	The code of the required authorization (from authorization scenarios)
+	href	[0..1]	±	The reference for calling the federated authorization
++	url	[1..1]	Text	The URL link or package federated authorization
++	id	[0..1]	Text	The potential id for calling the federated authorization
+	method	[0..1]	Text	The method of the use of the href link for the federated authorization.
+	formData	[0..1]	±	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]	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]	Text	An optional parameter. In the case of the POST method of the federated authorisation (authorisationType=USERAGENT_REDIRECT), the element contains the relayState for the return value.
+	signInfo	[1..1]	±	Information about the instruction authorization.
++	state	[1..1]	Text	A status of the transaction authorization in a format supported by the bank.
++	signId	[1..1]	Text	A status of the transaction authorization in a format supported by the bank.

#### Error codes defined for batch payment authorization POST service

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method that does not correspond to the licence, or invalid certificate.
404	[ID_NOT_FOUND]	If the field is filled in but is invalid.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC12	[InvalidAccountType] The account type does not match allowed account types (e.g., a non-paying account).
400	NARR	[AccessDenied - Unauthorized access to account.]
400	NARR	[AccessDenied - User is not in active state.]
400	NARR	[AccessDenied - Client is not in active state.]
400	NARR	[AccessDenied - Debtor payment account is not in active state.]
400	NARR	[AccessDenied - Debtor payment account is not available in internet banking]

400	NARR	[One of the Transactions has already been authorized.]
400	NARR	[Incorrect batch status.]
400	NARR	[Invalid Originator TPP]
400	NARR	[Batch payment has already been rejected]

### Batch payment – status (GET/my/batchpayments/{transactionIdentification}/status)

Resource for getting status of the batch payment.

#### Resource characteristics

<b>URI:</b>	<a href="/batchpayments/{transactionIdentification}/status">/batchpayments/{transactionIdentification}/status</a>
<b>HTTP Method:</b>	GET
<b>Request URL:</b>	<a href="https://api.kb.cz/sandbox/pisp/v1/batchpayments/{transactionIdentification}/status">https://api.kb.cz/sandbox/pisp/v1/batchpayments/{transactionIdentification}/status</a>
<b>Authorization:</b>	request requires <b>no authorization</b> of user/client as part of API calling
<b>Certification:</b>	the request <b>requires</b> 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.
<b>Paging:</b>	no
<b>Sorting:</b>	no
<b>Filtration:</b>	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 <b>application/json</b> format is primarily supported.
<b>Authorization</b>	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
<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>X-Request-ID</b>	Text	Yes	Unique identifier of the TPP application request.

Path parameter:

PARAMETER	TYPE	MANDATORY	PURPOSE
transactionIdentification	Text	Yes	Batch transaction identification.

#### Batch payment status response elements

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	transactionIdentification	[1..1]	Max35Text	Batch transaction identification.
+	signInfo	[1..1]	±	Information about the status and id of the unauthorised transaction.
++	state	[1..1]	StateCode	Information about the status of the batch transaction authorisation.

++	signId	[1..1]	Text	Identifier of the authorising process of the particular batch transaction.
+	instructionStatus	[1..1]	StatusCode	Batch transaction status identifier.
+	batchDigest	[1..1]	Max50Text	A unique ID, wrapped in a SHA HASH batch, to help check that there has been no change between batch initiation and authorization.
+	payments	[0..n]	±	Collection of payments
++	transactionIdentification	[1..1]	Max35Text	Identifier of the created transaction.
++	instructionIdentification	[0..1]	Max35Text	Instruction identification. If it is not filled in, NOTPROVIDED will be added.
++	instructionStatus	[1..1]	StatusCode	Status of established payment.
++	errorInfo	[0..1]	±	Used when instructionStatus is at the "RJCT" transaction level
+++	code	[0..1]	Max15Text	Error code
+++	description	[0..1]	Max150Text	Error message description detail.

**Error codes define for batch payment status GET service**

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method that does not correspond to the licence, or invalid certificate.
404	[ID_NOT_FOUND]	If the field is filled in but is invalid.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	NARR	[Invalid Originator TPP]

**7. PSD2 glossary – selected terms**

**API** – Application Programming Interface

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

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

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

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

**AIS** – Account Information Service

**AISP** – Account Information Service Provider

**PIS** – Payment Initiation Service

**PISP** – Payment Initiation Service Provider

**CIS** – Card-based Payment Instrument Issuance Service

**CISP** – Card-based Payment Instrument Issuance Service Provider

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