

TECHNICAL DOCUMENT

GSA PROTOCOL

CREATOR	MONET+,a.s. Za Dvorem 505, Zlín - Štípa Ondřej Grygar Gabriel Bílý
UPDATED	13.04.2022
VALID TO	next revision of the document
VERSION #	1.5

The ownership of the document remains with MONET+,a.s. The present document may not be reproduced and/or disclosed, in whole or in part, to any third party without the previous written confidentiality agreement. The information contained in the present document may be used for the purpose specified in the document title.

1. CONTENT

CONTENT	2
DOCUMENT VERSION HISTORY	5
LIST OF TERMS / ABBREVIATIONS	6
PURPOSE OF DOCUMENT	7
REST API (GSA version v1)	8
Cash Register Client/Terminal Server	8
Info Method	9
Terminal Status Method	10
Start Payment Method	11
Start MOTO Payment Method	12
Start Refund Method	14
Start Reverse Method	15
Start Preauth Method	16
Get available preauthorizations	17
Start Preauth completion Method	17
Start Preauth increment Method	19
Start Preauth cancel Method	20
Get preauth transactions	21
Status Method	22
Result Method	23
Confirm Method	27
Start Closing Method	28
Cancel transaction	29
Daughter companies	30
Last transaction	30
Get successful transactions	33
Cash register and terminal work as both client and server	36
Start Payment Method with callback	36
Start MOTO Payment Method with callback	38
Start Refund Method with callback	41
Start Reverse Method with callback	42
Start Preauth Method with callback	44
Start Preauth completion Method with callback	46
Start Preauth increment Method with callback	48

Start Preauth cancel Method with callback	50
Start Closing Method with callback	51
Info Method	52
Terminal Status Method	52
Status Method	52
Result Method	52
Confirm Method	52
Switchio Pay 3.1.0 – GSA V2	53
Cash Register Client/Terminal Server	53
Get transaction status	54
Info V2	54
Status V2	55
Payment polling V2	56
MOTO payment polling V2	58
Refund polling V2	60
Reversal polling V2	62
Preauthorization polling V2	64
PreAuth Complete – polling V2	66
PreAuth Increment – polling V2	68
PreAuth Cancel – polling V2	70
Get available preauthorizations V2	72
Operation status V2	73
Confirm transaction V2	74
Closing polling V2	75
Cancel transaction V2	76
Daughter companies V2	77
Last transaction V2	78
Get successful transactions V2	81
Transaction result V2	84
Cash register and terminal work as both client and server	88
Payment callback V2	89
MOTO payment callback V2	91
Refund callback V2	93
Reversal callback V2	95
Preauthorization callback V2	97
PreAuth Complete – callback V2	99
PreAuth Increment – callback V2	101
PreAuth Cancel – callback V2	103
Closing callback V2	105

CASHDESK	106
Login	106
Terminal Operation Examples	108
New payment	108
Refund	111
Reversal	114
Closing	117
APPENDIX 1: ENUMS	119
Status parameter	119
CardInputType parameter	119
CvmTypeList parameter	120
TransactionType parameter	120
ResultCode parameter	121
SLIP	134

2. DOCUMENT VERSION HISTORY

VERSION	DATE	AUTHOR	DESCRIPTION
1.0	10.11.2020	Ondřej Grygar Gabriel Bílý	1st version of the document
1.1	08.03.2021	Gabriel Bílý	New methods added: <ul style="list-style-type: none"> • Start Preauth Method • Get available preauthorizations • Start Preauth completion Method • Start Preauth increment Method • Start Preauth cancel Method • Get preauth transactions • Cancel transaction • Start Preauth Method with callback • Start Preauth completion Method with callback • Start Preauth increment Method with callback • Start Preauth cancel Method with callback
1.2	08.07.2021	Gabriel Bílý	Edited request endpoint for all methods
1.3	16.08.2021	Gabriel Bílý	<ul style="list-style-type: none"> • New Start MOTO Payment Method • New Daughter companies Method • The MerchantId parameter is replaced by daughterCompanyId
1.4	05.01.2022	Gabriel Bílý	<ul style="list-style-type: none"> • New transaction types PREAUTHORIZATION_INCREMENT and MOTO_SALE in TransactionType parameter ENUMS • Added information that for operations that require /confirm, that if this is not confirmed, an automatic reversal will be performed • Added request for Switchio Pay 3.0.0 and above for Start MOTO Payment Method • New methods Last Transaction, Get Successful Transactions • Complete revision of documentation
1.5	13.04.2022	Gabriel Bílý	<ul style="list-style-type: none"> • Added new Switchio Pay 3.1.0 – GSA V2 • Complete revision of documentation

Table 1 Version history

3. LIST OF TERMS / ABBREVIATIONS

GSA	Gapa Simple API
TRX	Transaction

4. PURPOSE OF DOCUMENT

The present document is intended for companies which implement the Gapa Simple Api (GSA) protocol to communicate with the MONET+ system.

The document is divided into 2 main parts, where we described following:

- Rest Api
- CashDesk

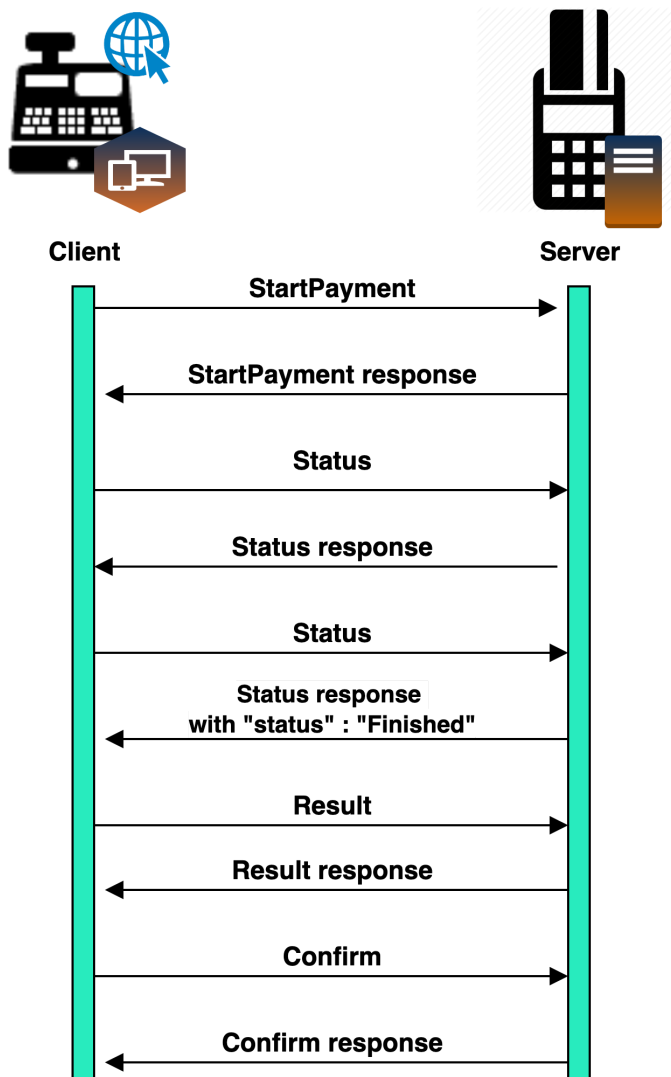
5. REST API (GSA version v1)

The Rest API enables interaction with the terminal through the internet and can be implemented on different platforms, such as desktop, android application, web etc. . The Implementation of the GSA is divided into to 2 modes:

- cash register is a client - terminal is a server
- cash register and terminal works both as client and server

5.1. Cash Register Client/Terminal Server

Example of payment workflow:



5.1.1. Info Method

The Info method purpose is to search up the local network and to find the terminal. If the response comes it's a terminal (most likely). The method sends the terminal ID in response.

Method: GET

Endpoint: ip:port/paya/info

Content-type: application/json

NOTE: Do not provide the body of request.

Response example:

JSON

```
{
  "protocol": "GSA",
  "version": 0,
  "terminalId": "TESTTERM01"
}
```

PARAMETR	TYPE	DESCRIPTION
protocol	string	Name of the used protocol
version	int	Version of the protocol
terminalId	string	ID of the terminal

Tab.1 *Parameters of the info method response*

5.1.2. Terminal Status Method

The Status method checks the terminal status and connection to the authorization server. The method also finds out time on the terminal.

Method: POST

Endpoint: ip:port/paya/terminal

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "checkOnline": true
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
checkOnline	bool	Information whether the test of connection to the authorization server is required

Tab. 2 Parameters of the status method request

Response example:

JSON

```
{
  "terminalId": "TESTTERM01",
  "dateTime": "1512469800000",
  "isOnline": 0,
  "onlineStatus": "Not requested"
}
```

PARAMETER	TYPE	DESCRIPTION
terminalId	string	ID of the terminal
dateTime	long	Date and time on terminal Format TBD.
isOnline	int	Status of the connection to the authorization server. 0 - wasn't required 1 - OK 2 - Error
onlineStatus	string	Message of the connection to the authorization server status.

Tab. 3 Parameters of the status method response

5.1.3. Start Payment Method

The Start Payment method triggers the payment. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/paya/payment

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-27a1c3290b6
amount	int	Amount of the payment, in cents 10053 is 100.53 euro
currencyCode	int	Code of the currency
invoiceNumber	string	Variable symbol length is 20 char
daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 4 Parameters of the payment method request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 5 Parameters of the payment method response

5.1.4. Start MOTO Payment Method

The Start MOTO Payment method triggers the MOTO payment. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/paya/payment/moto

Content-type: application/json

Request for Switchio Pay 2.5.0 and lower:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "cardNumber": "",
    "expiration": "0121",
    "cvc": "455",
    "amount": 10053,
    "currencyCode": 203,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
cardNumber	string	Card number
expiration	string	Card expiration, format MMY
cvc	string	Card Verification Code
amount	int	Amount of the payment, in cents 10053 is 100.53 euro
currencyCode	int	Code of the currency
invoiceNumber	string	Variable symbol length is 20 char
daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 6 Parameters of the MOTO payment method request

Request for Switchio Pay 3.0.0 and above:

CHANGE Request from Switchio Pay 3.0.0. It is not necessary to enter the card number, expiration and CVC - these values are entered by the user on the terminals. JSON is the same as for a normal payment.

