

The Sandbox SK Manual on the Payment Initiation Service

Change log

Date	Version	Description
29.01.2020	1	First document version
19.5.2020	2	Document edit
18.11.2020	3	Document edit (added certain specifics of payment system in Slovakia, etc.)

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. Procedure for generating PISP key/token	5
6. PISP Mock Calling Methods.....	8
7. Access to the application through the API console	9
“New Payment” PIS mock calling for the purpose of testing.....	9
“Payment status information” PIS mock calling for the purpose of testing	12
“Payment Authorization” PIS mock calling for the purpose of testing.....	15
Balance check” PIS mock calling for the purpose of testing	18
8. Access to the application through direct calling.....	20
New Payment – Payment Initiation (POST /my/payments)	20
New Payment – Payment Initiation report elements.....	21
New Payment – Payment Initiation response elements	32
Established/Initiated Payment Status (GET /payments/{paymentId}/status)	33
Established/initiated Payment Status Report Elements	33
Step II – Payment Authorisation Initiation – bank-specific (POST /my/payments/{paymentId}/sign/{signId }).....	35
Step II – Payment Authorisation Initiation – Bank-Specific.....	36
Balance Check (POST /my/payments/balanceCheck)	37
1.1. MESSAGE ELEMENTS Query for Balance Check	38
1.2. MESSAGE ELEMENTS Response for Balance Check	38
1.3. Return codes for the parameter „response“ – Code set:	38
9. PSD2 glossary – selected terms	39

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

Unauthorised payment types for Sandbox:

- Standing payment orders
- Direct debit orders/mandates
- Payment orders batches
- 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.koba.sk/portal/?tenant=api.kb.sk>. 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**. 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:

- 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.
- Remittance payment initiation – invalid IBAN error 400 - AC02 - [InvalidDebtorAccountNumber], debtorAccount.identification.iban field
- Remittance payment initiation – invalid amount error 400 - AM12 - [InvalidAmount], (amount.instructedAmount.value) field
- 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
- Remittance payment initiation – invalid field formats errors – returns a collection of errors

2. Payment status:

- Payment status – **ZU000007AX3** 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. 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.

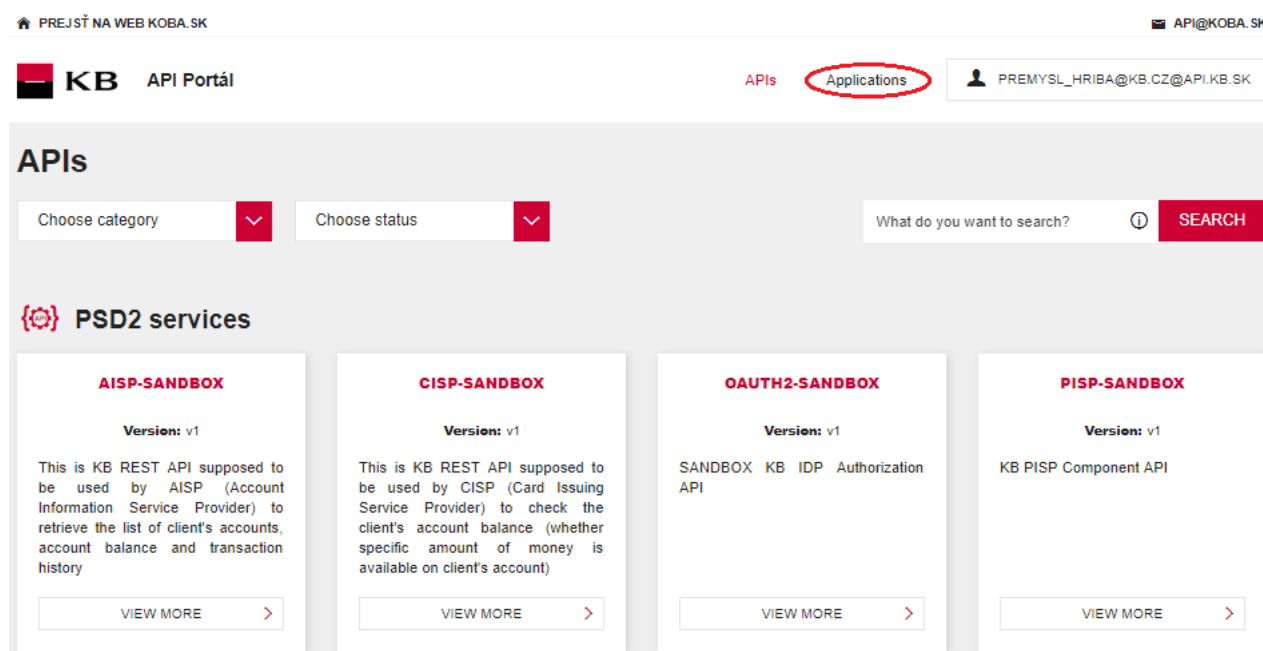
6. Procedure for generating PISP key/token

