



Technical Manual for Implementing the Batch Payment Direct API

21 March 2023

Version 4.0

CHANGE LOG

Doc Version	Date of effectiveness	Change description
1.0	15.07.2021	Public document
2.0	07.12.2021	Added new functionality to provide batch payment initialization request in KM payload format.
3.0	09.12.2021	Update and correction of parameter „Content-Type“
4.0	21.03.2023	Update of API version to v3 in all API endpoints.

TABLE OF CONTENTS

Change log	1
Background.....	2
1. Introduction	2
2. Batch Payment Initialization.....	4
3. Batch Payment Status	12
4. Batch Payment Detail	15
5. Batch Payment API – Swagger definition	17



BACKGROUND

- In connection with the relationship with KB, the Partner shall comply with the rules described in the [Code of Prudent Conduct](#).
- Before reading this Manual, we advise you to read the general description of the service. It also specifies the requirements and actions that must be met or taken prior to the technical connection to the Batch Payment Direct API.
- It is also necessary to pass the technical registration (see document Technical Manual for Registration) as a precondition before sending batch payments.

1. INTRODUCTION

KB issues the Batch Payment Initiation API, which contains of payment initiation service, information on the batch payment status including status of single payments and detail of batch payment.

Batch payment processing characteristics in general:

- Authorisation of initiated batch payment can solely be done within current KB direct channels such as Moje Banka Business, Mobilní Banka Business and Profibanka using standard KB security methods.
- The batch payment API can accept domestic and SEPA payments.
- Batch can be send in:
 - **online mode** to be processed during the business day,
 - **batch mode** to be processed overnight,
 - **continuous mode** which is a combination of both before mentioned modes (payment orders that cannot be processed during the day due to insufficient funds do not enter the system of multi-round processing of orders and they are sent for night processing.)
- In online mode maximum of 1 000 single payments can be send into KB within multiple batches. For batch mode the limitation for number of payment is not applied.
- One batch payment can contain maximum of 100 payments.
- Instant payments are not supported/allowed within the batch payment.
- The Batch Payment Initiation API can only be applied to payment accounts.
- Initiated batch payment can be altered in KB direct channels in the same way as it is possible now.

Batch payment processing characteristics in JSON format:

- The format is based on the standards defined by the Czech Banking Association in the [Czech Open Banking Standard](#).
- In JSON format KB can accept domestic or SEPA payments.
- Only one type of payment can be sent and processed during a single call (i.e. one batch can contain either set of domestic payments, or set of SEPA payments, not their combination).
- The allowed character set is based on the CERTIS character set for domestic payments and on the SWIFT character set for SEPA payments.

Batch payment processing characteristics in KM format:

- Only domestic payments can be imported using this format.
- Batch of direct debits are not supported.
- Payments submitted via the KM format may only be in CZK currency.
- The format is based on the document specification "[Klientský formát KM podporovaný v KB](#)"
- The UTF-8 character set is used for the file content.

- Batch payment initialization request payload in KM is encode to BASE64 format.

Time limitations:

- Batch payment orders with "batch" or "continuous" types of processing received by 18:00 will have due dates on the current banking day (until 20:30 for batch payment orders with the "online" type of processing). Orders received after this time will be due on the next banking day. To ensure their correct processing, their due dates should be equal to the date of the next banking day.
- The due date of the payments received for the processing on a KB business day between 20:31 and 23:59 hrs will fall on the next business day.

List of available resources:

1. POST new batch payment - payment initiation
2. GET initiated batch payment status
3. GET initiated batch payment detail

2. BATCH PAYMENT INITIALIZATION

Objective	Initiate batch payment for further authorization in KB direct channels
Type	REST API
Called API	[POST] https://api.kb.cz/directapi/batchda/v3/batchPayments
Requirements	Valid access_token
Request data	JSON/KM payload, access_token,
Response data	Information about initiated batch payment
Authorisation	The request requires an authorisation by the user/client as part of the API call.