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 203,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
amount	int	Amount of the payment, in cents 10053 is 100.53 euro
currencyCode	int	Code of the currency
invoiceNumber	string	Variable symbol length is 20 char
daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 7 Parameters of the MOTO payment method request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 8 Parameters of the MOTO payment response

5.1.5. Start Refund Method

The Refund method triggers the refund. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/paya/refund

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
amount	int	Amount of the refund, in cents 10053 is 100.53 euro
currencyCode	int	Code of the currency
invoiceNumber	string	Variable symbol, length is 20 char
daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 9 Parameters of the refund method request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the refund has started.
status	string	Message of the payment status.

Tab. 10 Parameters of the refund method response

5.1.6. Start Reverse Method

The Reversal method triggers the last payment return back to the customer's account.

Method: POST

Endpoint: ip:port/paya/reverse

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": ""

    "originalTransactionId": ""
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
originalTransactionId	string	ID of the original transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 11 Parameters of the reversal method request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the reversal has started.
status	string	Message of the payment status.

Tab. 12 Parameters of the reversal method response

5.1.7. Start Preauth Method

The Preauth method creates a new pre-authorisation transaction. Option to cancel the transaction with /cancel when obtaining the status with the value "WaitForCard". It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/paya/preauth

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227alc3290b6
amount	int	Amount of the refund, in cents 10053 is 100.53 euro
currencyCode	int	Code of the currency
invoiceNumber	string	Variable symbol, length is 20 char
daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 13 Parameters of the preauth method request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227alc3290b6
isStarted	bool	Information whether the preauth has started.
status	string	Message of the payment status.

Tab. 14 Parameters of the preauth method response

5.1.8. Get available preauthorizations

Returns a list of all available pre-authorisation to cancel or complete.

Method: GET

Endpoint: ip:port/paya/preauths

Content-type: application/json

Response example: TBD

5.1.9. Start Preauth completion Method

The Preauth completion method completes a preauth transaction that is identified by the originalTransactionId in the request. It is not possible to cancel with the /cancel endpoint. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/paya/preauth/complete

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": ""

"originalTransactionId": "",
    "finalAmount": 10053,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
originalTransactionId	string	ID of the original transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
finalAmount	int	Final amount of the preauth, in cents 10053 is 100.53 euro
invoiceNumber	string	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 15 Parameters of the preauth completion method request

Response example:

JSON

```
{  
  "transactionId": "",  
  "isStarted": false,  
  "status": "Other payment in  
progress"  
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the preauth completion has started.
status	string	Message of the payment status.

Tab. 16 Parameters of the preauth completion method response

5.1.10. Start Preauth increment Method

The Preauth increment method increments the amount for a preauth transaction that is identified by the originalTransactionId in the request. It is not possible to cancel with the /cancel endpoint.

Method: POST

Endpoint: ip:port/paya/preauth/increment

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": ""
  }
  "originalTransactionId": "",
  "incrementAmount": 10053,
  "invoiceNumber": "",
  "daughterCompanyId": ""
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
originalTransactionId	string	ID of the original transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
incrementAmount	int	Incremental amount of the preauth, in cents 10053 is 100.53 euro
invoiceNumber	string	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 17 Parameters of the preauth increment method request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
isStarted	bool	Information whether the preauth increment has started.
status	string	Message of the payment status.

Tab. 18 Parameters of the preauth increment method response

5.1.11. Start Preauth cancel Method

Cancels the preauth transaction that is identified by the originalTransactionId. It is not possible to cancel with the /cancel endpoint. It is NOT necessary to call /confirm after obtaining the Finished status.

Method: POST

Endpoint: ip:port/paya/preauth/cancel

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": ""
  },
  "originalTransactionId": "",
  "daughterCompanyId": ""
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
originalTransactionId	string	ID of the original transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 19 Parameters of the preauth cancel method request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the preauth cancel has started.
status	string	Message of the payment status.

Tab. 20 Parameters of the preauth cancel method response

5.1.12. Get preauth transactions

Get a list of available pre-authorisation to cancel or complete.

Method: POST

Endpoint: ip:port/paya/preauth/transactions

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "daughterCompanyId": ""
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
daughterCompanyId	string	ID of the daughter company. Default is "", this is the merchant identifier to be used to get preath

Tab. 21 *Parameters of the get preauth method request*

Response example:

Insert a result sheet or a blank sheet []

```
[{"transactionId":"a2334176-0c2a-4c7c-a0a8-e11d05413684","transactionType":"PREAUTHORIZATION","aid":"A0000000031010","amount":1200,"authCode":"057868","reversal":false,"counto":"VISA","currency":203,"dateTimeTerminal":"\\Feb 16, 2021 3:05:08 PM\\","dateTimeServer":"\\Feb 16, 2021 3:05:07 PM\\","emvResponseData":{"\\authorizationResponseCode\\":"00\\","smartCardScheme\\":1},"expiration":"****","invoiceNumber":"","responseCode":"OK","responseMessage":"potvrz eno","terminalIdAcquirer":"M1TNEXGO05","terminalIdIssuer":"M1TNEXGO05","merchantIdAcquirer":"TNEXGO05","merchantIdIssuer":"1357","pan":"451161****5652","panSequenceNumber":"02","sequenceNumber":"001002072","cvmTypeList":["NO"],"cardInputType":"CLESS"}]
```

5.1.13. Status Method

The Status method is called when we receive response of method:

- Start payment
- Start refund
- Start reverse
- Start closing

The purpose of the `Status` method is inquiring about the phase of the transaction, in regular intervals (3 seconds). Until the parameter `Status` from the `Status` method response contains *Finished*, then the `Result` method is called to obtain the result of the transaction.

Method: POST

Endpoint: ip:port/paya/status

Content-type: application/json

Request example:

JSON

```
{
  "transactionId": "",
  "secureString": "password"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
secureString	string	Password of the terminal

Tab. 22 Parameters of the Operation status method request

Response example:

JSON

```
{
  "transactionId": "",
  "status": "WaitForCard"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
status	string	Message of the Operation status. Enums of the status parameter are in the appendix section

Tab. 23 Parameters of the operation status method response

Following table summarizes possible response codes.

RESPONSE CODE	DESCRIPTION
200	<i>transactionId</i> is recognized by the terminal
400	invalid data
401	<i>secureString</i> does not match
404	<i>transactionId</i> is not recognized by terminal

Tab. 24 Response codes of the Operation Status method

5.1.14. Result Method

The Result method calls for the result of the payment. If the payment is still in progress, it returns progress information.

NOTE: Calling `/result` and retrieving the result is not considered complete completion and confirmation of the transaction if the transaction requires `/confirm`. If the endpoint `/confirm` is not called for a transaction that requires confirmation, an automatic reversal is performed.

Method: POST

Endpoint: `ip:port/paya/result`

Content-type: `application/json`

Request example:

JSON

```
{
  "transactionId": "",
  "secureString": "password"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <code>1e6a1251-762f-45fc-85fb-227a1c3290b6</code>
secureString	string	Password of the terminal

Tab. 25 Parameters of the Result method request

Response example, when the payment is in progress:

JSON

```
{
  "aid": "A00000000283101",
  "amount": 500,
  "aosa": "null",
  "appLabel": "Air Bank",
  "authCode": "114546",
  "availableBalance": "1000",
  "callReversal": false,
  "cardInputType": "CLESS",
  "conto": "MASTERCARD",
}
```

```

    "currency": 203,
    "cvmTypeList": ["PIN_ONLINE"],
    "dateTimeServer": "Jul 16, 2020 14:40:50",
    "dateTimeTerminal": "Jul 16, 2020 14:39:37",
    "emvResponseData": {
      "smartCardScheme": "1",
      "authorisationResponseCode": "00"
    },
    "expiration": "****",
    "invoiceNumber": "VS",
    "isReversed": false,
    "merchantIdAcquirer": "TNEXGO01",
    "merchantIdIssuer": "1357",
    "pan": "970348*****3910",
    "panSequenceNumber": "00",
    "responseCode": "OK",
    "responseMessage": "potvrzeno",
    "reversal": false,
    "sequenceNumber": "001001614",
    "terminalIdAcquirer": "M1TNEXGO01",
    "terminalIdIssuer": "M1TNEXGO01",
    "transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
    "transactionType": "SALE_ONLINE"
  }

```

PARAMETER	TYPE	DESCRIPTION
aid	string	Application identifier
amount	int	Amount of the transaction in cents. 500 is 5.00 euro
aosa	string	Available offline sending amount
appLabel	string	Label of application
authCode	int	Code of the authorization
availableBalance	int	Balance of the available funds
callReversal	bool	Information whether the reversal should be called or not.
cardInputType	string	Type of the input. Enum is in the appendix .
conto	string	Card issuer
currency	int	Code of the currency
cvmTypeList	list	Type of the TRX. Enum is in the appendix .
dateTimeServer	string	Date and time on server when, was the result made
dateTimeTerminal	string	Date and time on terminal, when was the result made
emvResponseData	obj	Object of the response.
smartCardScheme	int	Scheme of the smart card

authorisationResponseCode	string	The response code of authorization.
expiration	string	Expiration date of the card.
invoiceNumber	string	Variable symbol, max length is 20 char
isReversed	bool	Information whether the transaction was reversed.
merchantIdAcquirer	string	ID of the merchant's acquirer.
merchantIdIssuer	int	ID of the merchant's issuer
pan	int	Primary account number
panSequenceNumber	int	Sequence number of the primary account number
responseCode	string	Code of the response
responseMessage	string	Message of the response
reversal	bool	Information
sequenceNumber	int	Number of the sequence
terminalIdAcquirer	string	ID of the acquirer's terminal
terminalIdIssuer	string	ID of the issuer's terminal
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
transactionType	string	Type of the transaction. Enum is in the appendix .

