

# Sandbox příručka pro službu Iniciování platby

## Change log

<b>Date</b>	<b>Version</b>	<b>Description</b>
26.02.2020	5	Aktualizace dokumentu a printscreenů.
21.05.2020	6	Rožšíření dokumentu o trvalé příkazy (platné od 27.5.2020)
03.06.2022	7	Rožšíření dokumentu o dávkové platby

## Obsah

1. Služba Iniclace platby (PIS) .....	4
2. Sandbox API pro volání služby Iniclace platby .....	4
3. Vydání certifikátu .....	4
4. Definice mocku .....	4
6. Postup pro vygenerování klíče/tokenu ke službě PISP .....	5
7. Postup přístupu do aplikace přes konzoli na API portálu .....	7
Volání mocku PIS „New payment“ na otestování .....	7
Volání mocku PIS „Payment status information“ k otestování .....	11
Volání mocku PIS „Balance check“ k otestování .....	16
Volání mocku PIS „New Standing Order initiation“ k otestování .....	18
Volání mocku PIS „Standing Orders Authorization - start“ k otestování .....	21
Volání mocku PIS „Standing Order Status“ k otestování .....	23
Volání mocku PIS „Standing Order Detail“ k otestování .....	26
Volání mocku PIS „New batch payment - instruction initiation“ k otestování .....	29
Volání mocku PIS „Batch Authorization – starting the particular authorization method“ k otestování .....	31
Volání mocku PIS „Batch payment status“ k otestování .....	33
8. Postup přístupu do aplikace přes přímé volání .....	35
Nová platba – I. iniciace platby (POST /my/payments) .....	35
Elementy zprávy Nová platba – iniciace platby .....	37
Elementy response Nová platba - iniciace platby .....	47
Status založené/iniciované platby (GET /payments/{paymentId}/status) .....	48
Elementy zprávy Status založené/iniciované platby .....	48
Krok II. Iniclace autorizace platby – specifické pro každou banku (POST /my/payments/{paymentId}/sign/{signId}) .....	50
Elementy zprávy Krok II. Iniclace autorizace platby - specifické pro každou banku .....	51
Dotaz na dostatek prostředků (POST /accounts/balanceCheck) .....	52
Elementy zprávy Dotaz na dostatek prostředků .....	53
Elementy response Dotaz na dostatek prostředků .....	53
Návratové kódy parametru „response“ – Code set .....	53
Nový trvalý příkaz – iniciace platby (POST/my/standingorders) .....	54
Elementy zprávy Nová platba – iniciace trvalého příkazu .....	55
Elementy response Nový trvalý příkaz – iniciace trvalého příkazu .....	59
Iniclace autorizace trvalé platby – specifické pro každou banku (POST /standingorders/{transactionIdentification}/sign/{signId}) .....	60
Elementy zprávy Iniclace autorizace trvalého příkazu .....	61
Parametry requestu: .....	61
Detail trvalého příkazu (GET /my/standingorders/{transactionIdentification}) .....	62
Elementy zprávy Detail trvalého příkazu .....	63
Elementy response zprávy Detail trvalého příkazu .....	63
Status založeného/iniciovaného trvalého příkazu (GET /my/standingorders/{transactionIdentification}/status) .....	67
Elementy zprávy Status založeného/iniciovaného trvalého příkazu .....	68
Dávková platba – iniciace (POST /my/batchpayments) .....	70
Elementy request Dávková platba – iniciace .....	70
Elementy response Dávková platba – iniciace .....	71
Chybové kódy definované pro službu POST iniciace dávkové platby .....	72
Iniclace autorizace dávkové platby (POST /my/ batchpayments/{transactionIdentification}/sign/{signId}) .....	72
Elementy request iniciace autorizace dávkové platby .....	73
Elementy response iniciace autorizace dávkové platby .....	74
Chybové kódy definované pro službu POST iniciace autorizace dávkové platby .....	74
Dávková platba – status (GET/my/batchpayments/{transactionIdentification}/status) .....	75
Elementy response Dávková platba – Status .....	76
Chybové kódy definované pro službu GET status dávkové platby .....	76

## 1. Služba Iniciační platby (PIS)

Přehled implementovaných zdrojů:

- iniciační platby –
  - o Krok I. Iniciační autorizované platby
- status založené/iniciační platby – umožňuje získat informace o stavu platby během jejího zpracování v KB
- autorizace platby
  - o Krok II. Iniciační autorizované platby – pomocí tohoto resource je iniciována autorizace platby
- Trvalý příkaz – iniciační platby
- Iniciační autorizované trvalé platby
- Status a detail trvalého příkazu
- Dávková platba – iniciační
- Iniciační autorizace dávkové platby
- Status dávkové platby

Nepovolené typy plateb pro prostředí Sandbox:

- Inkasa (povolení k inkasu, příkaz k inkasu)
- Instantní platby
- Platba šekem

Komerční banka vychází z jednotné struktury a formátu informací, která byla vydefinována Českou bankovní asociací v dokumentu [Czech Open Banking Standard](#).

Informace poskytované prostřednictvím Otevřeného bankovní API jsou lokalizovány do českého i anglického jazyka.

Povolená znaková sada vychází ze znakové sady SWIFT pro zahraniční/SEPA platby.

Během jednoho volání může být zaslán a zpracován vždy jen jeden dotaz.

## 2. Sandbox API pro volání služby Iniciační platby

Sandbox umožňuje třetím stranám vyzkoušení služby (mock) pro získávání níže uvedených informací, vztahujících se k platebnímu účtu klienta Komerční banky, a.s., pobočky zahraniční banky (dále jen Komerční banky).

Do sandboxu API mohou přistupovat jakékoliv subjekty, tedy ne jen třetí strany s licenci pro PSD2 služby. Nutnou podmínkou je registrace na API portálu KB <https://api.kb.cz/portal/?tenant=api.kb.cz>. Bez registrace není možné služby na sandboxu využívat. Postup registrace je popsán v dokumentu API Sandbox Registrace.

## 3. Vydání certifikátu

Certifikát je nezbytnou podmínkou jak pro produkční volání, tak pro sandbox PSD2. Pro využití na sandboxu budou třetím stranám po jejich registraci na Sandboxu poskytnuty certifikáty od Komerční banky **na základě žádosti zasláné na emailovou schránku [api@kb.cz](mailto:api@kb.cz)**. Certifikáty ze sandboxu nejsou určeny pro produkční využití. Na produkci jsou taková volání zamítána a monitorována. Na produkci se používají kvalifikované PSD2 certifikáty vydané kvalifikovanou certifikační autoritou dle EU QTSP seznamu na <https://webgate.ec.europa.eu/tl-browser/>

## 4. Definice mocku

Jsou definovány vzorové requesty pro statické volání, které není možné ve volání třetí stranou měnit. V případě změny bude request zamítnut. Dle použitých vzorových requestů jsou vráceny odpovídající response, případně error kódy.

### 1. Iniciační platby:

- Iniciační platby (TPL) – pro získání vstupních hodnot, tedy validní struktury iniciované platby je nutné kliknout na šedé pole **Example Value** nacházející se vpravo od pole paymentRequest. Sandbox automaticky těmito hodnotami vyplní pole pro zadání platby.
- Iniciační platby tuzemské platby - error na nevalidní IBAN 400 - AC02 - [InvalidDebtorAccountNumber], pole debtorAccount.identification.iban
- Iniciační platby tuzemské platby - error na nevalidní částku 400 - AM12 - [InvalidAmount], pole (amount.instructedAmount.value)
- Iniciační platby tuzemské platby - error na nevalidní datum splatnosti, datum splatnosti je v budoucnosti, 400 NARR - [Entered Execution date cannot exceed actual Business date], pole requestedExecutionDate
- Iniciační platby tuzemské platby - error na nevalidní formáty polí - vrácení kolekce errorů

### 2. Status platby:

- Status platby – jako vstupní hodnotu paymentID (transactionID) je nutné vyplnit **ZU000000LAP**
- Status platby - error na neexistenci paymentID (transactionID), pokud není ID nalezeno, je vygenerován error 404 \_NOT\_FOUND

## 5. Hlášení chyb

**Hlášení chyb Sandboxu nebo jeho volání probíhá vždy pomocí mailové schránky [api@kb.cz](mailto:api@kb.cz). Odeslaný mail musí obsahovat níže uvedené náležitosti v případě chybějící požadované informace nebude možné dotaz nebo chybu zpracovat.**

PSD2 API: CZ, SK

Prostředí: Sandbox, Produkce

Zda bylo voláno z FE Sandbox vč. typu a verze použitého prohlížeče nebo v případě BE volání název a verzi programu pro BE volání

Typ volání

Datum a čas uskutečněného volání

IP adresu

Chybu a její co nej přesnější popis, který může být doplněn o příslušný otisk obrazovky

**Bez výše uvedených hodnot není možné hlášenou chybou zabývat**

## 6. Postup pro vygenerování klíče/tokenu ke službě PISP

### Předpoklady pro vygenerování klíče/tokenu

Uživatel je řádně zaregistrován a přihlášen do portálu Sandbox.

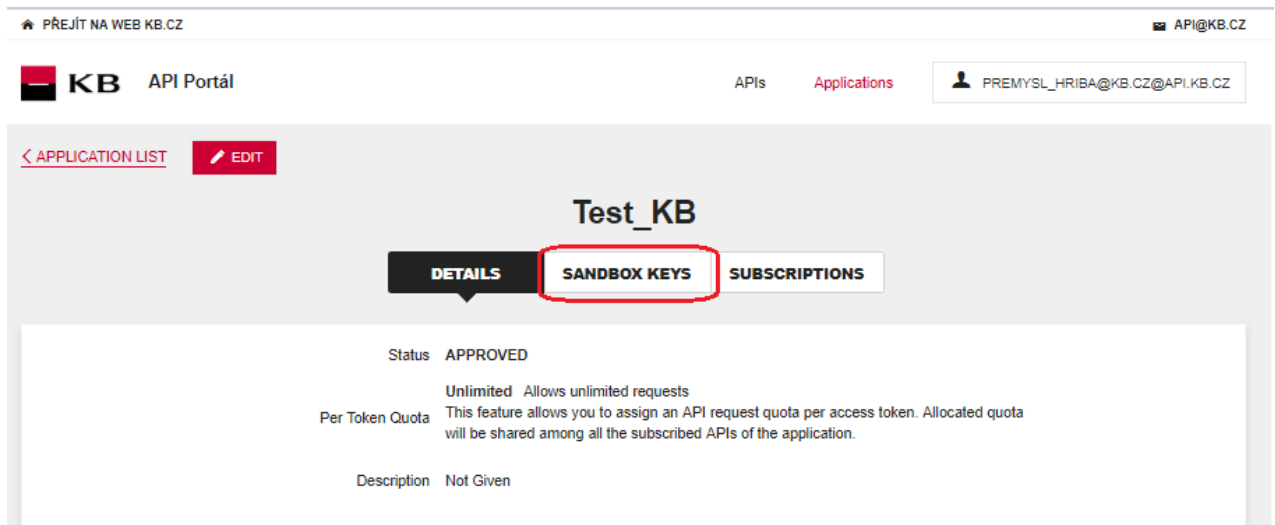
### Vstup do aplikačního menu a výběr požadované aplikace

Přihlášený uživatel vstoupí do aplikačního menu pomocí odkazu Applications v horní částky obrazovky

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

## Výběr funkčnosti aplikačního menu

Uživatel v aplikačním menu zvolí položky „SANBOX KEYS“



API Portal

APIs Applications

PREMYSL\_HRIBA@KB.CZ@API.KB.CZ

< APPLICATION LIST EDIT

### Test\_KB

DETAILS **SANDBOX KEYS** SUBSCRIPTIONS

Status APPROVED

Unlimited Allows unlimited requests

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

Description Not Given

## Generování certifikátu pro službu PISP

Uživatel v Sandboxu může zvolit vygenerování tokenu pro službu PISP za předpokladu, že je uživatel přihlášen k odběru služeb PISP.

