The Payment Initiation Service Manual

PIS

Change log

Date of publishing	Version	Date of effectiveness	Description
3.12.2018	1		First document version
29.4.2019	2	10.8.2019	New chapters 5, 6 and 7 added
29.4.2019	2	10.8.2019	Update of all request headers: TPP-Name, TPP-Identification
6.12.2019	3	6.12.2019	Removal of client's PIS consent, adding new priority INST for instant payments.
20.3.2020	4	20.4.2020	New chapter for Standing orders
19.5.2020	5	20.5.2020	Document edit.
2. 7. 2020	6	2. 7. 2020	Format edit (fixed bug)
17.9.2021	7	20.9.2021	Addition of response elements (openDate and lastDate) in chapter 12.2. MESSAGE ELEMENTS Standing order details – RESPONSE
12.04.2022	8	12.04.2022	New chapter for Batch payments
31.01.2023	9	01.02.2023	Migration of the new version of the PSD2 PISP API service: 1. New Payment – Payment Initiation 2. Payment Authorization Initiation 3. Established/initiated Payment Status 4. Info on entered/initiated payment 5. Query for Balance Check Summary of the most important changes: Change in request URL - adding **"my"** in the path and upgrading to v2, i.e. e.g.: a) for initiation , status, detail and autorization /serverapi/pisp/**v2/my/**payments... b) for balance check /serverapi/pisp/**v2/my/**balanceCheck Change in the request for authorization initialization - the original backURL parameter was renamed to redirectUrl - the meaning and description remain the same (synchronization with the standard).
26.4.2023	10	26.7.2023	**Modified description of ACSP and ACSC statuses** in chapters 4.1., 12.1. and 15.1. For ZPL and SEPA payments, payment symbols VS, KS are parsed only from the field **remittanceInformation.unstructured.creditorReferenceInformation.reference** For TUZEM payments, payment symbols VS, KS, SS are parsed from the field **remittanceInformation.structured.creditorReferenceInformation.reference** - see chapters 2.1. and 9.2.
20.7.2023	11	20.10.2023	Modified values of the selected 3 sub-fields (streetName, townName, country) in the field **creditor.postalAddress** in foreign payments (ZPL), they must be always provided [1..1].

Differences from Czech Open Banking Standard are highlighted yellow.

Contents

The Payment Initiation Service (PIS) Manual

1. The Payment Initiation Service (PIS)

KB issues the Payment Initiation Service (PIS) API, which contains a description of payment initiation services, the payment authorisation, and information on the payment status.

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 differences between the KB implementation mode and the published standard are described in this document and they are marked in yellow colour.

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

The allowed character set is based on the CERTIS character set (for domestic payments) and on the SWIFT character set (for cross-border payments/SEPA payments). Only one query can be sent and processed during a single call.

Payment processing:

- The multiple/multilevel authorisation is not supported in the case of payments initiated via the Payment Initiation API. However, a transaction may be authorised separately unless its amount is higher than the limit assigned to the Authoriser A.
- The Payment Initiation API service can only be applied to payment accounts.

Time limitations:

- The payments will always be processed in an on-line mode before 20:30 hrs, with the due date falling on the same business day.
- The due date of the payments received for the processing on a KB business day between 20:31 and 23:59 hrs will fall on the next succeeding business day.
- **Payments with forward due dates cannot be made** using the Payment Initiation API service (with the exception of payments from accounts in KB+). However, urgent and conversion payments can be submitted with the due date falling on the next succeeding business day even after the expiry of the respective cut-off time (COT).

Viewing the payments via direct banking channels:

- The user cannot cancel or alter the payments arranged via the Payment Initiation API service and received by the bank for the processing. The payments are irrevocable.
- Payments "for authorisation" are not viewed in any of the lists.

List of resources:

1. POST new payment - payment initiation
2. GET established/initiated payment status
3. POST Step II. Payment authorisation initiation
4. POST query for Balance Check
5. DELETE delete an entered payment
6. POST batch payments
7. POST batch payment authorization
8. GET batch payment status

Unsupported payment types:

1. Direct debit orders/mandates
2. Cheque payments
3. FX payments
4. Payments with double conversion

1.1. Error reporting

Reporting production errors or errors within particular calling 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.