Tab. 26 Parameters of the Result method response

Response example, after the start closing method is processed:

JSON

```
{
  "batchTotals": {
    "adjsAmount":0.00,
    "adjsCount":0,
    "batchNumber":11,
    "creditsAmount":0.00,
    "creditsCount":0,
    "debitsAmount":0.00,
    "debitsCount":0,
    "shiftNumber":1
  },
  "dateTimeServer":"Jul 9, 2020 10:31:38 AM",
  "dateTimeTerminal":"Jul 9, 2020 10:31:50 AM",
  "responseCode":"OK",
  "responseMessage":"Soucty sum souhlasi",
  "transactionType":"CLOSE_BATCH"
}
```

PARAMETER	TYPE	DESCRIPTION
batchTotals	obj	Object of the batch totals
adjsAmount	int	Amount of adjustments
adjsCount	int	Count of adjustments
batchNumber	int	Number of batch
creditsAmount	int	Amount of refund
creditsCount	int	Count of refund TRX
debitsAmount	int	Amount of payment TRX
debitsCount	int	Count of payment TRX
shiftNumber	int	number of shift
dateTimeServer	string	Date and time on server when, was the result made
dateTimeTerminal	string	Date and time on terminal, when was the result made
responseCode	string	Code of the response
responseMessage	string	Message of the response
transactionType	string	Type of the TRX. Enum is in the appendix .

Tab. 27 Parameters of the result method response when the Start closing is processed

5.1.15. Confirm Method

The Confirm method obtains information whether the payment is confirmed or not. The cash register must explicitly confirm the payment. The confirmation means that the customer signature matches and the receipt is printed.

NOTE: We can consider the payment is confirmed only when the terminal responds with parameter *isConfirm* : *true*. Upon expiration of the time period (1 minute), an automatic reversal of the unacknowledged transaction that required confirmation will be performed.

Method: POST

Endpoint: ip:port/paya/confirm

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "transactionId": "",
  "confirm": true
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
confirm	bool	Information whether is transaction confirmed or not

Tab. 28 Parameters of the Confirm method request

Response example:

JSON

```
{
  "transactionId": "",
  "isConfirmed": true
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
isConfirmed	bool	information whether the transaction is confirmed or not.

Tab. 29 Parameters of the Confirm method response

5.1.16. Start Closing Method

The Closing method calls to close the transaction.

Method: POST

Endpoint: ip:port/paya/closing

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 30 Parameters of the Start Closing method request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the transaction started or not.
status	string	Message of the transaction status

Tab. 31 Parameters of the Start Closing method response

5.1.17. Cancel transaction

Payment can only be canceled if the status "WaitForCard" occurs during progress, otherwise this request is rejected.

Calling is allowed for: payment (/payment, /cbpayment), preauth (/preauth, /cbpreauth), refund (/refund, /cbrefund)

Method: POST

Endpoint: ip:port/paya/cancel

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "transactionId": ""
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6

Tab. 32 Parameters of the cancel transaction request

Response example:

Following table summarizes possible response codes.

RESPONSE CODE	DESCRIPTION
200	The transaction is canceled, the client expects the status "Finished" in /progress request
404	The transaction with the specified transactionId was not found
405	The method is not allowed in its current state

Tab. 33 Response codes of the Cancel transaction

5.1.18. Daughter companies

Obtaining available merchants for a given terminal. Supported in version 3.0.0.

Method: POST

Endpoint: ip:port/paya/daughter_companies

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password"
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal

Tab. 34 Parameters of the Daughter companies method request

Response example:

JSON

```
[
  { "daughterCompanyId": "Kaufland" },
  { "daughterCompanyId": "Tesco" }
]
```

PARAMETER	TYPE	DESCRIPTION
daughterCompanyId	string	ID of the daughter company

Tab. 35 Parameters of the Daughter companies method response

5.1.19. Last transaction

Getting the last transaction, if no transaction is available in response, a 404 will be sent. Supported in version 3.0.0.

Method: POST

Endpoint: ip:port/paya/last_transaction

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password"
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal

Tab. 36 Parameters of the Last transaction method request

Response example:

JSON

```
{
  "aid": "A0000000283101",
  "amount": 500,
  "aosa": "null",
  "appLabel": "Air Bank",
  "authCode": "114546",
  "availableBalance": "1000",
  "callReversal": false,
  "cardInputType": "CLESS",
  "conto": "MASTERCARD",
  "currency": 203,
  "cvmTypeList": ["PIN_ONLINE"],
  "dateTimeServer": "Jul 16, 2020
14:40:50",
  "dateTimeTerminal": "Jul 16,
2020 14:39:37",
  "emvResponseData": {
    "smartCardScheme": "1",
    "authorisationResponseCode":
"00"
  },
  "expiration": "****",
  "invoiceNumber": "VS",
  "isReversed": false,
  "merchantIdAcquirer":
"TNEXGO01",
  "merchantIdIssuer": "1357",
  "pan": "970348*****3910",
  "panSequenceNumber": "00",
  "responseCode": "OK",
  "responseMessage": "potvrzeno",
  "reversal": false,
  "sequenceNumber": "001001614",
  "terminalIdAcquirer":
"MTNEXGO01",
  "terminalIdIssuer":
"MTNEXGO01",
  "transactionId":
"48a3ea7d-b82f-4b42-abbb-5043365e4fa
a",
  "transactionType": "SALE_ONLINE"
}
```

PARAMETER	TYPE	DESCRIPTION
daughterCompanyId	string	ID of the daughter company

Tab. 37 Parameters of the Last transaction method response

PARAMETER	TYPE	DESCRIPTION
aid	string	Application identifier
amount	int	Amount of the transaction in cents. 500 is 5.00 euro
aosa	string	Available offline sending amount

appLabel	string	Label of application
authCode	int	Code of the authorization
availableBalance	int	Balance of the available funds
callReversal	bool	Information whether the reversal should be called or not.
cardInputType	string	Type of the input. Enum is in the appendix .
conto	string	Card issuer
currency	int	Code of the currency
cvmTypeList	list	Type of the TRX. Enum is in the appendix .
dateTimeServer	string	Date and time on server when, was the result made
dateTimeTerminal	string	Date and time on terminal, when was the result made
emvResponseData	obj	Object of the response.
	smartCardScheme	int
	authorisationResponseCode	string
expiration	string	Expiration date of the card.
invoiceNumber	string	Variable symbol, max length is 20 char
isReversed	bool	Information whether the transaction was reversed.
merchantIdAcquirer	string	ID of the merchant's acquirer.
merchantIdIssuer	int	ID of the merchant's issuer
pan	int	Primary account number
panSequenceNumber	int	Sequence number of the primary account number
responseCode	string	Code of the response
responseMessage	string	Message of the response
reversal	bool	Information
sequenceNumber	int	Number of the sequence
terminalIdAcquirer	string	ID of the acquirer's terminal
terminalIdIssuer	string	ID of the issuer's terminal
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
transactionType	string	Type of the transaction. Enum is in the appendix .

Tab. 38 *Parameters of the Last transaction response*

5.1.20. Get successful transactions

Obtain all successful transactions. Supported in version 3.0.0.

Method: POST

Endpoint: ip:port/paya/transactions

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password"
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal

Tab. 39 Parameters of the Get successful transactions method request

Response example:

JSON

```
[
  {
    "aid": "A0000000283101",
    "amount": 500,
    "aosa": "null",
    "appLabel": "Air Bank",
    "authCode": "114546",
    "availableBalance": "1000",
    "callReversal": false,
    "cardInputType": "CLESS",
    "conto": "MASTERCARD",
    "currency": 203,
    "cvmTypeList": ["PIN_ONLINE"],
    "dateTimeServer": "Jul 16, 2020 14:40:50",
    "dateTimeTerminal": "Jul 16, 2020 14:39:37",
    "emvResponseData": {
      "smartCardScheme": "1",
      "authorisationResponseCode": "00"
    },
    "expiration": "****",
    "invoiceNumber": "VS",
    "isReversed": false,
    "merchantIdAcquirer": "TNEXGO01",
    "merchantIdIssuer": "1357",
    "pan": "970348*****3910",
    "panSequenceNumber": "00",
    "responseCode": "OK",
    "responseMessage": "potvrzeno",
    "reversal": false,
  }
]
```

PARAMETER	TYPE	DESCRIPTION
daughterCompanyId	string	ID of the daughter company

Tab. 40 Parameters of the Get successful transactions method response