Uživatel zvolí požadovaný scope pro generování klíče/tokenu a vygeneruje token pomocí funkčnosti „REGENERATE“

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

## Test\_KB

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

### Consumer Key

..... [COPY](#)

### Consumer Secret

..... [COPY](#)

### Grant Types

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

- |   |   |  |  |
|---|---|--|--|
| <input checked="" type="checkbox"/> Refresh Token | <input checked="" type="checkbox"/> SAML2             | <input checked="" type="checkbox"/> Implicit | <input checked="" type="checkbox"/> Password |
| <input checked="" type="checkbox"/> IWA-NTLM      | <input checked="" type="checkbox"/> Client Credential | <input checked="" type="checkbox"/> Code     |  |

### Callback URL

https://www.kb.cz

[UPDATE](#)

### Generating Access Tokens

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

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

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

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

### Generate a Test Access Token

#### Access Token

..... [COPY](#)

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

aisp : aisp.

**pisp : pisp.**

SELECT..

#### Validity period

3600

Seconds

[REGENERATE](#)

## 7. Postup přístupu do aplikace přes konzoli na API portálu

Volání mocku PIS „New payment“ na otestování

Uživatel vybere operaci, kterou si přeje otestovat. V tomto případě vybere operaci „New payment“. Tato operace umožňuje založit novou platbu. Po kliknutí na „SHOW MORE“ se rozbalí daná operace.

🏠 PŘEJÍT NA WEB KB.CZ ✉ API@KB.CZ

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

[< GO BACK](#)

## PISP-Sandbox

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

KB PISP Component API

API CONSOLE DOCUMENTATION

Try  Using  Key

Set Request Header

Authorization : Bearer

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

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

Method	Description	Action
<span style="background-color: red; color: white; padding: 2px 5px; border-radius: 3px; font-weight: bold;">POST</span>	New payment – payment initiation <code>/payments</code>	<a href="#">SHOW MORE</a> <span style="font-size: 0.8em;">▼</span>
<span style="background-color: red; color: white; padding: 2px 5px; border-radius: 3px; font-weight: bold;">POST</span>	Payment Authorization – starting the particular authorization method <code>/payments/{paymentId}/sign/{signId}</code>	<a href="#">SHOW MORE</a> <span style="font-size: 0.8em;">▼</span>
<span style="background-color: red; color: white; padding: 2px 5px; border-radius: 3px; font-weight: bold;">GET</span>	Payment status information <code>/payments/{paymentId}/status</code>	<a href="#">SHOW MORE</a> <span style="font-size: 0.8em;">▼</span>
<span style="background-color: red; color: white; padding: 2px 5px; border-radius: 3px; font-weight: bold;">POST</span>	Balance check <code>/payments/balanceCheck</code>	<a href="#">SHOW MORE</a> <span style="font-size: 0.8em;">▼</span>

**Applications**  **Tiers**

SUBSCRIBE



## Vyplnění požadovaných polí u operace „New payment“

Uživatel, který si přeje iniciovat novou platbu, vyplní všechna pole hodnotami v příslušném formátu. Pokud vše provedl, jak měl a neexistuje žádný jiný důvod proč platba nemohla proběhnout (např.: nedostatečný zůstatek na účtu), zobrazí se mu zpráva o provedení platby. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněná pole jsou rudě zvýrazněna. Pro „paymentRequest“ stačí kliknout na uvedený příklad a ten se zkopíruje do příslušného pole.

[PŘEJÍT NA WEB KB.CZ](#)
API@KB.CZ

---

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

---

[< GO BACK](#)

### PISP-Sandbox

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

KB PISP Component API

API CONSOLE
DOCUMENTATION

---

Try  Using  Key

Set Request Header  
 Authorization : Bearer

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

**PISP**
ShowHide | List Operations | Expand Operations

POST New payment – payment initiation /payments SHOW LESS ^

**Required Scopes**

Key	Name
pisp	pisp

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

Model **Example Value**

```

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

```

Response Content Type

**Parameters**

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

New payment request
body

Parameter content type:

Model **Example Value**

```

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

```

## Vypsání chyby u operace „New payment“

Pokud byla některá hodnota zadána špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.

### Response Messages

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


**TRY IT OUT**

## Volání mocku PIS „Payment status information“ k otestování

Uživatel vybere operaci, kterou si přeje otestovat. V tomto případě vybere operaci „Payment status information“. Tato operace umožňuje zobrazit stav platby. Jedná se o založenou platbu, která ještě nebyla klientem autorizována, nebo již byla autorizována a PISP se dotazuje na její stav (GET). Po kliknutí na „SHOW MORE“ se rozbalí daná operace.

[PŘEJÍT NA WEB KB.CZ](#)
API@KB.CZ

---


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

---

[< GO BACK](#)

### PISP-Sandbox

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

KB PISP Component API

API CONSOLE

DOCUMENTATION

Try TEST\_KB ▼ Using SANDBOX ▼ Key

Set Request Header

Authorization : Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.

Swagger ( /swagger.json )

#### PISP

Show/Hide | List Operations | Expand Operations

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

**Applications**

DEFAULTAPPLICATION ▼

**Tiers**

UNLIMITED ▼

SUBSCRIBE

## Vyplnění požadovaných polí u operace „Payment status information“

Uživatel, který si přeje zobrazit stav určité platby, vyplní všechna pole hodnotami v příslušném formátu. Pokud vše provedl, jak měl, zobrazí se mu informace o dané platbě. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněná pole jsou rudě zvýrazněna.

GET
Payment status information  
/payments/{paymentId}/status
SHOW LESS ^

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

Model | Example Value

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

Response Content Type  ▼

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
x-request-id	<input type="text"/>	External Request ID	header	string
TPP-Name	<input type="text"/>	Transaction initiator name	header	string
paymentId	<input type="text" value="(required)"/>	Unique bank transaction identification example: 'WU000024R2O'	path	string

## Vypsání chyby u operace „Payment status information“

Pokud byla některá hodnota zadaná špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.

## Response Messages

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

[TRY IT OUT](#)

## Volání mocku PIS „Payment Authorization“ k otestování

Uživatel vybere operaci, kterou si přeje otestovat. V tomto případě vybere operaci „Payment Authorization“. Tato operace umožňuje nastartovat konkrétní autorizační metody. Po kliknutí na „SHOW MORE“ se rozbalí daná operace.

🏠 PŘEJÍT NA WEB KB.CZ ✉ API@KB.CZ

---

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

---

[< GO BACK](#)

### PISP-Sandbox

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

KB PISP Component API

API CONSOLEDOCUMENTATION

Try  ▼ Using  ▼

Set Request Header

Authorization : Bearer

[Swagger \(/swaggerjson\)](#)

### PISP

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

<span style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">POST</span>	New payment – payment initiation <a href="#">/payments</a>	<a href="#">SHOW MORE</a> <span style="color: red;">▼</span>
<span style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">POST</span>	Payment Authorization – starting the particular authorization method <a href="#">/payments/{paymentId}/sign/{signId}</a>	<a href="#">SHOW MORE</a> <span style="color: red;">▼</span>
<span style="background-color: #6c757d; color: white; padding: 2px 5px; border-radius: 3px;">GET</span>	Payment status information <a href="#">/payments/{paymentId}/status</a>	<a href="#">SHOW MORE</a> <span style="color: red;">▼</span>
<span style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">POST</span>	Balance check <a href="#">/payments/balanceCheck</a>	<a href="#">SHOW MORE</a> <span style="color: red;">▼</span>

**Applications**  ▼ **Tiers**  ▼  SUBSCRIBE

## Vyplnění požadovaných polí u operace „Payment Authorization“

Uživatel, který si přeje nastartovat autorizační proces, vyplní všechna pole hodnotami v příslušném formátu. Pokud vše provedl, jak měl, zobrazí se mu přehled hodnot potřebných pro dokončení autorizace. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněná pole jsou rudě zvýrazněna. Pro pole „authInitiationRequest“ stačí kliknout na uvedený příklad a ten se zkopíruje do příslušného pole.

**Required Scopes**

Key	Name
piszp	piszp

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

Model | Example Value

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

Response Content Type:  ▼

**Parameters**

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

Parameter content type:  ▼

Model | Example Value

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

## Vypsání chyby u operace „Payment Authorization“

Pokud byla některá hodnota zadána špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.

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

[TRY IT OUT](#)

## Volání mocku PIS „Balance check“ k otestování

Uživatel vybere operaci, kterou si přeje otestovat. V tomto případě vybere operaci „Balance Check“. Tato operace umožňuje získat informaci o dostupnosti prostředků na účtu klienta.



[< GO BACK](#)

## PISP-Sandbox

Version: v1

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

 Status: PUBLISHED

KB PISP Component API

API CONSOLE
DOCUMENTATION

 Try TEST\_KB ▼ Using SANDBOX ▼ Key

Set Request Header

 Authorization : Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.
[Swagger \(/swagger.json\)](#)
[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

### PISP

<span style="border: 1px solid red; padding: 2px;">POST</span>	New payment – payment initiation <code>/payments</code>	SHOW MORE <span style="color: red;">▼</span>
<span style="border: 1px solid red; padding: 2px;">POST</span>	Payment Authorization – starting the particular authorization method <code>/payments/{paymentId}/sign/{signId}</code>	SHOW MORE <span style="color: red;">▼</span>
<span style="border: 1px solid red; padding: 2px;">GET</span>	Payment status information <code>/payments/{paymentId}/status</code>	SHOW MORE <span style="color: red;">▼</span>
<span style="border: 2px solid red; padding: 2px;">POST</span>	Balance check <code>/payments/balanceCheck</code>	SHOW MORE <span style="color: red;">▼</span>

Applications

DEFAULTAPPLICATION ▼

Tiers

UNLIMITED ▼
SUBSCRIBE

### Vyplnění požadovaných polí u operace „Check Balance“


Pro requestBody stačí kliknout na uvedený příklad a ten se zkopíruje do příslušného pole. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněné pole je červeně zvýrazněno

**Response Class (Status 200)**

successful operation

Model | Example Value

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

Response Content Type  

**Parameters**


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

balanceCheckRequest



Model | Example Value

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

Parameter content type:  

**Volání mocku PIS „New Standing Order initiation“ k otestování**

Tato operace umožňuje založit nový trvalý příkaz Po kliknutí na „SHOW MORE“ se rozbalí daná operace.

[← GO BACK](#)

## PISP-Sandbox

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

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try  Using  Key

Set Request Header

Authorization : Bearer  Swagger ( /swagger.json )

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

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

### Vyplnění požadovaných polí u operace „New Standing Order initiation“

Pro requestBody stačí kliknout na uvedený příklad a ten se zkopíruje do příslušného pole. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněné pole je červeně zvýrazněno.

### Response Class (Status 200)

OK

Model | Example Value

```
{
  "standingOrderIdentification": {
    "instructionIdentification": 12312425,
    "transactionIdentification": 358745
  },
  "amount": {
    "instructedAmount": {
      "value": 10.00,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2022-02-19",
}
```

Response Content Type  ▼

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Payload	<input type="text" value="(required)"/>	Request Body	body	<div style="border: 2px solid red; padding: 5px;"> <p>Model   Example Value</p> <pre>{   "standingOrderIdentification": {     "instructionIdentification": 12312425   },   "amount": {     "instructedAmount": {       "value": 10.00,       "currency": "CZK"     }   }, }</pre> </div>
	Parameter content type: <input type="text" value="application/json"/> ▼			
x-request-id	<input type="text"/>	External Request ID	header	string
TPP-Name	<input type="text" value="(required)"/>	Transaction Initiator name	header	string

### Vypsání chyby u operace "New Standing Order initiation"

Pokud byla některá hodnota zadána špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.