Authorization header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
x-api-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as the configuration element of communication.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.

Request header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
x-correlation-id	string	Yes	Unique identifier (GUID) for each request specified by TPP.
x-exchange-identification	string	Yes	Batch identifier assigned by a third party.
x-instruction-name	string	No	Optional batch name.
x-batch-processing-mode	string	Yes	Available values : ONLINE , CONTINUOUS , BATCH
Content-Type	Text	Yes	A specification of the required transfer format. Based on the prerequisites of the technical specification of this API standard, in this case the application/json and application/vnd.batchda.api+km format is primarily supported.

Response header parameters:

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

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

General error codes defined for the batch payment initiation POST service

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Missing token.
403	FORBIDDEN	Calling of the method that does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC03	[InvalidCreditorAccountNumber] – creditor account number is invalid or missing.
400	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
400	AC12	[InvalidAccountType] – the account type does not match allowed account types (e.g., a non-paying account).
400	AM11	[InvalidTransactionCurrency] – the request contains a currency that is not traded/supported.
400	AM12	[InvalidAmount] – wrong amount, e.g., too low or high amount or a wrong number format in terms of the number of decimal places according to ISO 4217.
400	FF01	[Invalid File Format] – invalid JSON format or other technical problem with the query processing.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RF01	[NotUniqueTransactionReference] – the request identifier is not unique.



400	RR10	[InvalidCharacterSet] – invalid character set in the request.
400	TYPE_DIFFERENT	Requested payments do not have the same serviceLevel.

KM format specific error codes defined for the batch payment initiation POST service

HTTP code	Error code	Message	Additional identifier	Scope
400	NARR	accounting code field contains non-numerical characters.	DA068	trn.debtorAccount
400	NARR	Account and contra-account numbers are identical.	DA069	-
400	NARR	Invalid row.	DA070	km.lineStructure
400	NARR	UHL file header must be specified at beginning of file.	DA071	km.uhl1
400	NARR	File must start with UHL1 header.	DA072	km.uhl1
400	NARR	Range of acc. files contains non-numerical characters.	DA073	km.uhl1
400	NARR	Range of acc. files contains non-numerical characters.	DA074	km.uhl1
400	NARR	Invalid location of file header - HSO.	DA075	hso.structure
400	NARR	rmoo field contains non-numerical characters.	DA076	hso.rmoo
400	NARR	Invalid rmoo field.	DA077	hso.rmoo
400	NARR	Account file number contains non-numerical characters.	DA078	hso.accountingFileNumber
400	NARR	Accounting file number is outside range specified in file header.	DA079	hso.accountingFileNumber
400	NARR	Invalid debtor bank code.	DA080	hso.debtorBankCode
400	NARR	Invalid format of group header - HSK.	DA081	hsk.structure
400	NARR	Invalid length of due date field.	DA082	hsk.executionDate
400	NARR	Due date field contains non-numerical characters.	DA083	hsk.executionDate
400	NARR	Group header must precede transactions.	DA084	trn.structure
400	NARR	Invalid number of fields in order.	DA085	trn.structure
400	NARR	Invalid number of fields in multiple order.	DA086	trn.structure