```

    "sequenceNumber": "001001614",
    "terminalIdAcquirer":
    "MITNEXGO01",
    "terminalIdIssuer":
    "MITNEXGO01",
    "transactionId":
    "48a3ea7d-b82f-4b42-abbb-5043365e4fa
    a",
    "transactionType": "SALE_ONLINE"
  }
]

```

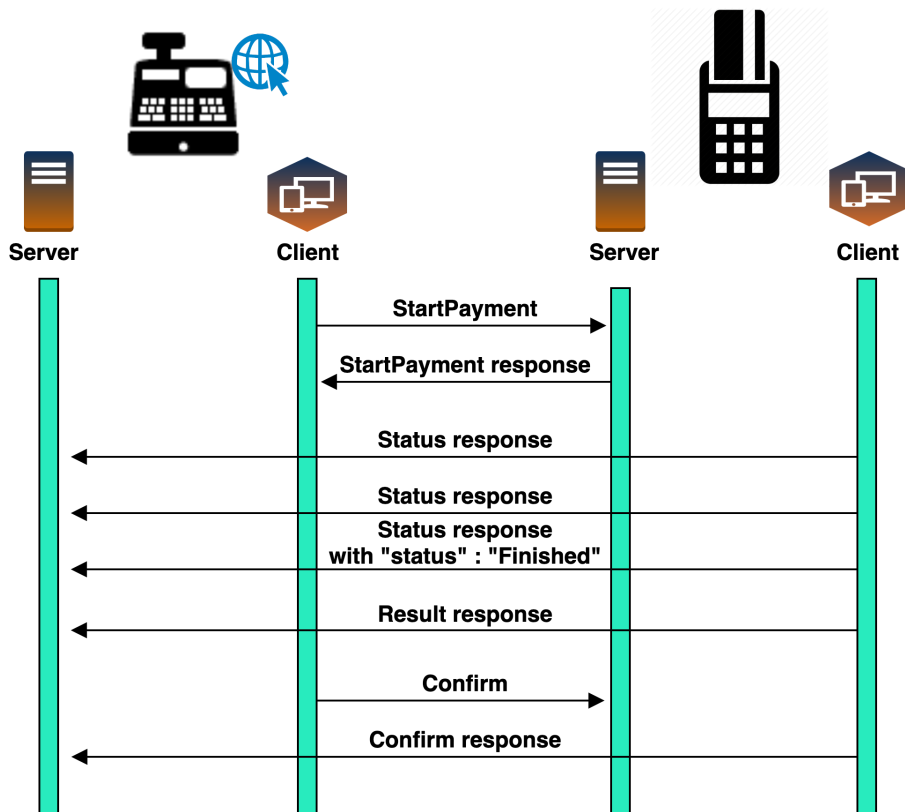
PARAMETER	TYPE	DESCRIPTION
aid	string	Application identifier
amount	int	Amount of the transaction in cents. 500 is 5.00 euro
aosa	string	Available offline sending amount
appLabel	string	Label of application
authCode	int	Code of the authorization
availableBalance	int	Balance of the available funds
callReversal	bool	Information whether the reversal should be called or not.
cardInputType	string	Type of the input. Enum is in the appendix .
conto	string	Card issuer
currency	int	Code of the currency
cvmTypeList	list	Type of the TRX. Enum is in the appendix .
dateTimeServer	string	Date and time on server when, was the result made
dateTimeTerminal	string	Date and time on terminal, when was the result made
emvResponseData	obj	Object of the response.
smartCardScheme	int	Scheme of the smart card
authorisationResponseCode	string	The response code of authorization.
expiration	string	Expiration date of the card.
invoiceNumber	string	Variable symbol, max length is 20 char
isReversed	bool	Information whether the transaction was reversed.
merchantIdAcquirer	string	ID of the merchant's acquirer.
merchantIdIssuer	int	ID of the merchant's issuer

pan	int	Primary account number
panSequenceNumber	int	Sequence number of the primary account number
responseCode	string	Code of the response
responseMessage	string	Message of the response
reversal	bool	Information
sequenceNumber	int	Number of the sequence
terminalIdAcquirer	string	ID of the acquirer's terminal
terminalIdIssuer	string	ID of the issuer's terminal
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
transactionType	string	Type of the transaction. Enum is in the appendix .

Tab. 41 Parameters of the Get successful transactions response

5.2. Cash register and terminal work as both client and server

Example of payment workflow:



5.2.1. Start Payment Method with callback

The Start Payment method triggers the payment. In the cash register and terminal works as both client and server mode we use callback within this method. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/paya/cbpayment

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
amount	int	Amount of the payment
currencyCode	int	Currency of the payment amount
invoiceNumber	string	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 42 Parameters of the Start payment method with callback request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 43 Parameters of the Start payment method with callback response

5.2.2. Start MOTO Payment Method with callback

The Start MOTO Payment method triggers the MOTO payment. In the cash register and terminal works as both client and server mode we use callback within this method. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: POST ip:port/paya/cbpayment/moto

Content-type: application/json

Request for Switchio Pay 2.5.0 and lower:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "cardNumber": "",
    "expiration": "0121",
    "cvc": "455",
    "amount": 10053,
    "currencyCode": 203,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
cardNumber	string	Card number
expiration	string	Card expiration, format MMY
cvc	string	Card Verification Code
amount	int	Amount of the payment
currencyCode	int	Currency of the payment amount
invoiceNumber	int	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of progress to callback
ip	string	IP of cash register to callback
port	int	Port of cash register to callback
endpoint	string	Endpoint of method to callback
resultCallback	obj	Object of result to callback
ip	string	IP of cash register to callback
port	int	Port of cash register to callback
endpoint	string	Endpoint of the method to callback

Tab. 44 Parameters of the Start MOTO payment method with callback request

Response example:

JSON

```
{  
  "transactionId": "",  
  "isStarted": false,  
  "status": "Other payment in  
progress"  
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85f b-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 45 Parameters of the Start MOTO payment
method with callback response

CHANGE Request from Switchio Pay 3.0.0. It is not necessary to enter the card number, expiration and CVC - these values are entered by the user on the terminals. JSON is the same as for a normal payment.

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 203,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER		TYPE	DESCRIPTION
secureString		string	Password of the terminal
request		obj	Object of the request
	transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
	amount	int	Amount of the payment
	currencyCode	int	Currency of the payment amount
	invoiceNumber	int	Variable symbol, length is 20 char
	daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback		obj	Object of progress to callback
	ip	string	IP of cash register to callback
	port	int	Port of cash register to callback
	endpoint	string	Endpoint of method to callback
resultCallback		obj	Object of result to callback
	ip	string	IP of cash register to callback
	port	int	Port of cash register to callback
	endpoint	string	Endpoint of the method to callback

Tab. 46 Parameters of the Start MOTO payment method with callback request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 47 Parameters of the Start payment method with callback response

5.2.3. Start Refund Method with callback

The Start Refund method triggers the refund. In the cash register and terminal works as both client and server mode we use callback within this method. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/paya/cbrefund

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
amount	int	Amount of the refund
currencyCode	int	Currency of the refund
invoiceNumber	string	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of progress to callback
ip	string	IP to callback

```
}
```

port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 48 Parameters of the Start refund method with callback request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the refund has started.
status	string	Message of the payment status.

Tab. 49 Parameters of the Start refund method with callback response

5.2.4. Start Reverse Method with callback

The Start Reverse method triggers the reversal. In the cash register and terminal works as both client and server mode we use callback within this method.

Method: POST

Endpoint: ip:port/paya/cbreverse

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",

```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6

```

"originalTransactionId":""
  "daughterCompanyId":""
},
"progressCallback":{
  "ip":"172.25.11.14",
  "port":7777,

"endpoint":"/gsa/progress"
},
"resultCallback":{
  "ip":"172.25.11.14",
  "port":7777,
  "endpoint":"/gsa/result"
}
}

```

originalTransactionId	string	ID of original transaction what is being reversed
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 50 Parameters of the start reverse method with callback request

Response example:

JSON

```

{
  "transactionId":"","
  "isStarted":false,
  "status":"Other payment in progress"
}

```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
isStarted	bool	Information whether the refund has started.
status	string	Message of the payment status.

Tab. 51 Parameters of the start reverse method with callback response

5.2.5. Start Preauth Method with callback

The Start Preauth method creates a new pre-authorization transaction. Option to cancel the transaction with /cancel when obtaining the status with the value "WaitForCard". It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/paya/cbpreauth

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
amount	int	Amount of the preauth
currencyCode	int	Currency of the preauth
invoiceNumber	string	Variable symbol, length is 20 char
daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 52 Parameters of the start preauth method with callback request

Response example:

JSON

```
{  
  "transactionId": "",  
  "isStarted": false,  
  "status": "Other payment in  
progress"  
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
isStarted	bool	Information whether the preauth has started.
status	string	Message of the payment status.

Tab. 53 Parameters of the start preauth method with callback response

5.2.6. Start Preauth completion Method with callback

The Start Preauth completion method completes a preauth transaction that is identified by originalTransactionId. It is not possible to cancel with the /cancel endpoint. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/paya/cbpreauth/complete

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "originalTransactionId": "",
    "finalAmount": 10053,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227alc3290b6
originalTransactionId	string	ID of the original transaction. Example: 1e6a1251-762f-45fc-85fb-227alc3290b6
finalAmount	int	Final amount of the preauth, in cents 10053 is 100.53 euro
invoiceNumber	int	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 54 Parameters of the start preauth completion method with callback request

Response example:

JSON

```
{  
  "transactionId": "",  
  "isStarted": false,  
  "status": "Other payment in  
progress"  
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the preauth completion has started.
status	string	Message of the payment status.

Tab. 55 Parameters of the start preauth completion method with callback response

5.2.7. Start Preauth increment Method with callback

The Start Preauth increment method increments the amount for a preauth transaction that is identified by the originalTransactionId in the request. It is not possible to cancel with the /cancel endpoint.

Method: POST

Endpoint: ip:port/paya/cbpreauth/increment

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "originalTransactionId": "",
    "incrementAmount": 10053,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
originalTransactionId	string	ID of the original transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
incrementAmount	int	Incremental amount of the preauth, in cents 10053 is 100.53 euro
invoiceNumber	string	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 56 Parameters of the start preauth increment method with callback request

Response example:

JSON

```
{  
  "transactionId": "",  
  "isStarted": false,  
  "status": "Other payment in  
progress"  
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the preauth increment has started.
status	string	Message of the payment status.