## Response Messages

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

**TRY IT OUT**

## Volání mocku PIS „Standing Orders Authorization - start“ k otestování

Uživatel vybere operaci, kterou si přeje otestovat. V tomto případě vybere operaci „Standing Orders Authorization - start“. Tato operace umožňuje nastartovat konkrétní autorizační metody. Po kliknutí na „SHOW MORE“ se rozbalí daná operace.

[← GO BACK](#)

## PISP-Sandbox

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

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try  Using  Key

Set Request Header

Authorization : Bearer  Swagger ( /swagger.json )

### PISP

Show/Hide | List Operations | Expand Operations

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

### Vyplnění požadovaných polí u operace „Standing Orders Authorization - start“

Pro requestBody stačí kliknout na uvedený příklad a ten se zkopíruje do příslušného pole. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněné pole je červeně zvýrazněno.

Parameter	Value	Description	Parameter Type	Data Type
transactionIdentification	<input type="text" value="(required)"/>	Unique bank transaction identification example: '358745'	path	string
signId	<input type="text" value="(required)"/>	Identifier of the change	path	string
authInitiationRequest	<input type="text" value="(required)"/>  Parameter content type: <input type="text" value="application/json"/>	Initiation request - identifier of the authorization method and back URL	body	<div style="border: 2px solid red; padding: 5px; display: inline-block;"> <b>Model Example Value</b>  <pre>{   "authorizationType": "USERAGENT_REDIRECT",   "redirectUrl": "string" }</pre> </div>
x-request-id	<input type="text"/>	External Request Id	header	string
TPP-Name	<input type="text" value="(required)"/>	Transaction Initiator name	header	string

## Vypsání chyby u operace „Standing Orders Authorization - start“

Pokud byla některá hodnota zadána špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.

### Response Messages

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

## Volání mocku PIS „Standing Order Status“ k otestování

Uživatel vybere operaci, kterou si přeje otestovat. V tomto případě vybere operaci „Standing Order Status“. Tato operace umožňuje zobrazit informace o stavu trvalém příkazu.

## PISP-Sandbox

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

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try PREMA\_TEST\_KB ▼ Using SANDBOX ▼ Key

Set Request Header

Authorization : Bearer
eyJ0eXAIOiJKV1QiLCJhbGciOiJSUzI1NiJ9.

Swagger ( /swagger.json )

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

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

### Vyplnění požadovaných polí u operace „Standing Order Status“

Uživatel, který si přeje zobrazit stav určité platby, vyplní všechna pole hodnotami v příslušném formátu. Pokud vše provedl, jak měl, zobrazí se mu informace o stavu trvalého příkazu. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněná pole jsou rudě zvýrazněna.



### Response Class (Status 200)

OK

Model | Example Value

```

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

```

Response Content Type  

#### Parameters

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

### Response Class (Status 200)


successful operation

Model | Example Value

```

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

```

Response Content Type  

#### Parameters

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

### Vypsání chyby u operace „Standing Order Status“

Pokud byla některá hodnota zadaná špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Input parameter is invalid	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_400",       "scope": "x-request-id",       "message": "Value of parameter x-request-id is wrong"     }   ] }</pre>	
401	Missing certificate or access token	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>	
403	Invalid certificate or token	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>	
404	Id does not exist	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ID_NOT_FOUND",       "message": "Parameter paymentsId not found"     }   ] }</pre>	
415	Invalid message charset	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR18",       "message": "InvalidCharacterSet"     }   ] }</pre>	
500	Unexpected error occurred	Model   Example Value	
		<pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>	

**TRY IT OUT** Hide Response

## Volání mocku PIS „Standing Order Detail“ k otestování

Uživatel vybere operaci, kterou si přeje otestovat. V tomto případě vybere operaci „Standing Order Detail“. Tato operace umožňuje zobrazit podrobnosti o trvalém příkazu.

## PISP-Sandbox

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

KB PISP Component API

API CONSOLE
DOCUMENTATION

Try PREMA\_TEST\_KB ▼ Using SANDBOX ▼ Key

Set Request Header

Authorization : Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiU9.

Swagger ( /swagger.json )

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

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

### Vyplnění požadovaných polí u operace „Standing Order Detail“

Uživatel, který si přeje zobrazit stav určité platby, vyplní všechna pole hodnotami v příslušném formátu. Pokud vše provedl, jak měl, zobrazí se mu podrobnosti o trvalém příkazu. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněná pole jsou rudě zvýrazněna.

### Response Class (Status 200)

OK

Model | Example Value

```

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

```

Response Content Type  

### Parameters

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

### Vypsání chyby u operace „Standing Order Status“

Pokud byla některá hodnota zadána špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.

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

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

## Volání mocku PIS „New batch payment - instruction initiation“ k otestování

Tato operace umožňuje založit dávkový příkaz. Po kliknutí na „SHOW MORE“ se rozbalí daná operace.

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

## Vyplnění požadovaných polí u operace „New batch payment - instruction initiation“

Pro requestBody stačí kliknout na uvedený příklad a ten se zkopíruje do příslušného pole. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněné pole je červeně zvýrazněno.

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

Response Content Type  


#### Parameters

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

paymentRequest

New batch payment request

body

Parameter content type:  

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



### Vypsání chyby u operace „New batch payment - instruction initiation“

Pokud byla některá hodnota zadána špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.

## Response Messages

HTTP Status Code	Reason	Response Model	Headers				
400	Input parameter is invalid	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "ERR_CODE_400",       "scope": "x-request-id",       "message": "Value of parameter x-request-id is wrong"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "ERR_CODE_400",       "scope": "x-request-id",       "message": "Value of parameter x-request-id is wrong"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "ERR_CODE_400",       "scope": "x-request-id",       "message": "Value of parameter x-request-id is wrong"     }   ] }</pre>						
401	Missing certificate or access token	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "ERR_CODE_401",       "message": "Missing certificate or access token"     }   ] }</pre>						
403	Invalid certificate or token	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "ERR_CODE_403",       "message": "Invalid certificate or token"     }   ] }</pre>						
415	Invalid message charset	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "RR10",       "message": "InvalidCharacterSet"     }   ] }</pre>						
500	Unexpected error occurred	<table border="1"> <thead> <tr> <th>Model</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td></td> <td> <pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre> </td> </tr> </tbody> </table>	Model	Example Value		<pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>	
Model	Example Value						
	<pre>{   "errors": [     {       "error": "ERR_CODE_500",       "message": "Internal Server Error"     }   ] }</pre>						

## Volání mocku PIS „Batch Authorization – starting the particular authorization method“ k otestování

Uživatel vybere operaci, kterou si přeje otestovat. V tomto případě vybere operaci „Batch Authorization – starting the particular authorization method“ Tato operace umožňuje nastartovat konkrétní autorizační metody. Po kliknutí na „SHOW MORE“ se rozbalí daná operace.

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

## Vyplnění požadovaných polí u operace „Batch Authorization – starting the particular authorization method“

Pro requestBody stačí kliknout na uvedený příklad a ten se zkopíruje do příslušného pole. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněné pole je červeně zvýrazněno.

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

Parameter content type:

Model	Example Value
	<pre>{   "authorizationType": "USERAGENT_REDIRECT",   "redirectUrl": "http://kb.cz",   "batchDigest": "AAF21A859D225FD1D1889B2F08DA3E4A9AD6F7CC" }</pre>

### Vypsání chyby u operace „Batch Authorization – starting the particular authorization method“

Pokud byla některá hodnota zadána špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.



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

### Volání mocku PIS „Batch payment status“ k otestování

Uživatel vybere operaci, kterou si přeje otestovat. V tomto případě vybere operaci „Standing Order Detail“. Tato operace umožňuje zobrazit podrobnosti o trvalém příkazu.

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

### Vyplnění požadovaných polí u operace „Batch payment status“

Uživatel, který si přeje zobrazit stav určité platby, vyplní všechna pole hodnotami v příslušném formátu. Pokud vše provedl, jak měl, zobrazí se mu podrobnosti o dávkové platbě. V případě, že nejsou vyplněna všechna povinná pole, výpis se neprovede a nevyplněná pole jsou rudě zvýrazněna.

Response Content Type  ▾

#### Parameters

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

### Vypsání chyby u operace „Batch payment status“

Pokud byla některá hodnota zadaná špatně, vypíše se po zmáčknutí tlačítka „TRY IT OUT“ jedna z následujících chybových hlášek, popřípadě error specifikovaný v definici mocku, jinak proběhne výpis výsledku.

## Response Messages

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

## 8. Postup přístupu do aplikace přes přímé volání

### Nová platba – I. iniciace platby (POST /my/payments)

Resource pro založení nové platby.

#### Charakteristika resource

**URI:** /payments  
**HTTP Metoda:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/payments>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu.

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

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>API-key</b>	Text	Ne	Volitelný řetězec vydaný komunikující třetí straně jako identifikátor volání této strany primárně sloužící jako konfigurační prvek komunikace.

<b>Authorization</b>	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.
----------------------	------	-----	---

Parametry hlavičky response:

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

Obsah POST requestu a response volání viz kapitola 0 ELEMENTY ZPRÁVY Nová platba – iniciace platby

Tabulka - Chybové kódy definované ČBA standardem pro službu POST iniciace platby

HTTP STATUS KÓD	ERROR KÓD	ÚČEL
401	UNAUTHORISED	Chybějící certifikát.
403	FORBIDDEN	Volání metody, která neodpovídá licenci, nebo neplatný certifikát.
400	FIELD_MISSING	Chybějící povinné pole v requestu.
400	FIELD_INVALID	Hodnota pole není validní.
400	AC02	[InvalidDebtorAccountNumber] – nevalidní identifikátoru účtu v obsahu požadavku.
400	AC03	[InvalidCreditorAccountNumber] - číslo účtu příjemce je uzavřeno, blokováno, příp. pro daný typ účtu jsou zakázané kreditní transakce nebo je číslo účtu příjemce uvedeno v nevalidním formátu (poznámka: validováno pouze pro in-house platby).
400	AC10	[InvalidDebtorAccountCurrency] – uvedená měna účtu plátce neodpovídá měně účtu klienta pro dané číslo účtu vedené v bance (měna účtu je nepovinná, v případě multicurrency účtů je měnu účtu nutné specifikovat - Raiffeisenbank).
400	AC12	[InvalidAccountType] - typ účtu neodpovídá povoleným typům účtů (např. se jedná o neplatební účet).
403	AG01	[TransactionForbidden] – neexistující souhlas s přístupem k informaci o dostatku prostředků na účtu.
400	AM05	[Duplication] - jedná se o duplicitu. Univerzální kód pro duplicitní dotaz CISP (validuje například ČSOB) nebo se jedná o duplicitní platbu přes PISP (neunikátní reference platby).
400	AM11	[InvalidTransactionCurrency] – v požadavku je uvedena neobchodovaná/nepodporovaná měna.
400	AM12	[InvalidAmount] – chybně zadaná částka. Např. příliš nízká, nebo vysoká částka, či špatný formát čísla včetně počtu desetinných míst dle ISO 4217.
400	FF01	[Invalid File Format] – nevalidní JSON formát, či jiný technický problém se zpracováním dotazu.
400	BE19	[InvalidChargeBearerCode] - neplatný typ poplatku pro daný typ transakce.
400	DT01	[InvalidDate] - "Datum neodpovídá" – viz. Níže *

<b>400, 50x</b>	NARR	Narrative – obecný důvod pro odmítnutí platby, s doplněním informace o chybě.
<b>400</b>	RC07	[InvalidCreditorBICIdentifier] – neplatný SWIFT / BIC kód banky příjemce.
<b>400</b>	RC10	[InvalidCreditorClearingSystemMemberIdentifier] - neplatná identifikace kódu banky příjemce.
<b>400</b>	RR03	[MissingCreditorNameOrAddress] - v poli chybí požadovaná data o názvu či adrese (nebo její části) příjemce. Pokud jsou údaje ve špatném formátu, je využitý error code FIELD_INVALID
<b>400</b>	RR10	[InvalidCharacterSet] – nevalidní znaková sada v požadavku.