The following must be specified:
PSD2 API domain: ***CZ*** *or* ***SK***
Environment: ***Sandbox*** *or* ***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 (type of PSD2 service)
Date and time of the call
IP address
The error specification and its most accurate description (incl.. „***x-request-id***“), which can be supplemented with the appropriate screenshot.

Without the above values, it is not possible to solve the reported error and KB may ask you to complete the necessary information (it may prolong the fixing the error).

2. New Payment – Payment Initiation (POST /my/payments)

Resource for establishing a new payment.

Resource characteristics

URI:	/my/payments
HTTP Method:	POST
Request URL:	https://api.kb.cz/serverapi/pisp/v2/my/payments
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:

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.
Authorisation	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
Date	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
X-request-id	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
User-involved	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
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 **application/json** format is primarily supported.

For the content of the request and response call POST see Chapter 3.1. New Payment – Payment Initiation MESSAGE ELEMENTS.

Error codes defined for the 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.
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] – creditor account number is invalid or missing.
400	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
400	AC12	[InvalidAccountType] – the account type does not match allowed account types (e.g., a non-paying account).
403	AG01	[TransactionForbidden] – absent consent to access to the account balance check.
400	AM11	[InvalidTransactionCurrency] – the request contains a currency that is not traded/supported.
400	AM12	[InvalidAmount] – 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] – 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] – the request identifier is not unique.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.

2.1. New Payment – Payment Initiation MESSAGE ELEMENTS

Considered payment types

Komerční banka processes SEPA payments as foreign currency payments (FCP). All SEPA-specific information is removed during the processing.

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*	*OCCUR-RENCE*	*PAYMENT TYPE*	*FORMAT TYPE*	*PRESENTATION*
+	paymentIdentification	[1..1]	ALL	PaymentIdentification1	Payment identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification
++	endToEndIdentification	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	End To End identification. The field is not required at the entry for SEPA.
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInformation	[0..1]	ALL	PaymentTypeInformation19	Information about the payment type
++	instructionPriority	[0..1]	ALL	Priority2Code. Allowed values: - NORM - HIGH	Priority of the instruction. The HIGH priority is translated to EXPRES

				- INST	for domestic payments. The NORM priority is translated to EXPRES and HIGH to URGENT for cross/border/SEPA payments. If the field is blank, we will fill in the standard priority (NORM).
++	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 částka a měna Not supported
+++	value	[0..0]	ALL	Amount	Equivalent transaction amount Not supported
+++	currency	[0..0]	ALL	CurrencyCode	Currency of the equivalent transaction amount Not supported
+	requestedExecutionDate	[0..1]	ALL	ISODate	Requested execution date of the payment. If the field is blank, we fill in a business day as per KB rules.
+	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. Mapping of the charges in the response to the payment initiation: DEBT = OWN CRED = BEN SHAR = SHA

					SLEV = SLV
+	chargesAccount	[0..0]	ALL	CashAccount16CZ	Charges account Not supported
++	identification	[0..0]	ALL	AccountIdentification4 ChoiceCZ	Charges account identification Not supported
+++	iban	[0..0]	ALL	IBAN2007Identifier	Charges account IBAN number Not supported
++	currency	[0..0]	ALL	CurrencyCode ISO 4217	Charges account currency Not supported
+	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..0] [0..0]	TUZEM SEPA ZPL	PostalAddress6CZ	Ultimate debtor's postal address. Not supported under SEPA.
+++	streetName	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max70Text	Ultimate debtor's street. Not supported under SEPA.
+++	buildingNumber	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max16Text	Ultimate debtor's building number. Not supported under SEPA.
+++	postCode	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max16Text	Ultimate debtor's Postal Code. Not supported under SEPA.
+++	townName	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max35Text	Ultimate debtor's town/city. Not supported under SEPA.
+++	country	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	CountryCode ISO3166	Ultimate debtor's country. Not supported under SEPA.