Tab. 57 Parameters of the start preauth increment method with callback response

5.2.8. Start Preauth cancel Method with callback

The Start Preauth cancel method cancels the preauth transaction that is identified by the originalTransactionId. It is not possible to cancel with the /cancel endpoint. It is NOT necessary to call /confirm after obtaining the Finished status.

Method: POST

Endpoint: ip:port/paya/cbpreauth/cancel

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "originalTransactionId": "",
    "daughterCompanyId": "",
    "progressCallback": {
      "ip": "172.25.11.14",
      "port": 7777,
      "endpoint": "/gsa/progress"
    },
    "resultCallback": {
      "ip": "172.25.11.14",
      "port": 7777,
      "endpoint": "/gsa/result"
    }
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
originalTransactionId	string	ID of the original transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 58 Parameters of the start preauth cancel method with callback request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the preauth cancel has started.
status	string	Message of the payment status.

Tab. 59 Parameters of the start preauth cancel method with callback response

5.2.9. Start Closing Method with callback

The Start Closing method triggers the closing. In the cash register and terminal works as both client and server mode we use callback within this method.

Method: POST

Endpoint: ip:port/paya/cbclosing

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback

endpoint	string	Endpoint of the method to callback
----------	--------	------------------------------------

Tab. 60 Parameters of the Start closing method with callback req.

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
isStarted	bool	Information whether the refund has started.
status	string	Message of the payment status.

Tab. 61 Parameters of the Start closing method response

5.2.10. Info Method

The Info method is the same as the [Info method](#) in the Cash register Client/ terminal server section.

5.2.11. Terminal Status Method

The Terminal Status method is the same as the [Terminal Status method](#) in the Cash register Client/ terminal server section.

5.2.12. Status Method

The Status method is the same as the [Status method](#) in the Cash register Client/ terminal server section.

5.2.13. Result Method

The Result method is the same as the [Result method](#) in the Cash register Client/ terminal server section.

5.2.14. Confirm Method

The Confirm method is the same as the [Confirm method](#) in the Cash register Client/ terminal server section.

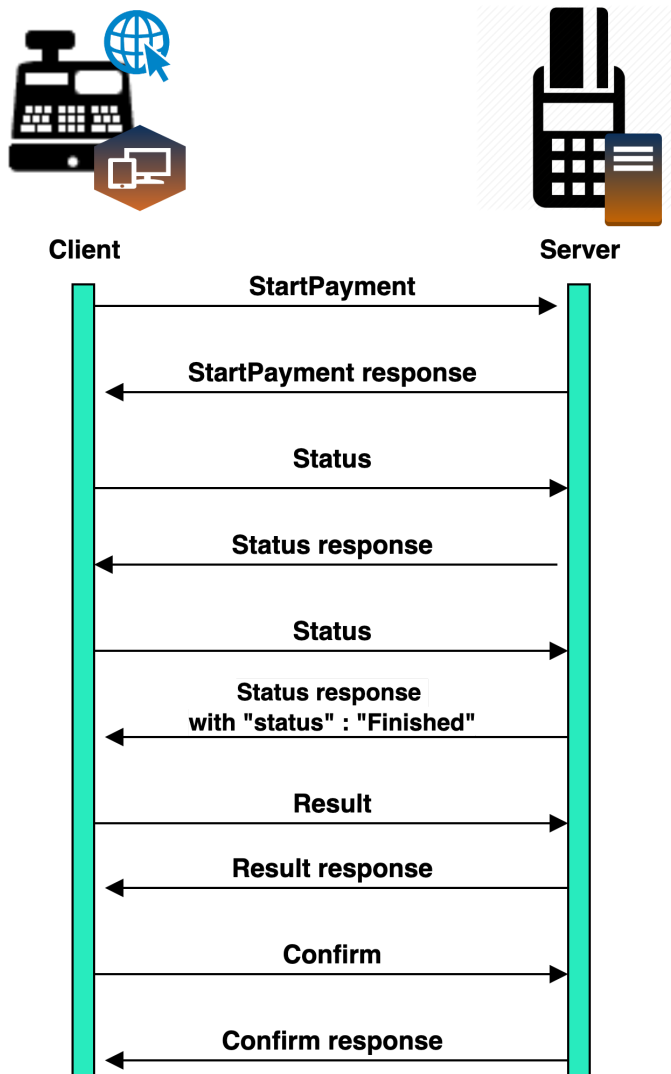
6. Switchio Pay 3.1.0 – GSA V2

Supported api from version 3.1.0:

- New version of endpoints starting with **ip:port/api/switchio/pay/v2/***, which replaces **ip:port/paya/***, example: to the endpoint of the first version: **ip:port/paya/payment**, an alternative endpoint is created: **ip:port/api/switchio/pay/v2/payment**
 - Old **ip:port/paya/*** endpoints will still be supported.
- Add a token field to the response for transactions.
- Convert responses to the JSON standard that did not conform to this standard

6.1. Cash Register Client/Terminal Server

Example of payment workflow:



6.1.1. Get transaction status

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/transaction_status

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "transactionId": ""
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
transactionId	string	ID of the transaction. Example: 48a3ea7d-b82f-4b42-abbb-5043365e4faa

Tab. 62 Parameters of the Get transaction status request

Response example:

JSON

```
{
  "isStarted": true,
  "transactionId": "",
  "status": "OK"
}
```

PARAMETER	TYPE	DESCRIPTION
isStarted	bool	Information whether the payment has started.
transactionId	string	ID of the transaction. Example: 48a3ea7d-b82f-4b42-abbb-5043365e4faa
status	string	Message of the payment status.

Tab. 63 Parameters of the Get transaction status response

6.1.2. Info V2

The Info method purpose is to search up the local network and to find the terminal. If the response comes it's a terminal (most likely).

Method: GET

Endpoint: ip:port/api/switchio/pay/v2/info

Content-type: application/json

Response example:

JSON

```
{
  "protocol": "GSA",
  "version": 0,
  "terminalId": "TESTTERM01"
}
```

PARAMETR	TYPE	DESCRIPTION
protocol	string	Name of the used protocol
version	int	Version of the protocol
terminalId	string	ID of the terminal

Tab. 64 Parameters of the Info V2 response

6.1.3. Status V2

The Status method checks the terminal status and connection to the authorization server. The method also finds out time on the terminal.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/terminal

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "checkOnline": true
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
checkOnline	bool	Information whether the test of connection to the authorization server is required

Tab. 65 Parameters of the Status V2 request

Response example:

JSON

```
{
  "terminalId": "TESTTERM01",
  "dateTime": "1512469800000",
  "isOnline": 0,
  "onlineStatus": "Not requested"
}
```

PARAMETER	TYPE	DESCRIPTION
terminalId	string	ID of the terminal
dateTime	long	Date and time on terminal Format TBD.
isOnline	int	Status of the connection to the authorization server. 0 - wasn't required 1 - OK 2 - Error
onlineStatus	string	Message of the connection to the authorization server status.

Tab. 66 Parameters of the Status V2 response

6.1.4. Payment polling V2

The Start Payment method triggers the payment. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/payment

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 203,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
amount	int	Amount of the payment, in cents 10053 is 100.53 euro
currencyCode	int	Code of the currency
invoiceNumber	string	Variable symbol length is 20 char
daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 67 Parameters of the Payment polling V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started – server busy

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Response - Closing

```
{
  "message": "Already processed"
}
```


JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

```
{
  "message": "Already processed"
}
```

6.1.6. Refund polling V2

The Refund method triggers the refund. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/refund

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
amount	int	Amount of the refund, in cents 10053 is 100.53 euro
currencyCode	int	Code of the currency
invoiceNumber	string	Variable symbol, length is 20 char
daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 71 Parameters of the Refund polling V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in
progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 72 Parameters of the Refund polling V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed

Response - Closing

```
{
  "message": "Already processed"
}
```

6.1.7. Reversal polling V2

The Reversal method triggers the last payment return back to the customer's account.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/reverse

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": ""

"originalTransactionId": ""
    "daughterCompanyId": ""
  }
}
```

PARAMETER		TYPE	DESCRIPTION
secureString		string	Password of the terminal
request		obj	Request object
	transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
	originalTransactionId	string	ID of the original transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
	daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 73 *Parameters of the Reversal polling V2 request*

Response

HTTP Status code

- **Status Code: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **Status Code: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in
progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 74 Parameters of the Reversal polling V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed
- **StatusCode: 404** → The transaction to be reversed was not found in the database

Response – Closing

```
{
  "message": "Already processed"
}
```

6.1.8. Preauthorization polling V2

The Preauth method creates a new pre-authorisation transaction. Option to cancel the transaction with /cancel when obtaining the status with the value "WaitForCard". It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/preauth

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": ""
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER		TYPE	DESCRIPTION
secureString		string	Password of the terminal
request		obj	Request object
	transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
	amount	int	Amount of the refund, in cents 10053 is 100.53 euro
	currencyCode	int	Code of the currency
	invoiceNumber	string	Variable symbol, length is 20 char
	daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 75 *Parameters of the preauth method request*

Response

HTTP Status code

- **Status Code: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **Status Code: 202** → the request was processed but the transaction was not started – server busy

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

```
{
  "message": "Already processed"
}
```

6.1.9. PreAuth Complete – polling V2

The Preauth completion method completes a preauth transaction that is identified by the originalTransactionId in the request. It is not possible to cancel with the /cancel endpoint. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/preauth/complete

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": ""
  },
  "originalTransactionId": "",
  "finalAmount": 10053,
  "invoiceNumber": "",
  "daughterCompanyId": ""
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
originalTransactionId	string	ID of the original transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
finalAmount	int	Final amount of the preauth, in cents 10053 is 100.53 euro
invoiceNumber	string	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 77 Parameters of the PreAuth Complete – polling V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 78 Parameters of the PreAuth Complete - polling V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed
- **StatusCode: 404** → Transaction with originalTransactionId not found

Response - Closing