Prerequisites for key/token generation

The user is registered and logged in to the Sandbox SK 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.SK'. A red circle highlights the 'Applications' link in the top navigation bar. Below the navigation bar, the main content area is titled 'APIs'. It features a search bar with the text 'What do you want to search?' and a 'SEARCH' button. Below the search bar, there is a section titled 'PSD2 services' with a gear icon. This section contains four cards representing different services: 'AISP-SANDBOX', 'CISP-SANDBOX', 'OAUTH2-SANDBOX', and 'PISP-SANDBOX'. Each card displays the service name, version (v1), a brief description, and a 'VIEW MORE' button with a right 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.

 Filter by ...

Name	Tier	Workflow Status	Subscriptions	Actions
DefaultApplication	Unlimited	ACTIVE	0	  
Test_KOBA	Unlimited	ACTIVE	4	  

Show 10 entries Showing 1 to 2 of 2 entries

1

Selection of application menu functionality

User selects „SANDBOX KEYS“ in application menu

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

Test_KOBA

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 Sandbox 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_KOBA

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

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.koba.sk/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.koba.sk/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 it's scopes and validity period.

Validity period

[REGENERATE](#)

7. PISP Mock Calling Methods

1. [New payment establishing - mock calling through the API console](#)
2. [Payment status request - mock calling through the API console](#)
3. ["Payment Authorization" PIS mock calling for the purpose of testing \(through the API console\)](#)

4. [Payment initiation resource direct calling](#)
5. [Payment status resource direct calling](#)
6. Payment authorisation resource direct calling – currently not implemented in the Sandbox.

8. 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.

PREJÍT NA WEB KOBASK API@KOBASK

KB API Portál APIs Applications PREMYSL_HRIBA@KB.CZ@API.KB.SK

[GO BACK](#)

PISP-Sandbox

Version: v1 | Updated: 22/Jan/2020 12:05:50 PM CET | Status: PUBLISHED

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try TEST_KOBA ▼
 Using SANDBOX ▼
Key

Set Request Header

Authorization : Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.

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 ▼

Applications

DEFAULTAPPLICATION

▼

Tiers

UNLIMITED

▼

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.

PREJŠŤ NA WEB KOBA.SK

API@KOBA.SK

KB API Portál

APIs Applications

PREMYSL_HRIBA@KB.CZ@API.KB.SK

[GO BACK](#)

PISP-Sandbox

Version: v1
Updated: 22/Jan/2020 12:05:50 PM CET
Status: PUBLISHED

KB PISP Component API

API CONSOLE

DOCUMENTATION

Try
TEST_KOBA
Using
SANDBOX
Key

Set Request Header

Authorization : Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiU9.

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

Show/Hide | List Operations | Expand Operations

PISP

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": "EUR"
      }
    }
  }
}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
x-request-id		External Request ID	header	string
TPP-Name	(required)	Transaction initiator name	header	string

Model | Example Value

```
{
  "paymentIdentification": {
    "instructionIdentification": "NOTPROVIDED"
  },
  "paymentTypeInformation": {
    "instructionPriority": "NORM"
  }
}
```

paymentRequest

(required)

Parameter content type:

application/json

New payment request

body

“New Payment” operation error message

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

Response Messages

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

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

“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.

[PREJÍT NA WEB KOBAS.SK](#)
[API@KOBAS.SK](#)


KB

API Portál

APIs

Applications

PREMYSL_HRIBA@KB.CZ@API.KB.SK

[< GO BACK](#)

PISP-Sandbox

Version: v1
Updated: 22/Jan/2020 12:05:50 PM CET
Status: PUBLISHED

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try
TEST_KOBA
Using
SANDBOX
Key

Set Request Header

Authorization : Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.