+++	addressLine	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max70Text	Unstructured record of the ultimate debtor’s address. Not supported under SEPA.
++	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	OrganisationIdentification4CZ	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.
++++++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code.
++++++	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 organisationIdentificatio

					n 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.
++++++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code.
++++++	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.
++	identification	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Party6Choice	Ultimate debtor’s identification
+++	organisationIdentifi	[0..0]	TUZEM	OrganisationIdentificati	Unique identification of

	cation	[1..1] [0..0]	SEPA ZPL	on4CZ	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.
++++++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code.
++++++	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.
++++++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code.

++++++	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.
+	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] [1..1]	TUZEM SEPA ZPL	BICIdentifier	BIC / SWIFT bank code
+++	clearingSystemMemberIdentification	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	ClearingSystemMemberIdentification2	Clearing system member identification.
++++	clearingSystemIdentification	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	ClearingSystemIdentification2Choice	Clearing system identification.
+++++	code	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	ExternalClearingSystemIdentification1Code	Code.
+++++	proprietary	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max35Text	Free format.
++++	memberIdentification	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max35Text	Member’s clearing code
+++	name	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max70Text	Name
+++	postalAddress	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	PostalAddress6CZ	Postal address
++++	streetName	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max70Text	Street
++++	buildingNumber	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max16Text	Building number
++++	postCode	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max16Text	Postal Code

++++	townName	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max35Text	Town/City
++++	country	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	CountryCode ISO3166	Country
++++	addressLine	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max70Text	Unstructured record of the address. Not supported.
+++	other	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	GenericFinancialIdentification1C	Other identification of the bank. Not supported.
++++	identification	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max35Text	Bank’s local code. Not supported.
+	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] [1..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] [1..1]	TUZEM SEPA ZPL	Max35Text	Town/City

+++	country	[0..0] [0..1] [1..1]	TUZEM SEPA ZPL	CountryCode ISO3166	Country
+++	addressLine	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max70Text	Unstructured record of the address. Not supported.
++	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	OrganisationIdentification4CZ	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.
++++++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code.
++++++	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.
++++++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code.
++++++	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.
+	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..0] [0..0]	TUZEM SEPA ZPL	PostalAddress6CZ	Postal address. Not supported.
+++	streetName	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max70Text	Street. Not supported.
+++	buildingNumber	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max16Text	Building number. Not supported.
+++	postCode	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max16Text	Postal Code Not supported.
+++	townName	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max35Text	Town/City Not supported.
+++	country	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	CountryCode ISO3166	Country Not supported.
+++	addressLine	[0..0] [0..0] [0..0]	TUZEM SEPA ZPL	Max70Text	Unstructured record of the address Not supported.
++	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	OrganisationIdentification4CZ	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	OrganisationIdentificationSchemeName1CZ	Type of the document used for the identification of the ultimate creditor as an organization/legal person.
++++++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code.
++++++	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.

++++++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code.
++++++	proprietary	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Type of the document used for the identification of the ultimate creditor as a natural person in the free text format.
+++++	issuer	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Issuer of the document used for the identification of the ultimate creditor as a natural person.
+	purpose	[0..0]	ALL	Purpose2Choice	Payment purpose
++	code	[0..0]	ALL	ExternalPurpose1Code	Payment purpose code
++	proprietary	[0..0]	ALL	Max35Text	Payment purpose in the free format
+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instruction for the next bank
+	remittanceInformation	[0..1]	ALL	RemittanceInformation5CZ	Information about the payment
++	unstructured	[0..1]	ALL	Max140Text, consisting of alphanumeric characters supported by CERTIS (CNB clearing), including supported special characters	Unstructured message for creditor (pls see below ***)**
++	structured	[0..1] [0..0] [0..0]	TUZEM SEPA ZPL	StructuredRemittanceInformation7CZ	Structured message for the creditor – variable, specific, and constant symbol ****)**
+++	creditorReferenceInformation	[0..1] [0..0] [0..0]	TUZEM SEPA ZPL	CreditorReferenceInformation2CZ	Creditor reference information
++++	reference	[0..3] [0..0] [0..0]	TUZEM SEPA ZPL	CreditorReferenceInformation2CZ	VS, SS & KS values