```
{
  "message": "Already processed"
}
```


Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in
progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 80 Parameters of the PreAuth Increment - polling V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed
- **StatusCode: 404** → Transaction with originalTransactionId not found

Response - Closing

```
{
  "message": "Already processed"
}
```

6.1.11. PreAuth Cancel – polling V2

Cancels the preauth transaction that is identified by the originalTransactionId. It is not possible to cancel with the /cancel endpoint. It is NOT necessary to call /confirm after obtaining the Finished status.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/preauth/cancel

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": ""
  },
  "originalTransactionId": "",
  "daughterCompanyId": ""
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Request object
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
originalTransactionId	string	ID of the original transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 81 Parameters of the PreAuth Cancel – polling V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started – server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 82 Parameters of the PreAuth Cancel - polling V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed
- **StatusCode: 404** → Transaction with originalTransactionId not found

Response - Closing

```
{
  "message": "Already processed"
}
```

6.1.12. Get available preauthorizations V2

Get a list of available pre-authorisation to cancel or complete.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/preauth/transactions

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "daughterCompanyId": ""
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
daughterCompanyId	string	ID of the daughter company. Default is "", this is the merchant identifier to be used to get preauth

Tab. 83 Parameters of the Get available preauthorizations V2 request

Response example:

Returns a list of results TransactionResultV2 data structure or a blank sheet []

```
[
  {
    "transactionId": "a2334176-0c2a-4c7c-a0a8-e11d05413684",
    "transactionType": "PREAUTHORIZATION",
    "aid": "A0000000031010",
    "amount": 1200,
    "authCode": "057868",
    "reversal": false,
    "conto": "VISA",
    "currency": 203,
    "dateTimeTerminal": "\"Feb 16, 2021 3:05:08 PM\"",
    "dateTimeServer": "\"Feb 16, 2021 3:05:07 PM\"",
    "emvResponseData": "\"{\\\"authorisationResponseCode\\\":\\\"00\\\",\\\"smartCardScheme\\\":1}\"",
    "expiration": "****",
    "invoiceNumber": "",
    "responseCode": "OK",
    "responseMessage": "potvrzeno",
    "terminalIdAcquirer": "M1TNEXGO05",
    "terminalIdIssuer": "M1TNEXGO05",
    "merchantIdAcquirer": "TNEXGO05",
    "merchantIdIssuer": "1357",
    "pan": "451161*****5652",
    "panSequenceNumber": "02",
    "sequenceNumber": "001002072",
    "cvmTypeList": ["NO"],
    "cardInputType": "CLESS"
  }
]
```


6.1.13. Operation status V2

The Status method is called when we receive response of method:

- Start payment
- Start refund
- Start reverse
- Start closing

The purpose of the Status method is inquiring about the phase of the transaction, in regular intervals (3 seconds). Until the parameter [Status](#) from the Status method response contains *Finished*, then the Result method is called to obtain the result of the transaction.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/status

Content-type: application/json

Request example:

JSON

```
{
  "transactionId": "",
  "secureString": "password"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
secureString	string	Password of the terminal

Tab. 84 Parameters of the Operation status V2 request

Response example:

JSON

```
{
  "transactionId": "",
  "status": "WaitForCard"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
status	string	Message of the Operation status. Enums of the status parameter are in the appendix section

Tab. 85 *Parameters of the Operation status V2 response*

Following table summarizes possible response codes.

RESPONSE CODE	DESCRIPTION
200	<i>transactionId</i> is recognized by the terminal
400	Invalid data
401	Unauthorized
404	<i>transactionId</i> is not recognized by terminal

Tab. 86 Response codes of the Operation status V2

Response - Closing

```
{
  "message": "Already processed"
}
```

6.1.14. Confirm transaction V2

The Confirm method obtains information whether the payment is confirmed or not. The cash register must explicitly confirm the payment. The confirmation means that the customer signature matches and the receipt is printed.

NOTE: We can consider the payment is confirmed only when the terminal responds with parameter *isConfirm* : *true*. Upon expiration of the time period (1 minute), an automatic reversal of the unacknowledged transaction that required confirmation will be performed.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/confirm

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "transactionId": "",
  "confirm": true
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
confirm	bool	Information whether is transaction confirmed or not

Tab. 87 Parameters of the Confirm transaction V2 request

Response example:

Status Code: 200

JSON

```
{
  "transactionId": "",
  "isConfirmed": true
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isConfirmed	bool	information whether the transaction is confirmed or not.

Tab. 88 Parameters of the Confirm transaction V2 response

Status Code: 404 → Transaction not found

Response - Closing

```
{
  "message": "Not found"
}
```

6.1.15. Closing polling V2

The Closing method calls to close the transaction.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/closing

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "daughterCompanyId": ""
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.

Tab. 89 Parameters of the Closing polling V2 request

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the transaction started or not.
status	string	Message of the transaction status

Tab. 90 Parameters of the Closing polling V2 response

6.1.16. Cancel transaction V2

Payment can only be canceled if the status "WaitForCard" occurs during progress, otherwise this request is rejected.

Calling is allowed for: payment (/payment, /cbpayment), preauth (/preauth, /cbpreauth), refund (/refund, /cbrefund)

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cancel

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "transactionId": ""
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6

Tab. 91 Parameters of the Cancel transaction V2 request

Response example:

Following table summarizes possible response codes.

RESPONSE CODE	DESCRIPTION
200	The transaction is canceled, the client expects the status "Finished" in /progress request
404	The transaction with the specified transactionId was not found
405	The method is not allowed in its current state

Tab. 92 Response codes of the Cancel transaction

Response - Closing

```
{
  "message": "OK"
}
```

6.1.17. Daughter companies V2

Obtaining available merchants for a given terminal. Supported in version 3.0.0.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/daughter_companies

Content-type: application/json

Request example:

JSON

```
[
  {
    "secureString": "password"
  }
]
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal

Tab. 93 Parameters of the Daughter companies V2 request

Response example:

JSON

```
[
  { "daughterCompanyId": "Kaufland" },
  { "daughterCompanyId": "Tesco" }
]
```

PARAMETER	TYPE	DESCRIPTION
daughterCompanyId	string	ID of the daughter company

Tab. 94 Parameters of the Daughter companies V2 response

6.1.18. Last transaction V2

Getting the last transaction, if no transaction is available in response, a 404 will be sent.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/last_transaction

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password"
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal

Tab. 95 *Parameters of the Last transaction V2 request*

Response example:

HTTP StatusCode: 200 → transaction found successfully

JSON

```
{
  "aid": "A0000000283101",
  "amount": 500,
  "aosa": "null",
  "appLabel": "Air Bank",
  "authCode": "114546",
  "availableBalance": "1000",
  "callReversal": false,
  "cardInputType": "CLESS",
  "conto": "MASTERCARD",
  "currency": 203,
  "cvmTypeList": ["PIN_ONLINE"],
  "dateTimeServer": "Jul 16, 2020
14:40:50",
  "dateTimeTerminal": "Jul 16,
2020 14:39:37",
  "emvResponseData": {
    "smartCardScheme": "1",
    "authorisationResponseCode": "00"
  },
  "expiration": "****",
  "invoiceNumber": "VS",
  "isReversed": false,
  "merchantIdAcquirer": "TNEXGO01",
  "merchantIdIssuer": "1357",
  "pan": "970348*****3910",
  "panSequenceNumber": "00",
  "responseCode": "OK",
  "responseMessage": "potvrzeno"
```

PARAMETER	TYPE	DESCRIPTION
daughterCompanyId	string	ID of the daughter company

Tab. 96 *Parameters of the Last transaction V2 response*

PARAMETER		TYPE	DESCRIPTION
aid		string	Application identifier
amount		int	Amount of the transaction in cents. 500 is 5.00 euro
aosa		string	Available offline sending amount
appLabel		string	Label of application
authCode		int	Code of the authorization
availableBalance		int	Balance of the available funds
callReversal		bool	Information whether the reversal should be called or not.
cardInputType		string	Type of the input. Enum is in the appendix .
conto		string	Card issuer
currency		int	Code of the currency
cvmTypeList		list	Type of the TRX. Enum is in the appendix .
dateTimeServer		string	Date and time on server when, was the result made
dateTimeTerminal		string	Date and time on terminal, when was the result made
emvResponseData		obj	Object of the response.
	smartCardScheme	int	Scheme of the smart card
	authorisationResponseCode	string	The response code of authorization.
expiration		string	Expiration date of the card.
invoiceNumber		string	Variable symbol, max length is 20 char
isReversed		bool	Information whether the transaction was reversed.
merchantIdAcquirer		string	ID of the merchant's acquirer.
merchantIdIssuer		int	ID of the merchant's issuer

pan	int	Primary account number
panSequenceNumber	int	Sequence number of the primary account number
responseCode	string	Code of the response
responseMessage	string	Message of the response
reversal	bool	Information
sequenceNumber	int	Number of the sequence
terminalIdAcquirer	string	ID of the acquirer's terminal
terminalIdIssuer	string	ID of the issuer's terminal
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
transactionType	string	Type of the transaction. Enum is in the appendix .

Tab. 97 Parameters of the Last transaction V2 response

HTTP StatusCode: 404 – No transaction found

Response – Closing

```
{
  "message": "Not found"
}
```


6.1.19. Get successful transactions V2

Obtain all successful transactions.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/transactions

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password"
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal

Tab. 98 Parameters of the Get successful transactions V2 request

Response example:

JSON