400	NARR	Invalid fields of constant symbol and bank	DA087	trn.constantSymbol
400	NARR	Invalid location of group end - KSK.	DA088	ksk.structure
400	NARR	Invalid format of group end - KSK.	DA089	ksk.structure
400	NARR	Invalid location of file end - KSO.	DA090	kso.structure
400	NARR	Invalid format of file end - KSO.	DA091	kso.structure
400	NARR	Total amount field in group header contains non-numerical characters.	DA092	hsk.batchSum
400	NARR	Total amount in group batch does not match sum of individual transactions.	DA093	hsk.batchSum
400	NARR	Beginning of range of acc. files greater than end.	DA094	km.uhl1
400	NARR	Invalid format of file header - HSO.	DA095	hso.structure
400	NARR	Invalid rmoo field length.	DA096	hso.rmoo
400	NARR	Invalid length of accounting file number.	DA097	hso.accountingFileNumber
400	NARR	Duplicate number of accounting file.	DA098	-
400	NARR	Invalid length of SS.	DA099	trn.specificSymbol
400	NARR	Batch rejected.	DA100	km.lineStructure
400	NARR	Required data not filled in: variable symbol	DA101	trn.variableSymbol
400	NARR	Direct Debit is not a supported type	DA102	hso.rmoo
400	NARR	Invalid creation date format	DA103	km.creationDate
400	NARR	Creation date field contains non-numerical characters.	DA104	km.creationDate
400	NARR	Invalid variable symbol format	DA105	trn.variableSymbol
400	NARR	Invalid constant symbol format	DA106	trn.constantSymbol
400	NARR	Invalid specific symbol format	DA107	trn.specificSymbol
400	FIELD_MISSING		DA0001	hsk.batchSum hsk.executionDate trn.debtorAccount trn.creditorAccount trn.amount trn.variableSymbol trn.constantSymbol
400	FIELD_INVALID		DA0002	hsk.executionDate trn.debtorAccount trn.creditorAccount

**Available payment status codes – StatusCode:**

HTTP STATUS CODE	STATUS CODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] – The authentication and syntactical and semantical validation are successful.
200	ACWC	[AcceptedWithChange] – An instruction is accepted. One or more transactions have not been accepted. Please obtain detailed information about the batch before proceeding to authorization. If payment status code ACWC appear than we fill in element statusChangeInfo on value BP01.

Example of batch payment initialization request payload in JSON:

```
{
  "batchPaymentDataCollectionRequest": [
    {
      "debtorAccount": {
        "identification": {
          "iban": "CZ3501000001000000000000"
        },
        "currency": "CZK"
      },
      "creditorAccount": {
        "identification": {
          "iban": "CZ4730300001000000000000"
        },
        "currency": ""
      },
      "amount": {
        "instructedAmount": {
          "value": "1",
          "currency": "CZK"
        }
      },
      "requestedExecutionDate": "2021-12-03",
      "paymentIdentification": {
        "instructionIdentification": "platba č. 001"
      },
      "paymentTypeInformation": {
        "instructionPriority": "NORM"
      },
      "remittanceInformation": {
        "unstructured": "Moje platba 1",
        "structured": {
          "creditorReferenceInformation": {
            "reference": [
              "VS:111",
            ]
          }
        }
      }
    }
  ]
}
```



```
        "KS:222",
        "SS:333"
    ]
}
},
{
    "debtorAccount": {
        "identification": {
            "iban": "CZ3501000001000000000000"
        },
        "currency": "CZK"
    },
    "creditorAccount": {
        "identification": {
            "iban": "CZ4730300002000000000000"
        },
        "currency": ""
    },
    "amount": {
        "instructedAmount": {
            "value": "2",
            "currency": "CZK"
        }
    },
    "requestedExecutionDate": "2021-12-03",
    "paymentIdentification": {
        "instructionIdentification": "platba č. 002"
    },
    "paymentTypeInformation": {
        "instructionPriority": "NORM"
    },
    "remittanceInformation": {
        "unstructured": "Moje platba 2",
        "structured": {
            "creditorReferenceInformation": {
                "reference": [
                    "VS:111",
                    "KS:222",
                    "SS:333"
                ]
            }
        }
    }
}
]
```

Example of batch payment initialization request payload in KM (decode from BASE64 format):

```
UHL1251021BATCHDA.CONVERSION.TOOL.1.0.00000999
1 1501 692001 0100
2 1235087960217 00200 061221
1439034017 0100 111 0030302222 333 AV:Test 123
1439034017 0100 111 0030302222 333 AV:Test 123
3 +
2 1235087960217 1000 061221
1439034017 1000 111 0030302222 333 AV:Test 123
3 +
2 1235087960217 2000 061221
1439034017 2000 111 0030302222 333 AV:Test 123
3 +
5 +
```