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

)** Note concerning the ***remittanceInformation.unstructured.creditorReferenceInformation.reference field: According to the Czech Banking Association standard, this field may also contain information about VS, KS and SS payment symbols; however:

a) For **TUZEM** (domestic) payments, we work with the information contained here as if it were a plain text description of the payment (information for the creditor) and no parsing of symbols will occur, even if they occur here.

b) For **ZPL/SEPA payments**, this field is used to pars payment symbols (VS or KS), i.e. they are recognized here and stored separately in the respective fields for these symbols 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"

****)** 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 - this only applies to TUZEM (domestic) payments.
In the case of ZPL/SEPA payments (unlike TUZEM payments), this field is ignored and the payment symbols (VS or KS) are obtained from the remittanceInformation.unstructured field.

2.2. 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 Creidt Transfer] Domestic payment
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. RJCT status is not supported as part of the response to the initiation.
200	ACWC	[AcceptedWithChange] – An instruction is accepted but a change will be made, such as date or remittance not sent.

3. 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.

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

Resource characteristics

URI:	/my/payments/{paymentId}/sign/{signId}
HTTP Method:	POST
Request URL:	https://api.kb.cz/serverapi/pisp/v2/my/payments/{paymentId}/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:

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.
Authorisation	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
Date	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
X-request-id	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
User-involved	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
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'

Path parameter requestu:

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

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 5.1. Payment Authorisation initiation message elements – 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 = the user has not been authenticated
403	FORBIDDEN	Invalid/missing certificate = the provider has not been authenticated
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	This resource cannot be authorised in this manner

3.1. Payment Authorisation initiation message elements

Parameters of the request:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	authorisationType	[1..1]	PISP ALL	Text	The code of the required authorisation (from authorisation scenarios)
+	redirectURL	[1..1]	PISP ALL	Text	The URL link for the return of a response concerning the executed authorisation. If the due date (required date of payment) is changed during the authorisation, such a change will be sent to the provided backURL as a changeInfo parameter with a new date.

Parameters of the response:

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

4. Established/Initiated Payment Status(GET /my/payments/{paymentId}/status)

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: /my/payments/{paymentId}/status
HTTP Method: GET
Request URL: https://api.kb.cz/serverapi/pisp/v2/my/payments/{paymentId}/status
Authorisation: 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**

Parameters of the request header:

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.
Date	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
X-request-id	text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
User-involved	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
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'

Path parameter of the request:

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

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 4.1. Established/initiated Payment Status REPORT ELEMENTS

Error codes defined 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.

4.1. 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 Status code – 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. In the case of RJCT, we return the reason for non-payment along with the status code.
200	ACSP	[AcceptedSettlementInProcess] - All preceding checks such as technical validation and customer profile were successful ~~and therefore the payment initiation has been accepted for execution~~, however it is not a confirmation of final processing on the part of debtor ASPSP´s side. This is an intermediate state that can result either in ACSC or RJCT. To obtain the status of entered payment call (GET /my/payments/{paymentId}).
200	ACSC	[AcceptedSettlementCompleted] - Settlement on the debotor´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. For the 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 sent.

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

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

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

Resource characteristics

URI:	/my/payments/{paymentId}
HTTP Method:	GET
Request URL:	https://api.kb.cz/serverapi/pisp/v2/my/payments/{paymentId}
Authorization:	request **requires** the authorization of user/client as part of the API calling
Use certificate:	request **requires** the use of the third-party qualified certificate

Paging:	**no**
Sorting:	**no**
Filtering:	**no**

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

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
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.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.
Date	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
X-request-id	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
User-involved	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
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

Path parameter of the request:

PARAMETER	TYPE	MANDA-TORY	PURPOSE
paymentId	Text	Yes	Identifier of the established payment.

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 6.1. MESSAGE ELEMENTS Info on entered/initiated payment

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

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

5.1. MESSAGE ELEMENTS Info on entered/initiated payment

The result of the message is an overview of entered or already initiated payment. Therefore, the list of elements corresponds to the elements from resource New payment, see 3.1. New Payment – Payment Initiation MESSAGE ELEMENTS.