### Elementy zprávy Nová platba – iniciace platby

#### Uvažované typy plateb

KÓD PLATBY	SERVICELEVELCODE	POPIS
<b>TUZEM</b>	DMCT	Domácí platba
<b>SEPA</b>	ESCT	SEPA platba
<b>ZPL</b>	XBCT	Zahraniční platba v rámci EHP, Zahraniční platba mimo EHP

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP PLATBY	TYP FORMÁTU	PREZENTACE
+	paymentIdentification	[1..1]	ALL	PaymentIdentification1	Identifikace platby
++	instructionIdentification	[1..1]	ALL	Max35Text	Identifikace instrukce
++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA ZPL	Max35Text	Identifikace End To End
++	transactionIdentification	[0..0]	ALL	Max35Text	Identifikace transakce
+	paymentTypeInformation	[0..1]	ALL	PaymentTypeInformation19	Informace o typu platby
++	instructionPriority	[0..1]	ALL	Priority2Code	Priorita instrukce
++	serviceLevel	[0..0]	ALL	ServiceLevel8CZ	Úroveň služeb
+++	code	[0..0]	ALL	ExternalServiceLevel1Code	Kód úrovně služeb
++	categoryPurpose	[0..0]	ALL	CategoryPurpose1Choice	Kategorie účelu platby
+++	code	[0..0]	ALL	ExternalCategoryPurpose1Code	Kód kategorie účelu platby
+++	proprietary	[0..0]	ALL	Max35Text	Volný formát kategorie účelu platby
+	amount	[1..1]	ALL	TUZEM- AmountType3CZ SEPA- AmountType3CZ EHP- AmountType3Choice NONEHP- AmountType3Choice	Částka
++	instructedAmount	[1..1]	ALL	CurrencyAndAmount	Částka a měna v instrukci
+++	value	[1..1]	ALL	Amount	Částka převodu
+++	currency	[1..1]	ALL	CurrencyCode	Měna převodu
++	equivalentAmount	[0..0]	ALL	CurrencyAndAmount	Ekvivalentní částka a měna
+++	value	[0..0]	ALL	Amount	Ekvivalentní částka

					transakce
+++	currency	[0..0]	ALL	CurrencyCode	Měna ekvivalentní částky transakce
+	requestedExecutionDate	[0..1]	ALL	ISODate	Požadované datum provedení platby
+	exchangeRateInformation	[0..0]	ALL	ExchangeRateInformation1	Smluvní kurz
++	exchangeRate	[0..0]	ALL	BaseOneRate	Dohodnutý směnný kurz
++	rateType	[0..0]	ALL	ExchangeRateType1Code	Typ dohodnutého směnného kurzu
++	contractIdentification	[0..0]	ALL	Max35Text	Identifikátor použití dohodnutého směnného kurzu
+	chargeBearer	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	ChargeBearerType1Code	Plátce poplatků
+	chargesAccount	[0..0]	ALL	CashAccount16CZ	Účet pro poplatky
++	identification	[0..0]	ALL	AccountIdentification4ChoiceCZ	Identifikace čísla účtu pro poplatky
+++	iban	[0..0]	ALL	IBAN2007Identifier	Číslo účtu pro poplatky ve formátu IBAN
++	currency	[0..0]	ALL	CurrencyCode ISO 4217	Měna účtu pro poplatky
+	ultimateDebtor	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PartyIdentification32CZ1	Původní plátce
++	name	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max70Text	Jméno původního plátce
++	postalAddress	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PostalAddress6CZ	Poštovní adresa původního plátce
+++	streetName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max70Text	Ulice původního plátce
+++	buildingNumber	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max16Text	Číslo budovy původního plátce
+++	postCode	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max16Text	PSC původního plátce
+++	townName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Město původního plátce

+++	country	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	CountryCode ISO3166	Země původního plátce
+++	addressLine	[0..0] [0..2] [0..0]	TUZEM SEPA ZPL	Max70Text	Nestrukturovaný zápis adresy původního plátce
++	identification	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Party6Choice	Identifikace původního plátce
+++	organisationIdentification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	OrganisationIdentification4CZ	Jednoznačná identifikace původního plátce jako organizace/právnícké osoby. buď organisationIdentification nebo privateIdentification
++++	bicOrBei	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	BICIdentifier	Identifikace původního plátce jako organizace/právnícké osoby ve formě BIC nebo BEI kódu.
++++	other	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	GenericOrganisationIdentification1	Jiná identifikace původního plátce jako organizace/právnícké osoby.
+++++	identification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Jiná identifikace původního plátce jako organizace/právnícké osoby v nestrukturované podobě.
+++++	schemeName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	OrganisationIdentificationSchemeName1CZ	Typ dokumentu pro identifikaci původního plátce jako organizace/právnícké osoby.
+++++	proprietary	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Typ dokumentu pro identifikaci původního plátce jako organizace/právnícké osoby ve volném textovém formátu.
+++++	issuer	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Výstavce dokumentu pro identifikaci původního plátce jako organizace/právnícké osoby.
+++	privateIdentification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	PersonIdentification5CZ	Jednoznačná identifikace původního plátce jako fyzické osoby. buď organisationIdentification nebo privateIdentification

++++	other	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	GenericPersonIdentification1	Jiná identifikace původního plátce jako fyzické osoby v nestrukturované podobě.
+++++	identification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Jiná identifikace původního plátce jako fyzické osoby v nestrukturované podobě.
+++++	schemeName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PersonIdentificationSchemeName1Choice	Typ dokumentu pro identifikaci původního plátce jako fyzické osoby.
+++++	proprietary	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Typ dokumentu pro identifikaci původního plátce jako fyzické osoby ve volném textovém formátu.
+++++	issuer	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Výstavce dokumentu pro identifikaci původního plátce jako fyzické osoby.
+	debtor	[0..0]	ALL	PartyIdentification32CZ2	Plátce
++	name	[0..0]	ALL	Max70Text	Jméno plátce
++	postalAddress	[0..0]	ALL	PostalAddress6CZ	Poštovní adresa plátce
+++	streetName	[0..0]	ALL	Max70Text	Název ulice použitý pro poštovní adresu plátce.
+++	buildingNumber	[0..0]	ALL	Max16Text	Číslo popisné použité pro poštovní adresu plátce.
+++	postCode	[0..0]	ALL	Max16Text	Poštovní směrovací číslo použité pro poštovní adresu plátce.
+++	townName	[0..0]	ALL	Max35Text	Název města použitý pro poštovní adresu plátce.
+++	country	[0..0]	ALL	CountryCode ISO3166	Název země použitý pro poštovní adresu plátce.
+++	addressLine	[0..0]	ALL	Max70Text	Nestrukturovaný záznam poštovní adresy plátce.
+	debtorAccount	[1..1]	ALL	CashAccount16CZ	Účet plátce
++	identification	[1..1]	ALL	AccountIdentification4ChoiceCZ	Identifikace účtu plátce
+++	iban	[1..1]	ALL	IBAN2007Identifier	Číslo účtu plátce ve formátu IBAN
+++	other	[0..0]	ALL	GenericAccountIdentification1CZ	Jiný formát čísla účtu plátce
++++	identification	[0..0]	ALL	Max34Text	Číslo účtu plátce v lokálním formátu BBAN
++	currency	[0..1]	ALL	CurrencyCode ISO 4217	Měna účtu plátce
+	intermediaryAgent1	[0..0]	ALL	BranchAndFinancialInstitutionIdentification4CZ	Mezibanka 1
++	financialInstitutionIdentification	[0..0]	ALL	FinancialInstitutionIdentification7CZ	Identifikace finanční instituce
+++	bic	[0..0]	ALL	BICIdentifier	BIC / SWIFT kód banky
+++	clearingSystemMemberIdentification	[0..0]	ALL	ClearingSystemMemberIdentification2	Identifikace účastníka clearingového systému
++++	clearingSystemIdentification	[0..0]	ALL	ClearingSystemIdentification2Choice	Identifikace clearingového systému



+++++	code	[0..0]	ALL	ClearingSystemIdentification1Code	Kód
+++++	proprietary	[0..0]	ALL	Max35Text	Volný formát
++++	memberIdentification	[0..0]	ALL	Max35Text	Clearingový kód účastníka
+++	name	[0..0]	ALL	Max70Text	Jméno
+++	postalAddress	[0..0]	ALL	PostalAddress6CZ	Poštovní adresa
++++	streetName	[0..0]	ALL	Max70Text	Ulice
++++	buildingNumber	[0..0]	ALL	Max16Text	Číslo budovy
++++	postCode	[0..0]	ALL	Max16Text	PSČ
++++	townName	[0..0]	ALL	Max35Text	Město
++++	country	[0..0]	ALL	CountryCode ISO3166	Země
++++	addressLine	[0..0]	ALL	Max70Text	Nestrukturovaný zápis adresy
+++	other	[0..0]	ALL	GenericFinancialIdentification1CZ	Jiná identifikace banky
++++	identification	[0..0]	ALL	Max35Text	Lokální kód banky
+	creditorAgent	[0..0] [0..1] [1..1]	TUZEM SEPA ZPL	BranchAndFinancialInstitutionIdentification4CZ	Banka příjemce
++	financialinstitutionidentification	[0..0] [0..1] [1..1]	TUZEM SEPA ZPL	FinancialInstitutionIdentification7CZ	Identifikace finanční instituce
+++	bic	[0..0] [1..1] [0..1]	TUZEM SEPA ZPL	BICIdentifier	BIC / SWIFT kód banky
+++	clearingSystemMemberIdentification	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	ClearingSystemMemberIdentification2	Identifikace účastníka clearingového systému
+++++	clearingSystemIdentification	[0..0] [0..0] [0..1] ... buď anebo některá z jiných variant identifikace banky příjemce	TUZEM SEPA ZPL	ClearingSystemIdentification2Choice	Identifikace clearingového systému
+++++	code	[0..0] [0..0] [1..1]... buď code anebo proprietary	TUZEM SEPA ZPL	ExternalClearingSystemIdentification1Code	Kód
+++++	proprietary	[0..0] [0..0] [1..1]... buď code anebo proprietary	TUZEM SEPA ZPL	Max35Text	Volný formát

++++	memberIdentification	[0..0] [0..0] [1..1]	TUZEM SEPA ZPL	Max35Text	Clearingový kód účastníka
+++	name	[0..0] [0..0] [0..1]... buď anebo některá z jiných variant identifika ce banky příjemce	TUZEM SEPA ZPL	Max70Text	Jméno
+++	postalAddress	[0..0] [0..0] [0..1]... buď anebo některá z jiných variant identifika ce banky příjemce	TUZEM SEPA ZPL	PostalAddress6CZ	Poštovní adresa
++++	streetName	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max70Text	Ulice
++++	buildingNumber	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max16Text	Číslo budovy
++++	postCode	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max16Text	PSC
++++	townName	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max35Text	Město
++++	country	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	CountryCode ISO3166	Země
++++	addressLine	[0..0] [0..0] [0..2]	TUZEM SEPA ZPL	Max70Text	Nestrukturovaný zápis adresy
+++	other	[0..0] [0..0] [0..1]... buď anebo některá z jiných	TUZEM SEPA ZPL	GenericFinancialIdentification1C	Jiná identifikace banky

		variant identifika ce banky příjemce			
++++	identification	[0..0] [0..0] [0..1]	TUZEM SEPA ZPL	Max35Text	Lokální kód banky
+	creditor	[0..0] [1..1] [1..1]	TUZEM SEPA ZPL	PartyIdentification32CZ2	Příjemce
++	name	[0..0] [1..1] [1..1]	TUZEM SEPA ZPL	Max70Text	Jméno příjemce
++	postalAddress	[0..0] [0..1] [1..1]	TUZEM SEPA ZPL	PostalAddress6CZ	Poštovní adresa
+++	streetName	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	Max70Text	Ulice
+++	buildingNumber	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	Max16Text	Číslo budovy
+++	postCode	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	Max16Text	PSC
+++	townName	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	Max35Text	Město
+++	country	[0..0] [0..1] [0..1]	TUZEM SEPA ZPL	CountryCode ISO3166	Země
+++	addressLine	[0..0] [0..2] [0..2]	TUZEM SEPA ZPL	Max70Text	Nestrukturovaný zápis adresy
+	creditorAccount	[1..1]	ALL	CashAccount16CZ	Účet příjemce
++	identification	[1..1]	ALL	AccountIdentification4Ch oiceCZ	Identifikace účtu příjemce
+++	iban	[1..1]	ALL	IBAN2007Identifier	Číslo účtu ve formátu IBAN