Example of batch payment initialization request payload in KM (encode to BASE64 format):

```
VUhMMTI1MTAyMUJBVENIREEuQ09OVkVSU01PTi5UT09MLjEuMC4wMDAwMDk5OQ0KMSAxNTA
xIDY5MjAwMSAwMTAwDQoyIDEyMzUwODc5NjAyMTcgMDAyMDAgMDYxMjIxIQoxNDM5MDM0MD
E3IDAxDGAgMTEwIDAAMzAzMDIyMjIyMzIEFWO1Rlc3QgMTIzDQoxNDM5MDM0MDE3IDAAM
DAGMTEwIDAAMzAzMDIyMjIyMzIEFWO1Rlc3QgMTIzDQoxNDM5MDM0MDE3IDAAMzAzMDIy
NyAxMDAwIDA2MTIyMQ0KMTQzOTAzNDANyAxMDAwIDExMSAwMDMwMzAyMjIyIDEyIDMzMyBBVj
pUZXR0IDEyMzUwODc5NjAyMTcgMDAwMCAwNjEyMjENCjE0MzkwMzQwMT
cgMjAwMCAwMTEwIDAAMzAzMDMwMjIyMjIyMzIEFWO1Rlc3QgMTIzDQoxNDM5MDM0MDE3
```

Example of batch payment initialization response:

```
{
  "batchDigest": "4F9FCE3F3AE76D54733C565CD40AD8CC0A34DA9B",
  "batchProcessingMode": "BATCH",
  "exchangeIdentification": "871233",
  "instructionStatus": "ACTC",
  "rejectedTransactionCreditCount": 0,
  "rejectedTransactionDebitCount": 0,
  "signInfo": {
    "signId": "0000146C42",
    "state": "OPEN"
  },
  "totalCreditAmount": 0,
  "totalDebitAmount": 3,
  "transactionCreditCount": 0,
  "transactionDebitCount": 2,
  "transactionIdentification": "0000146C42"
}
```

Example of error code for KM:

```
"error": "NARR",
"parameters": {
  "additionalProp1": "Lane:5",
  "additionalProp2": "DA068",
  "additionalProp3": {}
},
"scope": "trn.debtorAccount",
"message": "accounting code field contains non-numerical
characters."
```

3. BATCH PAYMENT STATUS

Objective	To obtain status of initiated batch payment
Type	REST API
Called API	[GET] https://api.kb.cz/directapi/batchda/v3/batchPayments/{transactionIdentification}/status
Requirements	Valid access_token
Request data	access_token
Response data	Status of initiated batch payment
Authorisation	The request requires an authorisation by the user/client as part of the API call.

Query parameters of the request: not defined

Authorization header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
x-api-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as the configuration element of communication.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.

Request header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
transactionIdentification	string	Yes	Unique bank batch identification
batchDigest	string	No	Unique batch digest
x-correlation-id	string	Yes	Unique identifier (GUID) for each request specified by TPP.

Response header parameters:

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

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

Error codes defined for the batch payment initiation POST service

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Missing token.
403	FORBIDDEN	Calling of the method that does not correspond to the licence, or invalid certificate.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.

Available payment status codes – StatusCode

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

Example of status initiated batch payment response in JSON:

```
{
  "batchDigest": "4F9FCE3F3AE76D54733C565CD40AD8CC0A34DA9B",
  "batchPaymentStatusCollectionResponse": [
    {
      "instructionIdentification": "217353",
      "instructionStatus": "ACTC",
      "transactionIdentification": "HU000000ZWA"
    },
    {
      "instructionIdentification": "282482",
      "instructionStatus": "ACTC",
      "transactionIdentification": "HU000000ZWB"
    }
  ],
  "batchProcessingMode": "BATCH",
  "instructionStatus": "ACTC",
  "rejectedTransactionCreditCount": 0,
  "rejectedTransactionDebitCount": 0,
  "signInfo": {
    "signId": "0000146C42",
    "state": "OPEN"
  },
  "totalCreditAmount": 0,
  "totalDebitAmount": 3,
  "transactionCreditCount": 0,
  "transactionDebitCount": 2,
  "transactionIdentification": "0000146C42"
}
```

4. BATCH PAYMENT DETAIL

Objective	To obtain status of initiated batch payment
Type	REST API
Called API	[GET] https://api.kb.cz/directapi/batchda/v3/batchPayments/{transactionIdentification}
Requirements	Valid access_token
Request data	access_token
Response data	Detail of initiated batch payment
Authorisation	The request requires an authorisation by the user/client as part of the API call.