6. Deleting the entered unauthorised payment (DELETE /payments/{paymentId})

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

Resource characteristics

URI:	/payments/{paymentId}
HTTP Method:	DELETE
Request URL:	https://api.kb.cz/serverapi/pisp/v1/payments/{paymentId}
Authorization:	request **requires** the authorization of user/client as part of the API calling
Use certificate:	request **requires** the use of the third-party qualified certificate

Paging:	**no**
Sorting:	**no**
Filtering:	**no**

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

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
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.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type.
X-request-id	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
TPP-Name	Text	No	The name of the original TPP that created the request. Eg. ‘Star corporation, a.s.’
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*	*MABDATORY*	*PURPOSE*
Content-Type	Text	Yes	Specification of required transfer format. From the precondition of technical specification of this API standard, in this case, **application/json** format is primarily supported.

Error codes defined for the service DELETE the entered unauthorised payment:

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

7. Query for Balance Check (POST /my/payments/balanceCheck)

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

Resource characteristics

URI:	/my/payments/balanceCheck
HTTP Method:	POST
Request URL:	https://api.kb.cz/serverapi/pisp/v2/my/payments/balanceCheck
Authorization:	request **requires** the authorization of user/client as part of API calling
Use certificate:	request **requires** the use of the qualified third-party certificate

Paging:	**no**
Sorting:	**no**
Filtering:	**no**

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

Parameters of the request header:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
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.
Authorization	Text	Yes	The parameter is used to pass an access token of the authenticated user together with its type
Date	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
X-request-id	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
User-involved	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
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 8.1. MESSAGE ELEMENTS Query for Balance Check.

Error codes defined for the POST service Query for balance check

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
400	AC12	[InvalidAccountType] - account type does not match allowed account types (e.g., a non-paying account).
403	AG01	[TransactionForbidden] – absent consent to access to balance check at the account.
400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.
400	FF01	[Invalid File Format] – invalid JSON forma tor other technical problem with the query processing.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RF01	[NotUniqueTransactionReference] – not unique request identifier.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.

7.1. 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

7.2. MESSAGE ELEMENTS Response for Balance Check

LEVEL	*MESSAGE ELEMENT*	*OCCURRENCE*	*FORMAT TYPE*	*PRESENTATION*
+	responseIdentification	[1..1]	Number (integre)	Unique identification of response to query for Balance Check (from ASPSP).
+	exchangeIdentification	[1..1]	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.

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

CODE	*DESCRIPTION*
APPR	Enough funds on this account
DECL	Unsufficient funds on this account or User Account Limit, Subject Limit or Payment Initiation User Limit exceeded

The Standing Order Initiation Service (PIS STO) Manual

8. The Standing Order Initiation Service (PIS STO)

KB issues the Payment Initiation Service (PIS) API, which contains a description of standing order initiation services, the standing order authorisation, and information on the standing order status.

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 differences between the KB implementation mode and the published standard are described in this document and they are marked in yellow colour.

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

The allowed character set is based on the CERTIS character set (for domestic payments) and on the SWIFT character set (for cross-border payments/SEPA payments). Only one query can be sent and processed during a single call.

Payment processing:

- The multiple/multilevel authorisation is not supported in the case of payments initiated via the Payment Initiation API. However, a transaction may be authorised separately unless its amount is higher than the limit assigned to the Authoriser A.
- The Payment Initiation API service can only be applied to payment accounts.

Time limitations:
A standing order can only be submitted with a due date of D + 2 (where D = current business day).

Viewing the payments via direct banking channels:

- Payments “for authorisation” are not viewed in any of the lists.

List of resources:

9. POST new payment – standing order initiation
10. GET established/initiated standing order status
11. GET detail established standing order
12. POST Step II. Standing order authorisation initiation

Unsupported payment types:

5. DELETE – removal standing order
6. SEPA standing order, EHP standing order, non EHP standing order
7. Standing order stoppage
8. Cheque payments
9. Irregular standing order

9. Standing order

9.1. New Payment – Payment Initiation (POST /payments)

Resource for entering a new standingorder.