[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 ▾

Applications
DEFAULTAPPLICATION
Tiers
UNLIMITED

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 ^

ⓘ

Required Scopes

Key	Name
pisp	pisp

Response Class (Status 200)

successful operation

Model

Example Value

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

Response Content Type

application/json

▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
x-request-id		External Request ID	header	string
TPP-Name		Transaction initiator name	header	string
paymentId	(required)	Unique bank transaction identification • example: 'ZU000007AX3'	path	string

“Payment status information” operation error message

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

Response Messages

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

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

“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.

[PREJÍŠ NA WEB KOBASK](#)
[API@KOBASK](#)


API Portál

APIs

Applications

 PREMYSL_HRIBA@KB.CZ@API.KB.SK

[← GO BACK](#)

PISP-Sandbox

Version: v1

Updated: 22/Jan/2020 12:05:50 PM CET

Status: **PUBLISHED**

KB PISP Component API

API CONSOLE

DOCUMENTATION

Try **TEST_KOBA**Using **SANDBOX**

Key

Set Request Header

Authorization : Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.

[Swagger \(/swagger.json \)](#)
[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

PISP

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 ▼

Applications

DEFAULTAPPLICATION

Tiers

UNLIMITED

 **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.

POST

Payment Authorization – starting the particular authorization method
/payments/{paymentId}/sign/{signId}

SHOW LESS ^

🕒 Required Scopes

Key	Name
pisp	pisp

Response Class (Status 200)

successful operation

Model Example Value

```

{
  "method": "string",
  "formData": {
    "SAMLRequest": "string",
    "relayState": "string"
  },
  "signInfo": {
    "state": "OPEN",
    "signId": "string",
    "signInfo": "ACTC"
  }
}

```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
x-request-id		External Request ID	header	string
TPP-Name		Transaction initiator name	header	string
paymentId	ZU000007AX3	Unique bank transaction identification example: 'ZU000007AX3'	path	string
signId	ZU000007AX3	Identifier of the change	path	string
authInitiationRequest	(required) Parameter content type: application/json	Initiation request identifier of the authorization method and back URL	body	Model Example Value <pre> { "authorizationType": "USERAGENT_REDIRECT", "backUrl": "string" } </pre>

“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](#)
[Hide Response](#)

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.

PREJÍŠ NA WEB KOBA.SK API@KOBA.SK

 **KB** API Portál
 APIs
Applications
PREMYSL_HRIBA@KB.CZ@API.KB.SK

[< GO BACK](#)

PISP-Sandbox

Version: v1 | Updated: 22/Jan/2020 12:05:50 PM CET | Status: PUBLISHED

KB PISP Component API

API CONSOLE

DOCUMENTATION

Try TEST_KOBA v Using SANDBOX v Key

Set Request Header

Authorization : Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.

Swagger (/swagger.json)

PISP

Show/Hide | List Operations | Expand Operations

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

Applications

v

Tiers

v

S
SUBSCRIBE

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

POST

Balance check
/payments/balanceCheck

SHOW LESS

🕒

Required Scopes

Key

Name

pisp

pisp

Response Class (Status 200)

successful operation

Model

Example Value

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

Response Content Type

application/json

▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
x-request-id		External Request ID	header	string
TPP-Name	(required)	Transaction initiator name	header	string

balanceCheckRequest

(required)

Parameter content type:
application/json

▼

Query for Balance Check

body

Model

Example Value

```
{
  "exchangeIdentification": "103149078",
  "card": {
    "cardHolderName": "Jan Novák",
    "maskedPAN": "1234****6789"
  }
}
```

9. 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.koba.sk/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
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
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.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.