+++	other	[0..0] [0..0] [1..1]	TUZEM SEPA ZPL	GenericAccountIdentificat ion1CZ:	Jiný formát čísla účtu
++++	identification	[0..0] [0..0] [1..1]	TUZEM SEPA ZPL	Max34Text	Číslo účtu v lokálním formátu BBAN
++	currency	[0..1] [0..0] [0..0]	TUZEM SEPA ZPL	CurrencyCode ISO4217	Měna účtu příjemce
+	ultimateCreditor	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PartyIdentification32CZ1	Konečný příjemce
++	name	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max70Text	Jméno
++	postalAddress	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PostalAddress6CZ	Poštovní adresa
+++	streetName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max70Text	Ulice
+++	buildingNumber	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max16Text	Číslo budovy
+++	postCode	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max16Text	PSC
+++	townName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Město
+++	country	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	CountryCode ISO3166	Země
+++	addressLine	[0..0] [0..2] [0..0]	TUZEM SEPA ZPL	Max70Text	Nestrukturovaný zápis adresy

++	identification	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Party6Choice	Identifikace konečného příjemce
+++	organisationIdentification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	OrganisationIdentification4CZ	Jednoznačná identifikace konečného příjemce jako organizace/právnícké osoby. buď organisationIdentification nebo privateIdentification
++++	bicOrBei	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	BICIdentifier	Identifikace konečného příjemce jako organizace/právnícké osoby ve formě BIC nebo BEI kódu.
++++	other	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	GenericOrganisationIdentification1	Jiná identifikace konečného příjemce jako organizace/právnícké osoby.
+++++	identification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Jiná identifikace konečného příjemce jako organizace/právnícké osoby v nestrukturované podobě.
+++++	schemeName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	OrganisationIdentificationSchemeName1CZ	Typ dokumentu pro identifikaci konečného příjemce jako organizace/právnícké osoby.
+++++	proprietary	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Typ dokumentu pro identifikaci konečného příjemce jako organizace/právnícké osoby ve volném textovém formátu.
+++++	issuer	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Výstavce dokumentu pro identifikaci konečného příjemce jako organizace/právnícké osoby.
+++	privateIdentification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	PersonIdentification5CZ	Jednoznačná identifikace konečného příjemce jako fyzické osoby. buď organisationIdentification nebo privateIdentification
++++	other	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	GenericPersonIdentification1	Jiná identifikace konečného příjemce jako fyzické osoby v nestrukturované podobě.
+++++	identification	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Jiná identifikace konečného příjemce jako fyzické osoby v nestrukturované podobě.

+++++	schemeName	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	PersonIdentificationSchemeName1Choice	Typ dokumentu pro identifikaci konečného příjemce jako fyzické osoby.
+++++	proprietary	[0..0] [1..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Typ dokumentu pro identifikaci konečného příjemce jako fyzické osoby ve volném textovém formátu.
+++++	issuer	[0..0] [0..1] [0..0]	TUZEM SEPA ZPL	Max35Text	Výstavce dokumentu pro identifikaci konečného příjemce jako fyzické osoby.
+	purpose	[0..0]	ALL	Purpose2Choice	Účel platby
++	code	[0..0]	ALL	ExternalPurpose1Code	Kód účelu platby
++	proprietary	[0..0]	ALL	Max35Text	Volný formát účelu platby
+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instrukce pro další banku
+	remittanceInformation	[0..1]	ALL	RemittanceInformation5CZ	Informace o platbě
++	unstructured	[0..1]	ALL	Max140Text, alfanumerických znaků podporovaných CERTIS (clearing ČNB), a to včetně podporovaných speciálních znaků	Nestrukturovaná zpráva pro příjemce (viz. Níže *)
++	structured	[0..1] [0..0] [0..0]	TUZEM SEPA ZPL	StructuredRemittanceInformation7CZ	Strukturovaná zpráva pro příjemce – variabilní, specifický a konstantní symbol
+++	creditorReferenceInformation	[0..1] [0..0] [0..0]	TUZEM SEPA ZPL	CreditorReferenceInformation2CZ	Informace o referenci příjemce
++++	reference	[0..3] [0..0] [0..0]	TUZEM SEPA ZPL	CreditorReferenceInformation2CZ	Hodnota VS, SS, KS

\* V případě, že poli **remittanceInformation.structured.creditorReferenceInformation.reference** bude obsahovat variabilní, konstantní nebo specifický symbol, dojde k jejich rozpoznání a samostatnému uložení do příslušných polí pro tyto symboly v modelu PISP.

- Hodnota **variabilního symbolu** se zaznamenává jako VS:max.10 číslic (např. VS:3451859072).
- Hodnota **konstantního symbolu** se zaznamenává jako KS:max.10 znaků (např. KS:0308).
- Hodnota **specifického symbolu** se zaznamenává jako SS:max.10 číslic (např. SS:8451201274).

JSON příklad elementu:

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

pozn. k poli *remittanceInformation.unstructured*: dle standardu bankovní asociace může i toto pole obsahovat informace o VS, KS a SS, nicméně my budeme s veškerými informacemi zde obsaženými pracovat tak, jako by se jednalo o prostý popis platby (informace pro příjemce) a k žádnému parsování symbolů nebude docházet, ani kdyby se zde vyskytly.

### Elementy response Nová platba - iniciace platby

Tabulka obsahuje jen elementy, které se vyskytují pouze v response zprávě.

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	transactionIdentification	[1..1]	Max35Text	Identifikátor založené transakce
+	serviceLevel	[1..1]	±	Zařazení služby (v rámci typu platby)
++	code	[1..1]	Text	Typ zadané platby
+	signInfo	[1..1]	±	Informace o stavu a id neautorizované transakce
++	state	[1..1]	StateCode	Informace o stavu autorizace transakce
++	signId	[0..1]	Text	Identifikátor autorizačního procesu konkrétní transakce.
++	signInfo	[1..1]	Status Code set	Identifikátor stavu transakce.

Kód	POPIS
DMCT	[DoMestic Credit Transfer] Domestic payment
ESCT	[SEPA Credit Transfer] – SEPA payment
XBCT	[Cross-Border Credit Transfer] – Cross border payment

Hodnoty elementu *serviceLevel.code* – Typ iniciované platby

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

Status kódy plateb – *StatusCode*

## Status založené/iniciované platby (GET /payments/{paymentId}/status)

Resource pro zobrazení stavu platby. Jedná se o založenou platbu, která ještě nebyla klientem autorizována, nebo již byla autorizována a PISP se dotazuje na její stav (GET).

Resource vrací informace jen k transakcím založeným prostřednictvím konkrétního providera. Informace o providerovi je přebírána z certifikátu, resp. z informace o typu licence.

Uživatelská autorizace tohoto resource je optional. Primárně je vyžadován pouze platný certifikát providera.

### Charakteristika resource

**URI:** /payments/{paymentId}/status  
**HTTP Metoda:** GET  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/payments/{paymentId}/status>  
**Authorization:** request **nevyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu.

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

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>API-key</b>	Text	Ne	Volitelný řetězec vydaný komunikující třetí straně jako identifikátor volání této strany primárně sloužící jako konfigurační prvek komunikace.

Parametry hlavičky response:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .

Obsah POST requestu a response volání viz kapitola 0 ELEMENTY ZPRÁVY Status založené/iniciované platby

Chybové kódy definované ČBA standardem pro službu GET Status založené/iniciované platby

HTTP STATUS KÓD	ERROR KÓD	ÚČEL
<b>401</b>	UNAUTHORISED	Nevalidní/chybějící certifikát = provider není autentizován
<b>404</b>	TRANSACTION_MISSING	Volání metody, která neodpovídá licenci, nebo neplatný certifikát.

### Elementy zprávy Status založené/iniciované platby

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP PLATBY	TYP FORMÁTU	PREZENTACE
+	instructionStatus	[1..1]	PISP ALL	StatusCode	Status založené platby

### Status kódy plateb – StatusCode

HTTP STATUS KÓD	STATUSCODE	ÚČEL
<b>200</b>	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful



200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACSP	[AcceptedSettlementInProgress] - All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution
200	ACSC	[AcceptedSettlementCompleted] - 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. It can only be used after bilateral agreement
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not change

## Krok II. Iniciale autorizace platby – specifické pro každou banku (POST /my/payments/{paymentId}/sign/{signId})

Tento resource je určen ke **startu konkrétní autorizační metody** z vybraného scénáře.

Vstupem je JSON objekt obsahující požadovaný typ autorizační metody - **CODE** a všechny elementy specifické pro tento krok.

Výstupem tohoto resource je přehled hodnot potřebných pro dokončení autorizace.

**Např. pro CODE odpovídající federované autorizaci bude odpovědí URL a parametry pro přesměrování na federovanou autorizační stránku.**

A např. pro CODE odpovídající autorizaci prostřednictvím OTP kódu zaslání SMS bude odpovědí jen potvrzení o odeslání kódu. Samotné odeslání iniciuje banka.

### Charakteristika resource

**URI:** /payments/{paymentId}/sign/{signId}  
**HTTP Metoda:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/payments/{paymentId}/sign/{signId}>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu

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

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>API-key</b>	Text	Ne	Volitelný řetězec vydaný komunikující třetí straně jako identifikátor volání této strany primárně sloužící jako konfigurační prvek komunikace.
<b>Authorization</b>	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.

Parametry hlavičky response:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .

Obsah POST requestu a response volání viz kapitola 0 ELEMENTY ZPRÁVY Krok II. Iniciale autorizace platby – specifické pro každou banku

Chybové kódy definované ČBA standardem pro službu POST iniciace autorizace platby:

HTTP STATUS KÓD	ERROR KÓD	ÚČEL
<b>401</b>	UNAUTHORISED	Nevalidní/chybějící access token = uživatel není autentizován
<b>403</b>	FORBIDDEN	Nevalidní/chybějící certifikát = provider není autentizován
<b>404</b>	ID_NOT_FOUND	Požadované id neexistuje
<b>400</b>	AUTH_LIMIT_EXCEEDED	Tento resource není možné autorizovat touto metodou

## Elementy zprávy Krok II. Iniciale autorizace platby - specifické pro každou banku

### Parametry requestu:

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP PLATBY	TYP FORMÁTU	PREZENTACE
+	authorizationType	[1..1]	PISP ALL	Text	Kód požadované autorizace (z autorizačních scénářů)

### Parametry response:

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP PLATBY	TYP FORMÁTU	PREZENTACE
+	authorizationType	[1..1]	PISP ALL	±	Kód požadované autorizace (z autorizačních scénářů)
+	href	[0..1]	PISP ALL	±	Reference pro vyvolání federované autorizace
++	url	[1..1]	PISP ALL	Text	URL link nebo package federované autorizace
++	id	[0..1]	PISP ALL	Text	Případné id pro vyvolání federované autorizace
+	method	[0..1]	PISP ALL	Text	Metoda pro použití href linku u federované autorizace.
+	formData	[0..1]	PISP ALL	±	Volitelný element. V případě metody POST federované autorizace (authorizationType=USERAGENT_REDIRECT) element obsahuje data pro odeslání v přesměrování na federovanou autorizaci.
++	SAMLRequest	[0..1]	PISP ALL	Text	Volitelný parametr. V případě metody POST federované autorizace (authorizationType=USERAGENT_REDIRECT) element obsahuje data SAML requestu.
++	relayState	[0..1]	PISP ALL	Text	Volitelný parametr. V případě metody POST federované autorizace (authorizationType=USERAGENT_REDIRECT) element obsahuje relayState pro návratovou hodnotu.
+	signInfo	[1..1]	PISP ALL	±	Informace o autorizaci instrukce
++	state	[1..1]	PISP ALL	Text	Status autorizace transakce v bankou podporovaném formátu
++	signId	[1..1]	PISP ALL	Text	Jedinečný identifikátor aktuální autorizace transakce

## Dotaz na dostatek prostředků (POST /accounts/balanceCheck)

Jedná se o zdroj pro zaslání dotazu na dostatek prostředků na konkrétním platebním účtu plátce účtu. Tento resource je autorizován. Souhlas s přístupem k informacím musí být udělen klientem mimo interakci tohoto API ještě před tím, než dojde k použití zdroje.