Resource characteristics

URI: /standingorders
HTTP Method: POST
Request URL: https://api.kb.cz/serverapi/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*
Content-Type	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát **application/json**.
Authorization	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.
X-request-id	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
TPP-Name	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“ V tomto poli jsou podporovány pouze znaky bez diakritiky.
TPP-Identification	Text	Ne	Identifikace původního TPP, které request vytvořilo. Např.: „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 **application/json** format is primarily supported.

For the content of the request and response call POST see Chapter 10.2. New Payment – standing orders Initiation message 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 požíváno.
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 forma tor 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)

9.2. New Payment – standing orders Initiation message elements

Considered standing orders types

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

LEVEL	MESSAGE ELLEMENT	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 ifnormation 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 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 reqularly 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	[0..1]	ALL		Elements restricting the validity of the standing order.
+++	lastExecutionDate	[0..1]	ALL	ISODate	Date when the last order will be processed. Null value responds to ending by

					the user.
+++	maxIterations	[0..1]	ALL	Number	Maximum number of iterations - processing of the standing order. Only applicable in combination with executionMode AFTER_MAX_ITERATION_EXCEEDED.
+++	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	[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] [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 …	*), **) the same context as in the chapter on initiating payments

)** Note concerning the ***remittanceInformation.unstructured.creditorReferenceInformation.reference field: According to the Czech Banking Association standard, this field may also contain information about VS, KS and SS payment symbols; however:

a) For **TUZEM** (domestic) payments, we work with the information contained here as if it were a plain text description of the payment (information for the creditor) and no parsing of symbols will occur, even if they occur here.

b) For **ZPL/SEPA payments**, this field is used to pars payment symbols (VS or KS), i.e. they are recognized here and stored separately in the respective fields for these symbols 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"

****)** 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 - this only applies to TUZEM (domestic) payments.
In the case of ZPL/SEPA payments (unlike TUZEM payments), this field is ignored and the payment symbols (VS or KS) are obtained from the remittanceInformation.unstructured field.

9.3. 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 estabilished standing order instruction.
+	paymentTypeInformation	[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 istructionStatus "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

10. 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/serverapi/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:

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.
Authorisation	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
X-request-id	Text	Ne	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
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	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 11.1. Payment Authorisation initiation message elements – 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

10.1. Payment Authorisation initiation message elements

Parameters of the request:

ÚROVEŇ	*PRVEK ZPRÁVY*	*VÝSKYT*	*TYP FORMÁT*	*ÚČEL*
+	authorizationType	[1..1]	Text	Code of required authorization (from authorization scenarios)
+	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.

Response parameters of Standing order authorization initiation:

ÚROVEŇ	*PRVEK ZPRÁVY*	*VÝSKYT*	*TYP FORMÁTU*	*PREZENTACE*
+	authorizationType	[1..1]	±	Code of required *authorization (from authorization scenarios)*
+	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

11. Standing order detail (GET /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/serverapi/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:	no
Sorting:	no
Filtration:	no

Query parameters of the request:

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

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.
Authorisation	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
X-request-id	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
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	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.

11.1. MESSAGE ELEMENTS Detail of a standing order

PARAMETER	TYPE	MANDATORY	PURPOSE
transactionIdentification	Max35Text	Yes	Identifier of established standing order instruction

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.

11.2. MESSAGE ELEMENTS Standing order details – RESPONSE

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]	TUZEM	Max35Text	End to end

		[0..0] [0..0] [0..0]	SEPA EHP NONEHP		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 remittanceinfromation 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_CANCELLATI

					ON (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 reqularly 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.
++	openDate	[0..1]	ALL	ISODate	Date when the standing order was created.
++	lastDate	[0..1]	ALL	ISODate	Date of the last execution of the standing order.
+	exchangeRateInformation	[0..0]	ALL	The same object as to initiate payments chapter …	
+	chargeBearer	[0..0]	TUZEM	ChargeBearerType1	Fee payer.