Response header parameters:

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

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
SEPA	ESCT	SEPA payment (domestic and within the EEA)
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]	SEPA ZPL	Max35Text	End To End identification.
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInformation	[0..1]	ALL	PaymentTypeInformation19	Information about the payment type
++	instructionPriority	[0..1]	ALL	Priority2Code	Priority of the instruction
++	serviceLevel	[0..0]	ALL	ServiceLevel8CZ	Service level
+++	code	[0..0]	ALL	ExternalServiceLevel1Code	Service level code
++	categoryPurpose	[0..0]	ALL	CategoryPurpose1Choice	Payment purpose category
+++	code	[0..0]	ALL	ExternalCategoryPurpose1Code	Payment purpose category code
+++	proprietary	[0..0]	ALL	Max35Text	Payment purpose category in the free format
+	amount	[1..1]	ALL	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]	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]	SEPA ZPL	PartyIdentification32CZ1	Ultimate debtor.
++	name	[0..0] [0..1] [0..0]	SEPA ZPL	Max70Text	Ultimate debtor's name.
++	postalAddress	[0..0] [0..1] [0..0]	SEPA ZPL	PostalAddress6CZ	Ultimate debtor's postal address.
+++	streetName	[0..0] [0..1] [0..0]	SEPA ZPL	Max70Text	Ultimate debtor's street.
+++	buildingNumber	[0..0] [0..1] [0..0]	SEPA ZPL	Max16Text	Ultimate debtor's building number.
+++	postCode	[0..0] [0..1] [0..0]	SEPA ZPL	Max16Text	Ultimate debtor's Postal Code.
+++	townName	[0..0] [0..1] [0..0]	SEPA ZPL	Max35Text	Ultimate debtor's town/city.
+++	country	[0..0] [0..1] [0..0]	SEPA ZPL	CountryCode ISO3166	Ultimate debtor's country.
+++	addressLine	[0..0] [0..2] [0..0]	SEPA ZPL	Max70Text	Unstructured record of the ultimate debtor's address.
++	identification	[0..0] [0..1] [0..0]	SEPA ZPL	Party6Choice	Ultimate debtor's identification.
+++	organisationIdentification	[0..0] [1..1] [0..0]	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]	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]	SEPA ZPL	GenericOrganisationIdentification1	Other identification of the ultimate debtor as an organization/legal person.
+++++	identification	[0..0] [1..1] [0..0]	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]	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]	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]	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]	SEPA ZPL	PersonIdentification5CZ	Unique identification of the ultimate debtor as a natural person. Either organisationIdentification or privateIdentification.
++++	other	[0..0] [0..1] [0..0]	SEPA ZPL	GenericPersonIdentification1	Other identification of the ultimate debtor as a natural person in the unstructured form.
+++++	identification	[0..0] [1..1] [0..0]	SEPA ZPL	Max35Text	Other identification of the ultimate debtor as a natural person in the unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0]	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]	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]	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]	SEPA ZPL	BranchAndFinancialInstitutionIdentification4CZ	Creditor's bank.
++	financialInstitutionIdentification	[0..0] [0..1] [1..1]	SEPA ZPL	FinancialInstitutionIdentification7CZ	Identification of the financial institution.
+++	bic	[0..0] [1..1] [0..1]	SEPA ZPL	BICIdentifier	BIC / SWIFT bank code.
+++	clearingSystemMemberIdentification	[0..0] [0..0] [0..1]	SEPA ZPL	ClearingSystemMemberIdentification2	Clearing system member identification.
++++	clearingSystemIdentification	[0..0] [0..0] [0..1] ... buď anebo některá z jiných variant identifikace banky příjemce	SEPA ZPL	ClearingSystemIdentification2Choice	Clearing system identification.
+++++	code	[0..0] [0..0] [1..1]... buď code anebo proprietary	SEPA ZPL	ExternalClearingSystemIdentification1Code	Code
+++++	proprietary	[0..0] [0..0] [1..1]... buď code anebo proprietary	SEPA ZPL	Max35Text	Free format
++++	memberIdentification	[0..0] [0..0] [1..1]	SEPA ZPL	Max35Text	Member's clearing code
+++	name	[0..0] [0..0] [0..1]... buď anebo některá z jiných variant identifikace banky příjemce	SEPA ZPL	Max70Text	Name

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

++	name	[0..0] [1..1] [1..1]	SEPA ZPL	Max70Text	Creditor's name
++	postalAddress	[0..0] [0..1] [1..1]	SEPA ZPL	PostalAddress6CZ	Postal address
+++	streetName	[0..0] [0..1] [0..1]	SEPA ZPL	Max70Text	Street
+++	buildingNumber	[0..0] [0..1] [0..1]	SEPA ZPL	Max16Text	Building number
+++	postCode	[0..0] [0..1] [0..1]	SEPA ZPL	Max16Text	Postal Code
+++	townName	[0..0] [0..1] [0..1]	SEPA ZPL	Max35Text	Town/City
+++	country	[0..0] [0..1] [0..1]	SEPA ZPL	CountryCode ISO3166	Country
+++	addressLine	[0..0] [0..2] [0..2]	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]	SEPA ZPL	GenericAccountIdentification1CZ:	Account number in other format.
++++	identification	[0..0] [0..0] [1..1]	SEPA ZPL	Max34Text	Account number in the local BBAN format.