### Charakteristika resource

<b>URI:</b>	/my/payments/balanceCheck
<b>HTTP Metoda:</b>	POST
<b>Request URL:</b>	<a href="https://api.kb.cz/sandbox/pisp/v1/payments/balanceCheck">https://api.kb.cz/sandbox/pisp/v1/payments/balanceCheck</a>
<b>Authorization:</b>	request <b>vyžaduje</b> autorizaci uživatele/klienta jako součást volání API
<b>Certification:</b>	request <b>vyžaduje</b> použití kvalifikovaného certifikátu třetí strany
<b>Stránkování:</b>	ne
<b>Třídění:</b>	ne
<b>Filtrování:</b>	ne

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>API-key</b>	Text	Ne	Volitelný řetězec vydaný komunikující třetí straně jako identifikátor volání této strany primárně sloužící jako konfigurační prvek komunikace.
<b>Authorization</b>	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.
<b>TPP-Name</b>	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“. V tomto poli jsou podporovány pouze znaky bez diakritiky.
<b>TPP-Identification</b>	Text	Ne	Identifikace původního TPP, které request vytvořilo. Např.: „CZ013574-15“

Parametry hlavičky response:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .

Obsah POST requestu a response volání viz kapitola 7.1. Elementy zprávy Dotaz na dostatek prostředků.

Chybové kódy definované pro službu POST Dotaz na dostatek prostředků

HTTP STATUS KÓD	ERROR KÓD	ÚČEL
<b>401</b>	UNAUTHORISED	Chybějící certifikát.
<b>403</b>	FORBIDDEN	Volání metody, která neodpovídá licenci, nebo neplatný certifikát.
<b>400</b>	FIELD_MISSING	Chybějící povinné pole v requestu.
<b>400</b>	FIELD_INVALID	Hodnota pole není validní.
<b>400</b>	AC02	[InvalidDebtorAccountNumber] – nevalidní identifikátoru účtu v obsahu požadavku.
<b>400</b>	AC09	[InvalidAccountCurrency] – uvedena nevalidní měna požadovaného účtu.
<b>400</b>	AC12	[InvalidAccountType] - typ účtu neodpovídá povoleným typům účtů (např. se jedná o neplatební účet).
<b>403</b>	AG01	[TransactionForbidden] – neexistující souhlas s přístupem k informacím o dostatku prostředků na účtu.
<b>400</b>	AM11	[InvalidTransactionCurrency] – v požadavku je uvedena neobchodovaná/nepodporovaná měna.
<b>400</b>	AM12	[InvalidAmount] – chybně zadaná částka. Např. příliš nízká, nebo vysoká částka, či špatný formát čísla včetně počtu desetinných míst dle ISO 4217.
<b>400</b>	FF01	[Invalid File Format] – nevalidní JSON formát, či jiný technický problém se zpracováním dotazu.
<b>400, 50x</b>	NARR	Narrative – obecný důvod pro odmítnutí platby, s doplněním

		informace o chybě.
400	RF01	[NotUniqueTransactionReference] – neunikátní identifikátor žádosti.
400	RR10	[InvalidCharacterSet] – nevalidní znaková sada v požadavku.

#### Elementy zprávy Dotaz na dostatek prostředků

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	exchangeIdentification	[1..1]	Max18Text	Jednoznačná identifikace dotazu
+	card	[0..1]	±	Transakční karta
++	cardholderName	[0..1]	Max45Text	Jméno držitele karty
++	maskedPan	[1..1]	Max30Text	Maskované číslo karty
+	debtorAccount	[1..1]	±	Účet plátce
++	identification	[1..1]	±	Identifikace účtu plátce
+++	iban	[1..1]	IBAN2007Identifier	IBAN
++	currency	[0..1]	CurrencyCode, ISO 4217	Měna účtu plátce
+	authenticationMethod	[0..1]	CodeSet	Metoda ověření klienta
+	merchant	[0..1]	±	Obchodník provádějící transakci
++	identification	[1..1]	Max35Text	Identifikace obchodníka
++	type	[0..1]	Code	Typ obchodníka
++	shortName	[1..1]	Max35Text	Název obchodníka
++	commonName	[1..1]	Max70Text	Jméno obchodníka tak, jak je uvedeno na potvrzení o platbě
++	address	[0..1]	Max140Text	Adresa obchodníka
++	countryCode	[0..1]	CountryCode, ISO 3166 (2 znaková verze kódu)	Země obchodníka
++	merchantCategoryCode	[1..1]	Min3Max4Text, ISO 18245	Kód obchodníka v návaznosti na typ obchodu
+	transactionDetails	[1..1]	±	detaily transakce
++	currency	[1..1]	CurrencyCode, ISO 4217	Měna dotazu na zůstatek
++	totalAmount	[1..1]	Amount	Částka dotazu na zůstatek

#### Elementy response Dotaz na dostatek prostředků

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	responseIdentification	[1..1]	Celé číslo	Jednoznačná identifikace odpovědi na dotaz na dostatek prostředků (ze strany ASPSP).
+	exchangeIdentification	[1..1]	Max18Text	Zopakovaná identifikace platební transakce (dotazu na dostatek prostředků) ze strany vydavatele karty, ke které se dotaz na dostatek prostředků na účtu váže.
+	response	[1..1]	Code set	Výsledek dotazu na dostatek prostředků.

#### Návratové kódy parametru „response“ – Code set:

KÓD	POPIS
APPR	Dostatek prostředků na účtu
DECL	Nedostatek prostředků na účtu

## Nový trvalý příkaz – iniciace platby (POST/my/standingorders)

Resource pro založení nového trvalého příkazu.

### Charakteristika resource

**URI:** /standingorders  
**HTTP Metoda:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/standingorders/>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu.

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

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>API-key</b>	Text	Ne	Volitelný řetězec vydaný komunikující třetí straně jako identifikátor volání této strany primárně sloužící jako konfigurační prvek komunikace.
<b>Authorization</b>	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.
<b>TPP-Name</b>	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“ V tomto poli jsou podporovány pouze znaky bez diakritiky.
<b>TPP-Identification</b>	Text	Ne	Identifikace původního TPP, které request vytvořilo. Např.: „CZ013574-15“

Parametry hlavičky response:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát application/json.

Obsah POST requestu a response volání viz kapitola 0 Elementy zprávy Nová platba – iniciace trvalého příkazu

Chybové kódy definované pro službu POST iniciace trvalého příkazu

HTTP CODE	STATUS	ERROR CODE	ÚČEL
401		UNAUTHORISED	Invalid/missing access token = user is not authenticated
401		UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
403		FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400		FIELD_MISSING	Missing mandatory field in the request.
400		FIELD_INVALID	FIELD value is not valid.
400		AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.

400	AC03	[InvalidCreditorAccountNumber] - recipient account number is closed, blocked, credit card are disabled for the account type, or the recipient's account number is in invalid format (note: only valid for in-house payments). Není používáno.
400	AC09	[InvalidAccountCurrency] – the specified payer account currency does not correspond to the client's account currency for the account number held in the bank (the account currency is optional, in the case of multicurrency accounts, the account currency must be specified - Raiffeisenbank). Využíváno namísto 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. Není podporováno
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. Není podporováno
400	RC10	[InvalidCreditorClearingSystemMemberIdentifier] - invalid recipient bank code identification. Není podporováno
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. Není podporováno
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)

### Elementy zprávy Nová platba – iniciace trvalého příkazu

#### Uvažované typy trvalých příkazů

Trvalé příkazy je povoleno zadávat výhradně jako tuzemskou platbu na území ČR.

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP PLATBY	TYP FORMÁTU	PREZANTACE
+	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 – Není podporováno
++	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. Není podporováno.
+++	currency	[0..0]	ALL	CurrencyCode	Transfer Currency. Není podporováno.
+	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. Toto pole není na vstupu podporováno. Pro alias je v KB využívána hodnota Remittanceinformation ze které je informace přejata do pole alias
++	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) V KB je u definovaného počtu opakování vypočteno a doplněno datum poslední platby 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). Není podporováno
+++	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). Není podporováno
+++	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, <del>IRREGULAR</del> . Není podporováno.
+++	intervalDue	[0..1]	ALL	Number	Value represents order number of the day within particular period when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12 for YEARLY. Není podporováno
++	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..1]	ALL	Amount	Maximum amount to be transferred using the standing order. Only applicable in combination with executionMode AFTER_MAX_AMOUNT_E

					XCEDED. Není podporováno
++++	value	[0..0]	ALL	Amount	Maximum amount of the transfer. Není podporováno
++++	currency	[0..0]	ALL	CurrencyCode	Transfer Currency of the maximum amount. Není podporováno
++	exceptions	[0..0]	ALL		Elements defining exceptions to executing a standing order. Není podporováno.
+++	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. Není podporováno
+++	breaks	[0..0]	ALL	Array of Intervals	Break periods. Není podporováno.
++++	validFromDate	[0..0]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date. Není podporováno.
++++	validToDate	[0..0]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date. Není podporováno.
+	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. Na vstupu je provedena validace oproti zadaným hodnotám KB s hodnotami dále nepracuje. EHP; NONEHP – Není podporováno
+	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 – Není podporováno
+	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 – Není podporováno

+	creditor	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	SEPA; EHP; NONEHP – Není podporováno
+	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 – Není podporováno
+	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 ...	

V případě, že v poli **remittanceInformation.structured.creditorReferenceInformation.reference** bude obsahovat variabilní, konstantní nebo specifický symbol, dojde k jejich rozpoznání a samostatnému uložení do příslušných polí pro tyto symboly v modelu PISP.

- Hodnota **variabilního symbolu** se zaznamenává jako VS:max.10 číslic (např. VS:3451859072).
- Hodnota **konstantního symbolu** se zaznamenává jako KS:max.10 znaků (např. KS:0308).
- Hodnota **specifického symbolu** se zaznamenává jako SS:max.10 číslic (např. SS:8451201274).

JSON příklad elementu:

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

pozn. k poli remittanceInformation.unstructured: dle standardu bankovní asociace může i toto pole obsahovat informace o VS, KS a SS, nicméně my budeme s veškerými informacemi zde obsaženými pracovat tak, jako by se jednalo o prostý popis platby (informace pro příjemce) a k žádnému parsování symbolů nebude docházet, ani kdyby se zde vyskytly.

#### Elementy response Nový trvalý příkaz – iniciace trvalého příkazu

Tabulka obsahuje jen elementy, které se vyskytují pouze v response zprávě.

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	standingOrderIdentification	[1..1]	ALL	Max35Text
++	transactionIdentification	[1..1]	Max35Text	Identifier of established standing order instruction.
+	paymentTypeInfoInformation	[1..1]	ALL	
++	serviceLevel	[1..1]	±	Service placement (within type of payment).
+++	code	[1..1]	Text	Type of entered payment. Je podporováno pouze DMCT - DomesticPayment
+	signInfo	[1..1]	±	Status information and id of unauthorized transactions
++	state	[1..1]	StateCode	Transaction authorization status V KB je vždy vrácen stav OPEN

++	signId	[1..1]	Text	Identifier of the authorization process of a particular transaction.
+	instructionStatus	[1..1]	StatusCode	Status of entered standing order V KB jsou vráceny stavy ACWC nebo ACTC
+	statusChangeInfo	[0..1]	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC".