		[0..0] [0..0] [0..1]	SEPA EHP NONEHP	Code	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] [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] [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 istructionStatus "ACWC". Not supported.

JSON příklad response:

```
{
   "standingOrderIdentification":    {
      "instructionIdentification": "NOTPROVIDED",
      "transactionIdentification": "33838828"
   },
   "amount": {"instructedAmount":    {
      "value": 1,
      "currency": "CZK"
   }},
   "requestedExecutionDate": "2021-10-05",
   "standingOrder":    {
      "alias": "TESTOVACI - AT",
      "execution":       {
         "mode": "UNTIL_DATE",
         "modeDue": "DUE_DAY_OF_MONTH",
         "interval": "MONTHLY"
      },
      "validity": {"lastExecutionDate": "2022-11-30"},
      "exceptions": {},
      "openDate": "2021-06-21",
      "lastDate": "2021-09-03"
   },
   "debtorAccount":    {
      "identification": {"iban": "CZ410100000115XXXXXXXXXX"},
      "currency": "CZK"
   },
   "creditorAccount":    {
      "identification": {"iban": "CZ080100000123XXXXXXXXXX"},
      "currency": "CZK"
   },
   "remittanceInformation":    {
      "unstructured": "TESTOVACI",
      "structured": {"creditorReferenceInformation": {"reference":       [
         "VS:111",
         "SS:333",
         "KS:222"
      ]}}
   },
   "paymentTypeInformation": {"serviceLevel": {"code": "DMCT"}},
   "instructionStatus": "ACSP",
   "signInfo":    {
      "state": "CLOSE",
      "signId": "33838828"
   }
}
```

12. Standing order status (GET /standingorders/{transactionIdentification})/status

Resource for getting status of the standing order.

Resource characteristics

URI:	/standingorders/{transactionIdentification}/status
HTTP Metoda:	GET
Request URL:	https://api.kb.cz/serverapi/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:	no
Třídění:	no
Filtrování:	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.
X-request-id	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
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	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.

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

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

12.1. MESSAGE ELEMENTS Status of entered/initiated standing order

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYP	PRESENTATION
+	instructionStatus	[1..1]	StatusCode	Status of entered standing

				order
+	realizedPayments	[0..0] conditional if it is an authorized standing order	Array	Array of transaction identificators. Not supported
++	transactionIdentification	[1..1]	Max35Text	Identifier of estabilished 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

HTTP STATUS CODE	*STATUSCODE*	*PURPOSE*
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACSP	[AcceptedSettlementInProcess] - All preceding checks such as technical validation and customer profile were successful,~~ and therefore the payment initiation has been accepted for execution~~ however it is not a confirmation of final processing on the part of debtor ASPSP´s side. This is an intermediate state that can result either in ACSC or RJCT. To obtain the status of entered standing order call (GET /my/payments/{paymentId}). **Usage**: this can be used by the first agent to report to the debtor that the standing order has been entered/initiated.
200	ACSC	[AcceptedSettlementCompleted] – Standing order on the debtor's account has been completed. **Warning**: this status is provided for transaction status reasons, not for financial information. For the financial information it can only be used after bilateral agreement.

13. Batch Payments – initiation (POST /batchpayments)

Resource for establishing a batch payment.

Additional information:

- The maximum number of orders in one batch is allowed 150.
- The daily number of payments in a batch payment is limited to 500.
- 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/serverapi/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*
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
Date	DateTime	Ano	Each transaction request includes the date, time, and when the message was created. In timestamp format.
X-request-id	Text	Ne	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
User-involved	Boolean	Ano	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
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'

13.1. 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

Example of JSON request:

```
{
"exchangeIdentification": " Mzdy202103abcd",
"instructionName": " Mzdy březen 2022.",
"payments": ["ZU00000FHK2","ZU00000FHK3"]
}
```

13.2. 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.