++	currency	[0..1] [0..0] [0..0]	SEPA ZPL	CurrencyCode ISO4217	Creditor's account currency.
+	ultimateCreditor	[0..0] [0..1] [0..0]	SEPA ZPL	PartyIdentification32CZ1	Ultimate creditor.
++	name	[0..0] [0..1] [0..0]	SEPA ZPL	Max70Text	Name
++	postalAddress	[0..0] [0..1] [0..0]	SEPA ZPL	PostalAddress6CZ	Postal address
+++	streetName	[0..0] [0..1] [0..0]	SEPA ZPL	Max70Text	Street
+++	buildingNumber	[0..0] [0..1] [0..0]	SEPA ZPL	Max16Text	Building number
+++	postCode	[0..0] [0..1] [0..0]	SEPA ZPL	Max16Text	Postal Code
+++	townName	[0..0] [0..1] [0..0]	SEPA ZPL	Max35Text	Town/City
+++	country	[0..0] [0..1] [0..0]	SEPA ZPL	CountryCode ISO3166	Country
+++	addressLine	[0..0] [0..2] [0..0]	SEPA ZPL	Max70Text	Unstructured record of the address.
++	identification	[0..0] [0..1] [0..0]	SEPA ZPL	Party6Choice	Ultimate creditor's identification.

+++	organisationIdentification	[0..0] [1..1] [0..0]	SEPA ZPL	OrganisationIdentification4CZ	Unique identification of the ultimate creditor as an organization/ legal person. Either organisationIdentification or privateIdentification.
++++	bicOrBei	[0..0] [0..1] [0..0]	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]	SEPA ZPL	GenericOrganisationIdentification1	Other identification of the ultimate creditor as an organization/ legal person.
+++++	identification	[0..0] [1..1] [0..0]	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]	SEPA ZPL	OrganisationIdentificationSchemeName1CZ	Type of the document used for the identification of the ultimate creditor as an organization/legal person.
+++++	proprietary	[0..0] [1..1] [0..0]	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]	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]	SEPA ZPL	PersonIdentification5CZ	Unique identification of the ultimate creditor as a natural person. Either organisationIdentification or privateIdentification.
++++	other	[0..0] [0..1] [0..0]	SEPA ZPL	GenericPersonIdentification1	Other identification of the ultimate creditor as a natural person in the unstructured form.
+++++	identification	[0..0] [1..1] [0..0]	SEPA ZPL	Max35Text	Other identification of the ultimate creditor as a natural person in the unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0]	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]	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]	SEPA ZPL	Max35Text	Issuer of the document used for the identification of the ultimate creditor as a natural person.
+	purpose	[0..0]	ALL	Purpose2Choice	Payment purpose
++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code
++	proprietary	[0..0]	ALL	Max35Text	Payment purpose in the free format
+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instruction for the next bank
+	remittanceInformation	[0..1]	ALL	RemittanceInformation5CZ	Information about the payment
++	unstructured	[0..1]	ALL	Max140Text, consisting of alphanumeric characters supported by CERTIS (CNB clearing), including supported special characters	Unstructured report for the creditor (see below*)
++	structured	[0..1] [0..0] [0..0]	SEPA ZPL	StructuredRemittanceInformation7CZ	Structured message for the creditor – variable, specific, and constant symbol.
+++	creditorReferenceInformation	[0..1] [0..0] [0..0]	SEPA ZPL	CreditorReferenceInformation2CZ	Creditor reference information.
++++	reference	[0..3] [0..0] [0..0]	SEPA ZPL	CreditorReferenceInformation2CZ	VS, SS & KS values

ZPL = cross-border payment

EHP = EEA

*** remittanceInformation.unstructured** - (this information is optional): used to further identify the payment. In Slovakia, payment symbols (variable, specific and constant symbol) are used for this purpose. They are also transmitted in the "EndToEndIdentification" field in the sense of nationally agreed rules.