KÓD	POPIS
DMCT	[DoMestic Credit Transfer] Domestic payment
ESCT	[SEPA Credit Transfer] – SEPA payment. Není podporováno.
XBCT	[Cross-Border Credit Transfer] – Cross border payment. Není podporováno
EXCT	[EEA X-border Credit Transfers] – Cross border payment within the EEP. Není podporováno
NXCT	[Non-EEA X-border Credit Transfers] – Cross border payment outside the EEP. Není podporováno.

#### Hodnoty elementu serviceLevel.code – Typ iniciované platby

#### Status kódy plateb – StatusCode

HTTP STATUS KÓD	STATUS CODE	ÚČEL
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

#### Iniciace autorizace trvalé platby – specifické pro každou banku (POST / standingorders/{transactionIdentification}/sign/{signId})

Tento resource je určen ke **startu konkrétní autorizační metody** z vybraného scénáře.

Vstupem je JSON objekt obsahující požadovaný typ autorizační metody - **CODE** a všechny elementy specifické pro tento krok.

Výstupem tohoto resource je přehled hodnot potřebných pro dokončení autorizace.

**Např. pro CODE odpovídající federované autorizaci bude odpovědí URL a parametry pro přesměrování na federovanou autorizační stránku.**

A např. pro CODE odpovídající autorizaci prostřednictvím OTP kódu zaslání SMS bude odpovědí jen potvrzení o odeslání kódu. Samotné odeslání iniciuje banka.

**Požadavek na autorizaci trvalého příkazu je nutné poslat nejpozději do 5-ti minut od zaslání iniciace platby třetí stranou.**

#### Charakteristika resource

**URI:** standingorders/{transactionIdentification}/sign/{signId}  
**HTTP Metoda:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/standingorders/{transactionIdentification}/sign/{signId}>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu

Stránkování: ne  
 Třídění: ne  
 Filtrování: ne

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>API-key</b>	Text	Ne	Volitelný řetězec vydaný komunikující třetí straně jako identifikátor volání této strany primárně sloužící jako konfigurační prvek komunikace.
<b>Authorization</b>	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.
<b>TPP-Name</b>	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“. V tomto poli jsou podporovány pouze znaky bez diakritiky.
<b>TPP-Identification</b>	Text	Ne	Identifikace původního TPP, které request vytvořilo. Např.: „CZ013574-15“

Parametry hlavičky response:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .

Obsah POST requestu a response volání viz kapitola 0 ELEMENTY ZPRÁVY Krok II. Iniciale autorizace trvalého příkazu – specifické pro každou banku

Chybové kódy definované pro službu POST Autorizace trvalého příkazu:

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

### Elementy zprávy Iniciale autorizace trvalého příkazu

Parametry requestu:

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.
--	--	--	--	--

▪ **Parametry response iniciace autorizace trvalého příkazu:**

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	authorizationType	[1..1]	±	Code of required <i>authorization (from authorization scenarios)</i>
+	Href	[1..1]	±	Reference to call federated authorization
++	url	[1..1]	Text	URL link or package of federated authorization. Vždy vráceno.
++	Id	[1..1]	Text	Possible id for calling federated authorization. Vždy vráceno
+	method	[1..1]	Text	Method to use href link and federated authorization. Vždy vráceno.
+	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 využívá pouze GET.
++	relayState	[0..1]	Text	Optional parameter. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains relayState for returnable value. KB využívá pouze 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

**Detail trvalého příkazu (GET /my/standingorders/{transactionIdentification})**

Resource pro zobrazení informace o založeném trvalém příkazu. Jedná se o trvalý příkaz, který byl přijat k autorizaci, ale ještě nebyl klientem autorizován. Resource pracuje pouze s transakcemi založenými prostřednictvím konkrétního providera.

Resource pro zjištění detailu transakce. Informace o providerovi je přebírána z certifikátu, resp. informaci o licenci.

**Charakteristika resource**

**URI:** /standingorders/{transactionIdentification}  
**HTTP Metoda:** GET  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/standingorders/{transactionIdentification}>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu.

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

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

Query parametry requestu:

Parametry hlavičky requestu:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>API-key</b>	Text	Ne	Volitelný řetězec vydaný komunikující třetí straně jako identifikátor volání této strany primárně sloužící jako konfigurační prvek komunikace.
<b>TPP-Name</b>	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“. V tomto poli jsou podporovány pouze znaky bez diakritiky.
<b>TPP-Identification</b>	Text	Ne	Identifikace původního TPP, které request vytvořilo. Např.: „CZ013574-15“

Parametry hlavičky response:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .

### Elementy zprávy Detail trvalého příkazu

PARAMETER	TYPE	MANDATORY	PURPOSE
<b>transactionIdentification</b>	Max35Text	Yes	Identifier of established standing order instruction

Chybové kódy definované pro službu GET Detail trvalého příkazu

HTTP STATUS CODE	CHYBOVÝ KÓD	ÚČEL
<b>401</b>	UNAUTHORISED	Invalid/missing access token = user is not authenticated
<b>401</b>	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
<b>501</b>	NOT_IMPLEMENTED	Method not implemented
<b>404</b>	TRANSACTION_MISSING	Calling of the method which does not correspond to the licence, or invalid certificate.

### Elementy response zprávy Detail trvalého příkazu

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP PLATBY	TYP FOTMÁRU	PREZENTACE
+	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 – Není podporováno
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInfoInformation	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	amount	[1..1]	ALL	STDO - AmountType3CZ ESST - AmountType3CZ	Payment amount

				XBST AmountType3Choice	-
++	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. V poli alias je vždy zobrazena informace z pole remittanceinformation.
++	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) podle počtu opakování je bankou automaticky doplněno datum poslední platby 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). Není podporováno
+++	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). Není podporováno
+++	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, <del>IRREGULAR</del> — Není podporováno.
+++	intervalDue	[0..0]	ALL	Number	Value represents order number of the day within particular period when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12 for YEARLY. Není podporováno.
++	validity	[1..1]	ALL		Elements restricting the validity of the standing order. V případě, že není vyplněn datum konečné splatnosti, je vráceno vždy prázdné pole.
+++	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. Není podporováno
+++	maxAmount	[0..0]	ALL	Amount	Maximum amount to be transferred using the standing order. Only applicable in combination with executionMode AFTER_MAX_AMOUNT_EXCEEDED. Není podporováno.

++++	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. Není podporováno
+++	breaks	[0..1]	ALL	Array of Intervals	Break periods.
++++	validFromDate	[0..1]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date.
++++	validToDate	[0..1]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date.
+	exchangeRateInformation	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	chargeBearer	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	ChargeBearerType1Code	Fee payer.  EHP - Pouze kontrola formátu zpracování není podporováno
+	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 - Není podporováno
+	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 - Není podporováno
+	creditor	[0..0] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	SEPA, EHP, NONEHP - Není podporováno
+	creditorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	ultimateCreditor	[0..0] [0..0]	TUZEM SEPA	The same object as to initiate payments	SEPA - Není podporováno

		[0..0] [0..0]	EHP NONEHP	chapter ...	
+	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 ...	Je vraceno vždy i v případě, že nebylo nic zadáno.
+	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. Vraceno vždy.
+	instructionStatus	[1..1]	ALL	Status Code set	Transaction status identifier
+	statusChangeInfo	[0..0]	ALL	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC". Není podporováno

### Status založeného/iniciovaného trvalého příkazu (GET /my/standingorders/{transactionIdentification})/status

Resource pro získání stavu trvalého příkazu

#### Charakteristika resource

**URI:** /standingorders/{transactionIdentification}/status  
**HTTP Metoda:** GET  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/standingorders/{transactionIdentification}/status>  
**Authorization:** request **nevyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu.

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

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

PARAMETR	TPY	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>API-key</b>	Text	Ne	Volitelný řetězec vydaný komunikující třetí straně jako identifikátor volání této strany primárně sloužící jako konfigurační prvek komunikace.
<b>TPP-Name</b>	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“. V tomto poli jsou podporovány pouze znaky bez diakritiky.
<b>TPP-Identification</b>	Text	Ne	Identifikace původního TPP, které request vytvořilo. Např.: „CZ013574-15“

Parametry hlavičky response:

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 <b>application/json</b> .

Obsah POST requestu a response volání viz kapitola 0 Elementy zprávy Status založeného/iniciovaného trvalého příkazu

Chybové kódy definované pro službu GET Status založeného/iniciované trvalého příkazu

HTTP STATUS KÓD	CHYBOVÝ KÓD	ÚČEL
401	UNAUTHORISED	Nevalidní/chybějící certifikát = provider není autentizován
404	TRANSACTION_MISSING	Volání metody, která neodpovídá licenci, nebo neplatný certifikát.

### Elementy zprávy Status založeného/iniciovaného trvalého příkazu

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORÁTU	PREZENTACE
+	instructionStatus	[1..1]	StatusCode	Status of entered standing order
+	realizedPayments	[0..0] conditional if it is an authorized standing order	Array	Array of transaction identifiers. Není podporováno
++	transactionIdentification	[1..1]	Max35Text	Identifier of established transaction
++	date	[0..1]	ISODate/ISODateTime	Due date/payment foreign currency in the format ISODate, , or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.STZD, depending on the transaction type and method how the bank presents data (and time) of due date/payment foreign currency. Mainly for card or cash transactions it is posted as ISODateTime.
+	errorinfo	[0..1]	Array	
++	error	[0..1]	String	
++	parameters	[0..1]	Array	
++	message	[0..1]	String	

### Status kódy plateb – StatusCode

HTTP STATUS KÓD	STATUSCODE	ÚČEL
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the
200	ACSP	[AcceptedSettlementInProgress] - All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution
200	ACSC	[AcceptedSettlementCompleted] – Standing order on the debtor's account has been completed



## Dávková platba – iniciace (POST /my/batchpayments)

Resource pro založení dávkové platby.

### Dodatečné informace:

- Dávková platba může samostatně zahrnovat transakce stejného typu tuzemské i SEPA platby (například: pouze SEPA, pouze Domáci).
- Okamžité a zahraniční platby nejsou podporovány.

### Charakteristika resource

**URI:** /batchpayments  
**HTTP Metoda:** POST  
**Request URL:** <https://api.kb.cz/sandbox/pisp/v1/batchpayments>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu.

**Stránkování:** ne

**Třídění:** ne

**Filtrování:** ne

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

PARAMETR	TYP	POVINNÝ	ÚČEL
<b>Content-Type</b>	Text	Ano	Specifikace požadovaného formátu přenosu. Z předpokladu technické specifikace tohoto standardu API je v tomto případě primárně podporován formát <b>application/json</b> .
<b>Authorization</b>	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.
<b>TPP-Name</b>	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“ V tomto poli jsou podporovány pouze znaky bez diakritiky.
<b>X-Request-ID</b>	Text	Ano	Unikátní identifikátor požadavku aplikace TPP.

### Elementy request Dávková platba – iniciace

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	exchangeIdentification	[1..1]	Max14Text	Jednoznačná identifikace dotazu.
+	instructionName	[0..1]	Max35Text	Volitelné pojmenování dávky.
+	payments	[1..1]	Max35Text	Jednoznačná identifikace konkrétní transakce pro autorizační proces.