Authorization header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
x-api-key	Text	No	An optional string issued to a communicating third party as the call identifier of that party primarily serving as the configuration element of communication.
Authorization	Text	Yes	A parameter used for forwarding the authenticated user's access token along with its type.

Request header parameters:

PARAMETER	TYPE	MANDATORY	PURPOSE
transactionIdentification	string	Yes	Unique bank batch identification
batchDigest	string	No	Unique batch digest
x-correlation-id	string	Yes	Unique identifier (GUID) for each request specified by TPP.

Response header parameters:

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

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

Error codes defined for the batch payment initiation POST service

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Missing token.
403	FORBIDDEN	Calling of the method that does not correspond to the licence, or invalid certificate.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.

Example of detail of initiated batch payment response in JSON:

```
{
  "transactionIdentification": "0000146C42",
  "batchDigest": "4F9FCE3F3AE76D54733C565CD40AD8CC0A34DA9B",
  "transactionCreditCount": 0,
  "rejectedTransactionCreditCount": 0,
  "transactionDebitCount": 2,
  "rejectedTransactionDebitCount": 0,
  "totalCreditAmount": 0,
  "totalDebitAmount": 3,
  "batchProcessingMode": "BATCH",
  "signInfo": {
    "state": "OPEN",
    "signId": "0000146C42"
  },
  "instructionStatus": "ACTC",
  "batchPaymentDataCollectionResponse": [
    {
      "PaymentIdentification": {
        "instructionIdentification": "217353",
        "transactionIdentification": "HU000000ZWA"
      },
      "paymentTypeInformation": {
        "instructionPriority": "NORM",
        "serviceLevel": {"code": "DMCT"}
      },
      "amount": {"instructedAmount": {
        "value": 1,
        "currency": "CZK"
      }},
      "requestedExecutionDate": "2021-12-03",
      "debtorAccount": {
        "identification": {"iban": "CZ4730300002000000000000"},
        "currency": "CZK"
      },
      "creditorAccount": {
        "identification": {"iban": "CZ4730300002000000000000"},
        "currency": "CZK"
      },
      "remittanceInformation": {
        "unstructured": "Moje platba 1",

```



```
    "structured": {"creditorReferenceInformation":
{"reference": [
    "KS:222",
    "SS:333",
    "VS:111"
  ]}}
  },
  "instructionStatus": "ACTC"
},
{
  "PaymentIdentification": {
    "instructionIdentification": "282482",
    "transactionIdentification": "HU000000ZWB"
  },
  "paymentTypeInformation": {
    "instructionPriority": "NORM",
    "serviceLevel": {"code": "DMCT"}
  },
  "amount": {"instructedAmount": {
    "value": 2,
    "currency": "CZK"
  }},
  "requestedExecutionDate": "2021-12-03",
  "debtorAccount": {
    "identification": {"iban": "CZ3501000001000000000000"},
    "currency": "CZK"
  },
  "creditorAccount": {
    "identification": {"iban": "CZ4730300002000000000000"},
    "currency": "CZK"
  },
  "remittanceInformation": {
    "unstructured": "Moje platba 2",
    "structured": {"creditorReferenceInformation":
{"reference": [
    "KS:222",
    "SS:333",
    "VS:111"
  ]}}
  },
  "instructionStatus": "ACTC"
}
]
}
```

5. BATCH PAYMENT API – SWAGGER DEFINITION

Swagger definition of all API resources ...