Example of JSON response:

```
{
  "batchDigest": "8680F5342F4A4E6B25BEF4E14B20CA1211427652",
  "instructionStatus": "ACTC",
  "signInfo":   {
    "signId": "001RE6ZN7E",
    "state": "OPEN"
  },
  "transactionIdentification": "001RE6ZN7E"
}
```

Status codes of payment – StatusCode

HTTP STATUS CODE	*STATUS CODE*	*PURPOSE*
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent

13.3. 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 iniciated by different applications]
400	NARR	[Daily limit of transactions in batches exhausted.]

14 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/serverapi/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*
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json format is primarily supported.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.
Date	DateTime	Ano	Each transaction request includes the date, time, and when the message was created. In timestamp format.
X-request-id	Text	Ne	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
User-involved	Boolean	Ano	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
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'

Path parameters of the request:

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

14.1. 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.

Example of JSON request:

```
{

"authorizationType":"USERAGENT_REDIRECT",

"redirectUrl":"https://www.kb.cz",

"batchDigest":"8680F5342F4A4E6B25BEF4E14B20CA1211427652"
}
```

14.2. Batch payment authorization – inititation 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.

Example of JSON response:

```
{
  "authorizationType": "USERAGENT_REDIRECT",
  "href": {"url": "https://testlogin.kb.cz:443/autfe/ssologin?SAMLRequest=rVZpk6rIEv0rhvOR6GZHNLp7gkUQVHZx%2BfKCpUBklQIBf%2F3D7r7b3Lk35k28r5lZ52Tmqag6L3%2F2eTa5gRomZfE6xZ%"},
  "method": "GET",
  "signInfo":   {
    "signId": "001RE6ZN6W",
    "state": "OPEN"
  }
}
```

14.3. 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]

15. Batch payment – status (GET /batchpayments/{transactionIdentification}/status)

Resource for getting status of the batch payment.

Resource characteristics

URI:	/batchpayments/{transactionIdentification}/status
HTTP Method:	GET
Request URL:	https://api.kb.cz/serverapi/pisp/v1/batchpayments/{transactionIdentification}/status
Authorization:	request requires **no authorization** of user/client as part of API calling
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:	no
Sorting:	no
Filtration:	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.
Date	DateTime	Yes	Each transaction request includes the date, time, and when the message was created. In timestamp format.
X-request-id	Text	No	Unique identification of each specific request. The value of this parameter should therefore be generated randomly and the individual x-request-id should not match each other within a short time interval from one request.
User-involved	Boolean	Yes	The false/true flag identifies whether the request was sent by the end user based on their activity, not by a client application without the knowledge of the logged in user.
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'

Path parameter of the request:

PARAMETER	*TYPE*	*MANDATORY*	*PURPOSE*
transactionIdentification	Text	Yes	Batch transaction identification.

15.1. 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.

Status codes of Batch payments – StatusCode

HTTP STATUS CODE	*STATUSCODE*	*PURPOSE*
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACSP	[AcceptedSettlementInProcess] - All preceding checks such as technical validation and customer profile were successful ~~and therefore the payment initiation has been accepted for execution~~, however it is not a confirmation of final processing on the part of debtor ASPSP´s side. This is an intermediate state that can result either in ACSC or RJCT. To obtain the status of entered payment call (GET /my/payments/{paymentId}). **Usage**: this can be used by the first agent to report to the debtor that the batch payment has been entered/initiated.
200	ACSC	[AcceptedSettlementCompleted] – Batch payment on the debtor's account has been completed. **Warning**: this status is provided for transaction status reasons, not for financial information. For the financial information it can only be used after bilateral agreement.

Example of JSON response:

```
{
   "batchDigest": "8680F5342F4A4E6B25BEF4E14B20CA1211427652",
   "instructionStatus": "ACSP",
   "payments":    [
            {
         "instructionIdentification": "NOTPROVIDED",
         "instructionStatus": "ACSP",
         "transactionIdentification": "ZU00000FHK3"
      },
            {
         "instructionIdentification": "NOTPROVIDED",
         "instructionStatus": "ACSP",
         "transactionIdentification": "ZU00000FHK2"
      }
   ],
   "signInfo":    {
      "signId": "001RE6ZN7E",
      "state": "CLOSE"
   },
   "transactionIdentification": "001RE6ZN7E"
}
```

15.2. 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]