**Elementy response Dávková platba – iniciace**

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	transactionIdentification	[1..1]	Max35Text	Identifikátor založené dávky.
+	signInfo	[1..1]	±	Informace o stavu a id neautorizované dávky.
++	state	[1..1]	StateCode	Informace o stavu autorizace transakce.
++	signId	[1..1]	Text	Identifikátor autorizačního procesu konkrétní dávky.
+	instructionStatus	[1..1]	StatusCode	Status založené dávky
+	statusChangeInfo	[0..1]	Text	Určuje změnu, která byla provedena v instrukci. Používá se s instructionStatus "ACWC".
+	batchDigest	[1..1]	Max50Text	Unikátní ID, zabalené do batch SHA HASH, pomoci, kterého jde zkontrolovat, že mezi iniciací a autorizací dávky nedošlo k žádné změně.

## Status codes of payment – StatusCode

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

## Chybové kódy definované pro službu POST iniciace dávkové platby

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Chybí kvalifikovaný certifikát.
403	FORBIDDEN	Volání metody, která neodpovídá licenci, nebo neplatný kvalifikovaný certifikát.
404	[ID_NOT_FOUND]	V případě, že je pole vyplněno, ale je nevalidní.
404	NOT_FOUND	Pokud parametr není vyplněn nebo pole neodpovídá max. délce.
400	TYPE_DIFFERENT	Platby mohou být jednoho typu Domestic nebo jen SEPA.
400	FIELD_MISSING	Povinné pole nebylo vyplněno.
400	FIELD_INVALID	Hodnota pole není validní.
400	AC12	[InvalidAccountType] Typ účtu neodpovídá povoleným typům účtů. Např. se jedná o neplatební účet.
400	NARR	[Exceeded number of orders in a single batch]
400	NARR	[Batch payment is empty]
400	NARR	[ID is not unique]
400	NARR	[AccessDenied - Unauthorized access to account.]
400	NARR	[AccessDenied - User is not in active state.]
400	NARR	[AccessDenied - Client is not in active state.]
400	NARR	[AccessDenied - Debtor payment account is not in active state.]
400	NARR	[AccessDenied - Debtor payment account is not available in internet banking]
400	NARR	[Foreign payment - unsupported payment type.]
400	NARR	[Transaction has already been authorized or rejected.]
400	NARR	[Payment has already been incorporated into another batch payment.]
400	NARR	[Invalid Originator TPP]
400	NARR	[Payments initiated by different applications]
400	NARR	[Daily limit of transactions in batches exhausted.]

## Iniciace autorizace dávkové platby (POST /my/ batchpayments/{transactionIdentification}/sign/{signId})

Tento resource je určen ke startu konkrétní autorizační metody z vybraného scénáře.

Vstupem je JSON objekt obsahující požadovaný typ autorizační metody - CODE a všechny elementy specifické pro tento krok.



Výstupem tohoto resource je přehled hodnot potřebných pro dokončení autorizace.

Např. pro CODE odpovídající federované autorizaci bude odpovědí URL a parametry pro přesměrování na federovanou autorizační stránku.

A např. pro CODE odpovídající autorizaci prostřednictvím OTP kódu zasláného SMS bude odpovědí jen potvrzení o odeslání kódu. Samotné odeslání iniciuje banka.

### Charakteristika resource

**URI:** </batchpayments/{transactionIdentification}/sign/{signId}>  
**HTTP Metoda:** POST  
**RequestURL:** <https://api.kb.cz/sandbox/pisp/v1/batchpayments/{transactionIdentification}/sign/{signId}>  
**Authorization:** request **vyžaduje** autorizaci uživatele/klienta jako součást volání API  
**Certification:** request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu.

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

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

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 <b>application/json</b> .
Authorization	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.
TPP-Name	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“ V tomto poli jsou podporovány pouze znaky bez diakritiky.
X-Request-ID	Text	Ano	Unikátní identifikátor požadavku aplikace TPP.

Path parametry:

PARAMETR	TYP	POVINNÝ	ÚČEL
transactionIdentification	Text	Ano	Identifikátor založené dávky.
signId	Text	Ano	Identifikátor autorizačního procesu konkrétní dávky.

### Elementy request iniciace autorizace dávkové platby

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	authorizationType	[1..1]	Text	Kód požadované autorizace (z autorizačních scénářů) Povolena hodnota "USERAGENT_REDIRECT",
+	redirectUrl	[0..1]	Text	URL odkaz nebo balíček adresy pro zpětné volání federované autorizace. Tuto adresu používá autorizace federované banky k přesměrování zpět do aplikace TPP po dokončení autorizace.

+	batchDigest	[0..1]	Max50Text	Unikátní ID, zabalené do batch SHA HASH, pomocí kterého jde zkontrolovat, že mezi iniciací a autorizací dávky nedošlo k žádné změně. Doporučujeme jeho využití z pohledu bezpečnosti.
---	-------------	--------	-----------	---

#### Elementy response iniciace autorizace dávkové platby

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP PLATBY	TYP FORMÁTU	PREZENTACE
+	authorizationType	[1..1]	PISP ALL	±	Kód požadované <i>autorizace</i> (z <i>autorizačních scénářů</i> )
+	href	[0..1]	PISP ALL	±	Reference pro vyvolání federované autorizace
++	url	[1..1]	PISP ALL	Text	URL link nebo package federované autorizace
++	id	[0..1]	PISP ALL	Text	Případné id pro vyvolání federované autorizace
+	method	[0..1]	PISP ALL	Text	Metoda pro použití href linku u federované autorizace.
+	formData	[0..1]	PISP ALL	±	Volitelný element. V případě metody POST federované autorizace (authorizationType=USERAGENT_REDIRECT) element obsahuje data pro odeslání v přesměrování na federovanou autorizaci.
++	SAMLRequest	[0..1]	PISP ALL	Text	Volitelný parametr. V případě metody POST federované autorizace (authorizationType=USERAGENT_REDIRECT) element obsahuje data SAML requestu.
++	relayState	[0..1]	PISP ALL	Text	Volitelný parametr. V případě metody POST federované autorizace (authorizationType=USERAGENT_REDIRECT) element obsahuje relayState pro návratovou hodnotu.
+	signInfo	[1..1]	PISP ALL	±	Informace o autorizaci instrukce.
++	state	[1..1]	PISP ALL	Text	Status autorizace transakce v bankou podporovaném formátu.
++	signId	[1..1]	PISP ALL	Text	Jedinečný identifikátor aktuální autorizace dávky.

#### Chybové kódy definované pro službu POST iniciace autorizace dávkové platby

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Chybí kvalifikovaný certifikát.
403	FORBIDDEN	Volání metody, která neodpovídá licenci, nebo neplatný kvalifikovaný certifikát.
404	[ID_NOT_FOUND]	V případě, že je pole vyplněno, ale je nevalidní.
400	FIELD_MISSING	Povinné pole nebylo vyplněno.
400	FIELD_INVALID	Hodnota pole není validní.
400	AC12	[InvalidAccountType] Typ účtu neodpovídá povoleným typům účtů. Např. se jedná o neplatební účet.
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]

#### Dávková platba – status (GET/my/batchpayments/{transactionIdentification}/status)

Resource pro zjištění statusu založené dávkové platby.

#### Charakteristika resource

URI: </batchpayments/{transactionIdentification}/status>

HTTP Metoda: GET

Request URL:

<https://api.kb.cz/sandbox/pisp/v1/batchpayments/{transactionIdentification}/status>

Authorization: request **nevyžaduje** autorizaci uživatele/klienta jako součást volání API

Certification: request **vyžaduje** použití kvalifikovaného certifikátu třetí strany v rámci navázání oboustranné TLS komunikace se serverem. Třetí strana je identifikována ověřením platnosti a obsahu tohoto certifikátu.

Stránkování: ne

Třídění: ne

Filtrování: ne

Query parametry requestu: **nedefinovány**

Parametry hlavičky requestu:

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 <b>application/json</b> .
Authorization	Text	Ano	Parametr slouží pro předání access tokenu autentizovaného uživatele spolu s jeho typem.
TPP-Name	Text	Ano	Název původního TPP, které request vytvořilo. Např.: „Star Corporation, a.s.“ V tomto poli jsou podporovány pouze znaky bez diakritiky.
X-Request-ID	Text	Ano	Unikátní identifikátor požadavku aplikace TPP.

Path parameter:

PARAMETR	TYP	POVINNÝ	ÚČEL
transactionIdentification	Text	Ano	Identifikátor založené dávky.

#### Elementy response Dávková platba – Status

ÚROVEŇ	PRVEK ZPRÁVY	VÝSKYT	TYP FORMÁTU	PREZENTACE
+	transactionIdentification	[1..1]	Max35Text	Identifikátor založené dávky.
+	signInfo	[1..1]	±	Informace o stavu a id neautorizované dávky.
++	state	[1..1]	StateCode	Informace o stavu autorizace dávky.
++	signId	[1..1]	Text	Identifikátor autorizačního procesu konkrétní dávky.
+	instructionStatus	[1..1]	StatusCode	Status založené dávky
+	batchDigest	[1..1]	Max50Text	Unikátní ID, zabalené do batch SHA HASH, pomoci, kterého jde zkontrolovat, že mezi iniciací a autorizací dávky nedošlo k žádné změně.
+	payments	[0..n]	±	Identifikace konkrétních transakcí.
++	transactionIdentification	[1..1]	Max35Text	Identifikátor založené transakce.
++	instructionIdentification	[0..1]	Max35Text	Identifikace instrukce. Pokud není vyplněna, tak se doplní NOTPROVIDED.
++	instructionStatus	[1..1]	StatusCode	Status založené platby.
++	errorInfo	[0..1]	±	Používá se když je instructionStatus na úrovni transakce "RJCT"
+++	code	[0..1]	Max15Text	Chybový kód
+++	description	[0..1]	Max150Text	Detail popisu chybové zprávy.

#### Chybové kódy definované pro službu GET status dávkové platby

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Chybí kvalifikovaný certifikát.
403	FORBIDDEN	Volání metody, která neodpovídá licenci, nebo neplatný kvalifikovaný certifikát.
404	[ID_NOT_FOUND]	V případě, že je pole vyplněno, ale je nevalidní.
400	FIELD_MISSING	Povinné pole nebylo vyplněno.
400	FIELD_INVALID	Hodnota pole není validní.
400	NARR	[Invalid Originator TPP]



## 10. PSD2 slovník - výběr

**API** – Application Programming Interface

**REST** - (Representational State Transfer) - REST je architektura API, která nám umožňuje přistupovat k datům a provádět nad nimi CRUD operace. Architektura, která pracuje obvykle přes protokol HTTP/HTTPS. REST je bezstavový, čímž jednak značně zjednodušuje komunikaci s API a umožňuje paralelní zpracování obsahu. Zároveň ho lze dost snadno použít s HTTP, což je velice rozšířený protokol. V neposlední řadě nám poskytuje určitý standard, takže není problém použít cizí API nebo naopak nabízet vlastní API velkému množství dalších uživatelů. Rozhraní REST je použitelné pro jednotný a snadný přístup ke zdrojům (resources). Zdrojem mohou být data, stejně jako stavy aplikace (pokud je lze popsat konkrétními daty). Všechny zdroje mají vlastní identifikátor URI a REST definuje čtyři základní metody pro přístup k nim (GET, PUT, POST, DELETE). Významy jednotlivých HTTP Verbs jsou následující:

- GET - získání dat
- POST - vytvoření
- PUT - úpravy (upraví celý zdroj - chová se jako SET)
- DELETE - smazání
- PATCH - částečné úpravy

**REST API** - rozhraní pro distribuované prostředí orientované na data, nikoli na volání procedur jako např. RPC (XML-RPC) či SOAP. Webové služby definují vzdálené procedury a protokol pro jejich volání, REST určuje, jak se přistupuje k datům. REST API využívá metody HTTP jako jsou @GET, @PUT, @POST, @DELETE, @PATCH.

**TPP** – Third Party Provider (třetí strana registrovaná/licencovaná ČNB)

**AIS** - Account Information Service

**AISP** - Account Information Service Provider

**PIS** - Payment Initiation Service

**PISP** - Payment Initiation Service Provider

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

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

**ASPSP** – Account Servicing Payment Service Provider (banka vedoucí platební účet plátce)