Note to the field **remittanceInformation.unstructured**: according to the standard of the Czech Banking Association, this field may also contain information about VS, KS and SS, however the Bank will work with all information contained here as if it were a simple description of the payment (information for beneficiaries) and none symbol parsing will not occur even if they occur here.

- 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 element: "Payer reference":

The form of entering symbols according to the Slovak national convention is not arbitrary, but is prescribed as follows:
/VS1234567890/SS1234567890/KS1234 or /VS1234567890/SS/KS1234

The following conditions are also applied:

In the case of a foreign SEPA payment, the "Payer's reference" field:

1. must not contain the character "/" in the first or last position,
2. must not contain "/" (two slashes in a row).

We would also like to point out that in the processing of AISP information, the Bank does not evaluate all slashes entered in the PISP in the **Remittance Information.Unstructured** field as valid, resp. semantic and in the AISP output it is no longer mentioned.

For example:

"VS: 36718 //", for a payment in the AISP, enter "VS: 36718", ie all slashes are discarded by the system.

New Payment – Payment Initiation response elements

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

LEVEL	MESSAGE ELEMENT	OCCUR- RENCE	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
DMCT	[DoMestic Credit Transfer] Domestic payment. Not supported SK
ESCT	[SEPA Credit Transfer] – SEPA payment
XBCT	[Cross-Border Credit Transfer] – Cross border payment

Payment status codes – 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	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.koba.sk/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
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
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.

Response header parameters:

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

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
401	UNAUTHORISED	Invalid/missing certificate = the provider has not been authenticated
404	TRANSACTION_MISSING	Calling of a method that does not match with the licence, or invalid certificate.

Established/initiated Payment Status Report Elements

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

Payment codes status – StatusCode

HTTP STATUS CODE	STATUS CODE	PURPOSE
------------------	-------------	---------

200	ACTC	[AcceptedTechnicalValidation] – The authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACSP	[AcceptedSettlementInProgress] – All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution
200	ACSC	[AcceptedSettlementCompleted] – Settlement on the debtor's account has been completed. Usage: this can be used by the first agent to report to the debtor that the transaction has been completed. Warning: this status is provided for transaction status reasons, not for financial information. It can only be used after bilateral agreement
200	ACWC	[AcceptedWithChange] – An instruction is accepted but a change will be made, such as date or remittance not change

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.koba.sk/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	MANDA-TORY	PURPOSE
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
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.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.

Response header parameters:

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

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

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

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = the user has not been authenticated
403	FORBIDDEN	Invalid/missing certificate = the provider has not been authenticated
404	ID_NOT_FOUND	The required id does not exist

400	AUTH_LIMIT_EXCEED	This resource cannot be authorised in this manner
-----	-------------------	---

Step II – Payment Authorisation Initiation – Bank-Specific

Request parameters:

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

Response parameters:

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

Balance Check (POST /my/payments/balanceCheck)

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

Resource characteristics

URI: /my/payments/balanceCheck
HTTP Method: POST
Request URL: <https://api.koba.sk/serverapi/pisp/v1/payments/balanceCheck>
Authorization: request **requires** the authorization of user/client as part of API calling
Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no
Sorting: no
Filtering: no

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

Parameters of the request header:

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

Parameters of the response header:

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

The content of POST request and response for calling, please see Chapter 7.1. MESSAGE ELEMENTS Query for balance check.

Error codes defined for the POST service Query for balance check

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
400	AC12	[InvalidAccountType] - account type does not match allowed account types (e.g., a non-paying account).
403	AG01	[TransactionForbidden] – absent consent to access to balance check at the account.
400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.

400	FF01	[Invalid File Format] – invalid JSON format or 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.

1.1. MESSAGE ELEMENTS Query for Balance Check

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	exchangeIdentification	[1..1]	Max18Text	Clear query identification
+	card	[0..1]	±	Transaction card
++	cardholderName	[0..1]	Max45Text	Card holder name
++	maskedPan	[1..1]	Max30Text	Masked card number
+	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

1.2. MESSAGE ELEMENTS Response for Balance Check

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	responseIdentification	[1..1]	Number (integer)	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.

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

CODE	DESCRIPTION
------	-------------

APPR	Enough funds on this account
DECL	Unsufficient funds on this account

10. 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 or other EU National Competent Authority)

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)