```
[
  {
    "aid": "A0000000283101",
    "amount": 500,
    "aosa": "null",
    "appLabel": "Air Bank",
    "authCode": "114546",
    "availableBalance": "1000",
    "callReversal": false,
    "cardInputType": "CLESS",
    "conto": "MASTERCARD",
    "currency": 203,
    "cvmTypeList": ["PIN_ONLINE"],
    "dateTimeServer": "Jul 16, 2020 14:40:50",
    "dateTimeTerminal": "Jul 16, 2020 14:39:37",
    "emvResponseData": {
      "smartCardScheme": "1",
      "authorisationResponseCode": "00"
    },
    "expiration": "****",
    "invoiceNumber": "VS",
    "isReversed": false,
    "merchantIdAcquirer": "TNEXGO01",
    "merchantIdIssuer": "1357",
    "pan": "970348*****3910",
    "panSequenceNumber": "00",
    "responseCode": "OK",
    "responseMessage": "potvrzeno",
    "reversal": false,
  }
]
```

PARAMETER	TYPE	DESCRIPTION
daughterCompanyId	string	ID of the daughter company

Tab. 99 Parameters of the Get successful transactions V2 response

```

    "tokens":
    ["8798F7W98723KDGf08723HKDJHF897"],
    "sequenceNumber": "001001614",
    "terminalIdAcquirer":
    "M1TNEXGO01",
    "terminalIdIssuer":
    "M1TNEXGO01",
    "transactionId":
    "48a3ea7d-b82f-4b42-abbb-5043365e4fa
    a",
    "transactionType": "SALE_ONLINE"
  }
]

```

PARAMETER	TYPE	DESCRIPTION
aid	string	Application identifier
amount	int	Amount of the transaction in cents. 500 is 5.00 euro
aosa	string	Available offline sending amount
appLabel	string	Label of application
authCode	int	Code of the authorization
availableBalance	int	Balance of the available funds
callReversal	bool	Information whether the reversal should be called or not.
cardInputType	string	Type of the input. Enum is in the appendix .
conto	string	Card issuer
currency	int	Code of the currency
cvmTypeList	list	Type of the TRX. Enum is in the appendix .
dateTimeServer	string	Date and time on server when, was the result made
dateTimeTerminal	string	Date and time on terminal, when was the result made
emvResponseData	obj	Object of the response.
smartCardScheme	int	Scheme of the smart card
	authorisationResponseCode	The response code of authorization.
expiration	string	Expiration date of the card.
invoiceNumber	string	Variable symbol, max length is 20 char
isReversed	bool	Information whether the transaction was reversed.
merchantIdAcquirer	string	ID of the merchant's acquirer.

merchantIdIssuer	int	ID of the merchant's issuer
pan	int	Primary account number
panSequenceNumber	int	Sequence number of the primary account number
responseCode	string	Code of the response
responseMessage	string	Message of the response
reversal	bool	Information
sequenceNumber	int	Number of the sequence
terminalIdAcquirer	string	ID of the acquirer's terminal
terminalIdIssuer	string	ID of the issuer's terminal
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
transactionType	string	Type of the transaction. Enum is in the appendix .

Tab. 100 Parameters of the Get successful transactions V2 response

6.1.20. Transaction result V2

The Result method calls for the result of the payment. If the payment is still in progress, it returns progress information.

NOTE: Calling /result and retrieving the result is not considered complete completion and confirmation of the transaction if the transaction requires /api/switchio/pay/v2/confirm. If the endpoint /api/switchio/pay/v2/confirm is not called for a transaction that requires confirmation, an automatic reversal is performed.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/result

Content-type: application/json

Request example:

JSON

```
{
  "transactionId": "",
  "secureString": "password"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
secureString	string	Password of the terminal

Tab. 101 Parameters of the Transaction result V2 request

Response example, when the payment is in progress:

Transaction with HTTP status code 200

JSON

```
{
  "aid": "A0000000283101",
  "amount": 500,
  "aosa": "null",
  "appLabel": "Air Bank",
  "authCode": "114546",
  "availableBalance": "1000",
  "callReversal": false,
  "cardInputType": "CLESS",
  "conto": "MASTERCARD",
  "currency": 203,
  "cvmTypeList": ["PIN ONLINE"],
  "dateTimeServer": "Jul 16, 2020 14:40:50",
  "dateTimeTerminal": "Jul 16, 2020 14:39:37",
  "emvResponseData": {
    "smartCardScheme": "1",
    "authorisationResponseCode": "00"
  },
  "expiration": "****",
  "invoiceNumber": "VS",
  "isReversed": false,
}
```

```

    "merchantIdAcquirer": "TNEXGO01",
    "merchantIdIssuer": "1357",
    "pan": "970348*****3910",
    "panSequenceNumber": "00",
    "responseCode": "OK",
    "responseMessage": "potvrzeno",
    "reversal": false,
    "sequenceNumber": "001001614",
    "terminalIdAcquirer": "M1TNEXGO01",
    "terminalIdIssuer": "M1TNEXGO01",
    "transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
    "transactionType": "SALE_ONLINE",
    "tokens": ["fkdfkjbejrhbjb" ] //default - empty list
  }

```

PARAMETER		TYPE	DESCRIPTION
aid		string	Application identifier
amount		int	Amount of the transaction in cents. 500 is 5.00 euro
aosa		string	Available offline sending amount
appLabel		string	Label of application
authCode		int	Code of the authorization
availableBalance		int	Balance of the available funds
callReversal		bool	Information whether the reversal should be called or not.
cardInputType		string	Type of the input. Enum is in the appendix .
conto		string	Card issuer
currency		int	Code of the currency
cvmTypeList		list	Type of the TRX. Enum is in the appendix .
dateTimeServer		string	Date and time on server when, was the result made
dateTimeTerminal		string	Date and time on terminal, when was the result made
emvResponseData		obj	Object of the response.
	smartCardScheme	int	Scheme of the smart card.
	authorisationResponseCode	string	The response code of authorization.
expiration		string	Expiration date of the card.
invoiceNumber		string	Variable symbol, max length is 20 char
isReversed		bool	Information whether the transaction was reversed.
merchantIdAcquirer		string	ID of the merchant's acquirer.

merchantIdIssuer	int	ID of the merchant's issuer
pan	int	Primary account number
panSequenceNumber	int	Sequence number of the primary account number
responseCode	string	Code of the response
responseMessage	string	Message of the response
reversal	bool	Information
sequenceNumber	int	Number of the sequence
terminalIdAcquirer	string	ID of the acquirer's terminal
terminalIdIssuer	string	ID of the issuer's terminal
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
transactionType	string	Type of the transaction. Enum is in the appendix .

Tab. 102 Parameters of the Result method response

Response example, after the start closing method is processed:

JSON

```
{
  "batchTotals": {
    "adjsAmount":0.00,
    "adjsCount":0,
    "batchNumber":11,
    "creditsAmount":0.00,
    "creditsCount":0,
    "debitsAmount":0.00,
    "debitsCount":0,
    "shiftNumber":1
  },
  "dateTimeServer":"Jul 9, 2020 10:31:38 AM",
  "dateTimeTerminal":"Jul 9, 2020 10:31:50 AM",
  "responseCode":"OK",
  "responseMessage":"Soucty sum souhlasi",
  "transactionType":"CLOSE_BATCH"
}
```

PARAMETER	TYPE	DESCRIPTION
batchTotals	obj	Object of the batch totals
adjsAmount	int	Amount of adjustments
adjsCount	int	Count of adjustments
batchNumber	int	Number of batch
creditsAmount	int	Amount of refund
creditsCount	int	Count of refund TRX
debitsAmount	int	Amount of payment TRX
debitsCount	int	Count of payment TRX
shiftNumber	int	number of shift
dateTimeServer	string	Date and time on server when, was the result made
dateTimeTerminal	string	Date and time on terminal, when was the result made
responseCode	string	Code of the response
responseMessage	string	Message of the response
transactionType	string	Type of the TRX. Enum is in the appendix .

Tab. 103 Parameters of the result method response when the Start closing is processed

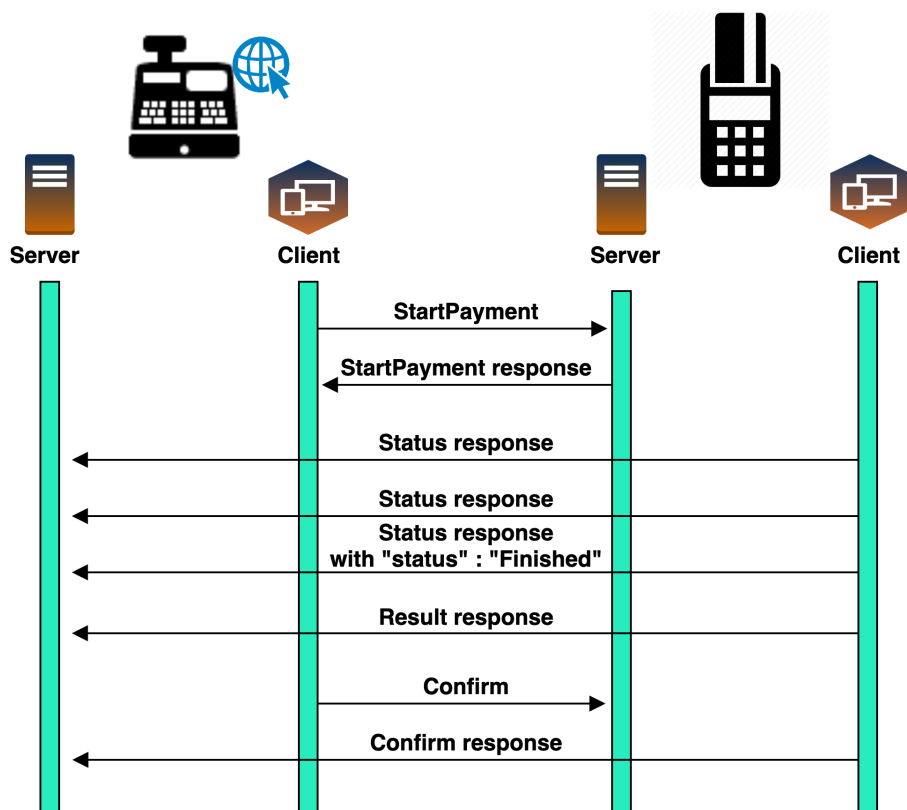
Response - HTTP status code != 200

Response - Closing

```
{
  "message": "Unauthorized"
}
```

6.2. Cash register and terminal work as both client and server

Example of payment workflow:



6.2.1. Payment callback V2

The Start Payment method triggers the payment. In the cash register and terminal works as both client and server mode we use callback within this method. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cbpayment

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
amount	int	Amount of the payment
currencyCode	int	Currency of the payment amount
invoiceNumber	string	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 104 Parameters of the Payment callback V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in
progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 105 Parameters of the Payment callback V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed

Response - Closing

```
{
  "message": "Already processed"
}
```

6.2.2. MOTO payment callback V2

The Start MOTO Payment method triggers the MOTO payment. In the cash register and terminal works as both client and server mode we use callback within this method. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cbpayment/moto

Content-type: application/json

Request:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 203,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
amount	int	Amount of the payment
currencyCode	int	Currency of the payment amount
invoiceNumber	int	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of progress to callback
ip	string	IP of cash register to callback
port	int	Port of cash register to callback
endpoint	string	Endpoint of method to callback
resultCallback	obj	Object of result to callback
ip	string	IP of cash register to callback
port	int	Port of cash register to callback
endpoint	string	Endpoint of the method to callback

Tab. 106 Parameters of the MOTO payment callback V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 107 Parameters of the MOTO payment callback V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed

Response - Closing

```
{
  "message": "Already processed"
}
```

6.2.3. Refund callback V2

The Start Refund method triggers the refund. In the cash register and terminal works as both client and server mode we use callback within this method. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cbrefund

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
amount	int	Amount of the refund
currencyCode	int	Currency of the refund
invoiceNumber	string	Variable symbol, length is 20 char
daughterCom panyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 108 Parameters of the Start refund method with callback request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in
progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 109 Parameters of the Refund callback V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed

Response - Closing

```
{
  "message": "Already processed"
}
```

6.2.4. Reversal callback V2

The Start Reverse method triggers the reversal. In the cash register and terminal works as both client and server mode we use callback within this method.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cbreverse

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "originalTransactionId": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER		TYPE	DESCRIPTION
secureString		string	Password of the terminal
request		obj	Object of the request
	transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
	originalTransactionId	string	ID of original transaction what is being reversed
	daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback		obj	Object of the progress to callback
	ip	string	IP to callback
	port	int	Port to callback
	endpoint	string	Endpoint of the method to callback
resultCallback		obj	Object of result to callback
	ip	string	IP to callback
	port	int	Port to callback
	endpoint	string	Endpoint of the method to callback

Tab. 110 *Parameters of the Reversal callback V2 request*

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in
progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. III Parameters of the Reversal callback V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed
- **StatusCode: 404** → The transaction to be reversed was not found in the database

Response - Closing

```
{
  "message": "Already processed"
}
```


6.2.5. Preauthorization callback V2

The Start Preauth method creates a new pre-authorization transaction. Option to cancel the transaction with /cancel when obtaining the status with the value "WaitForCard". It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cbpreauth

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "amount": 10053,
    "currencyCode": 978,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER		TYPE	DESCRIPTION
secureString		string	Password of the terminal
request		obj	Object of the request
	transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
	amount	int	Amount of the preauth
	currencyCode	int	Currency of the preauth
	invoiceNumber	string	Variable symbol, length is 20 char
	daughterComp anyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback		obj	Object of the progress to callback
	ip	string	IP to callback
	port	int	Port to callback
	endpoint	string	Endpoint of the method to callback
resultCallback		obj	Object of result to callback
	ip	string	IP to callback
	port	int	Port to callback
	endpoint	string	Endpoint of the method to callback

Tab. 112 Parameters of the Preauthorization callback V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in
progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 113 Parameters of the Preauthorization callback V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed

Response - Closing

```
{
  "message": "Already processed"
}
```

6.2.6. PreAuth Complete - callback V2

The Start Preauth completion method completes a preauth transaction that is identified by originalTransactionId. It is not possible to cancel with the /cancel endpoint. It is also necessary to call /confirm after obtaining the Finished status. If it is not called /confirm after the payment, the reversed transaction will be performed automatically.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cbpreauth/complete

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "originalTransactionId": "",
    "finalAmount": 10053,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227alc3290b6
originalTransactionId	string	ID of the original transaction. Example: 1e6a1251-762f-45fc-85fb-227alc3290b6
finalAmount	int	Final amount of the preauth, in cents 10053 is 100.53 euro
invoiceNumber	int	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 114 Parameters of the PreAuth Complete - callback V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 115 Parameters of the PreAuth Complete - callback V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed
- **StatusCode: 404** → Transaction with originalTransactionId not found

Response - Closing

```
{
  "message": "Already processed"
}
```

6.2.7. PreAuth Increment – callback V2

The Start Preauth increment method increments the amount for a preauth transaction that is identified by the originalTransactionId in the request. It is not possible to cancel with the /cancel endpoint.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cbpreauth/increment

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "originalTransactionId": "",
    "incrementAmount": 10053,
    "invoiceNumber": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
originalTransactionId	string	ID of the original transaction. Example: <i>1e6a1251-762f-45fc-85fb-227a1c3290b6</i>
incrementAmount	int	Incremental amount of the preauth, in cents 10053 is 100.53 euro
invoiceNumber	string	Variable symbol, length is 20 char
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 116 Parameters of the PreAuth Increment - callback V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 117 Parameters of the PreAuth Increment - callback V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed
- **StatusCode: 404** → Transaction with originalTransactionId not found

Response - Closing

```
{
  "message": "Already processed"
}
```

6.2.8. PreAuth Cancel – callback V2

The Start Preauth cancel method cancels the preauth transaction that is identified by the originalTransactionId. It is not possible to cancel with the /cancel endpoint. It is NOT necessary to call /confirm after obtaining the Finished status.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cbpreauth/cancel

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "originalTransactionId": "",
    "daughterCompanyId": "",
    "progressCallback": {
      "ip": "172.25.11.14",
      "port": 7777,
      "endpoint": "/gsa/progress"
    },
    "resultCallback": {
      "ip": "172.25.11.14",
      "port": 7777,
      "endpoint": "/gsa/result"
    }
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
originalTransactionId	string	ID of the original transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 118 Parameters of the PreAuth Cancel – callback V2 request

Response

HTTP Status code

- **StatusCode: 200** → the transaction starts, the client expects the status "Finished" in /progress request
- **StatusCode: 202** → the request was processed but the transaction was not started - server busy

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the payment has started.
status	string	Message of the payment status.

Tab. 119 Parameters of the PreAuth Cancel - callback V2 response

- **StatusCode: 400** → A transaction with the same ID has already been processed
- **StatusCode: 404** → Transaction with originalTransactionId not found

Response - Closing

```
{
  "message": "Already processed"
}
```


6.2.9. Closing callback V2

The Start Closing method triggers the closing. In the cash register and terminal works as both client and server mode we use callback within this method.

Method: POST

Endpoint: ip:port/api/switchio/pay/v2/cbclosing

Content-type: application/json

Request example:

JSON

```
{
  "secureString": "password",
  "request": {
    "transactionId": "",
    "daughterCompanyId": ""
  },
  "progressCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/progress"
  },
  "resultCallback": {
    "ip": "172.25.11.14",
    "port": 7777,
    "endpoint": "/gsa/result"
  }
}
```

PARAMETER	TYPE	DESCRIPTION
secureString	string	Password of the terminal
request	obj	Object of the request
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
daughterCompanyId	string	ID of the daughter company. Required only when virtualization is enabled.
progressCallback	obj	Object of the progress to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback
resultCallback	obj	Object of result to callback
ip	string	IP to callback
port	int	Port to callback
endpoint	string	Endpoint of the method to callback

Tab. 120 Parameters of the Closing callback V2 req.

Response example:

JSON

```
{
  "transactionId": "",
  "isStarted": false,
  "status": "Other payment in progress"
}
```

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction. Example: 1e6a1251-762f-45fc-85fb-227a1c3290b6
isStarted	bool	Information whether the refund has started.
status	string	Message of the payment status.

Tab. 121 Parameters of the Closing callback V2 response

7. CASHDESK

Following section describes the CashDesk (<http://demo.cashdesk.switchio.com/#/>), the Monet+ demo-implementation of the cash register's application. This implementation communicates with a Switchio Pay application through the GSA protocol. The Switchio Pay is a payment application present in the terminal. Terminal works as the server, the cash register works as a client.

The communication between the terminal and the cash register is not via the public internet. Cash register doesn't have to have connectivity to the internet. But must be in the local network with the terminal.

7.1. Login

The web address will be provided by the Monet+.

To login into the application you will need the following:

- IP address
- port number
- password

The IP address is subject to the local network. The port number can be set according to the client's request. And the password is generated and provided by the Monet+.

Enter the IP address and the port number to the colons and tap the *Make connection* button. (see Picture 1 – Login I) When the connection is made, a new tab occurs. Enter the password given from the Monet+ and tap the *Login* button.

Payment terminal management

IP address Port number

Enter the data for the connection to the payment terminal

Make a connection

Pic.1 Login I

Payment terminal management

Terminal id

TNEXGO12 X

Password

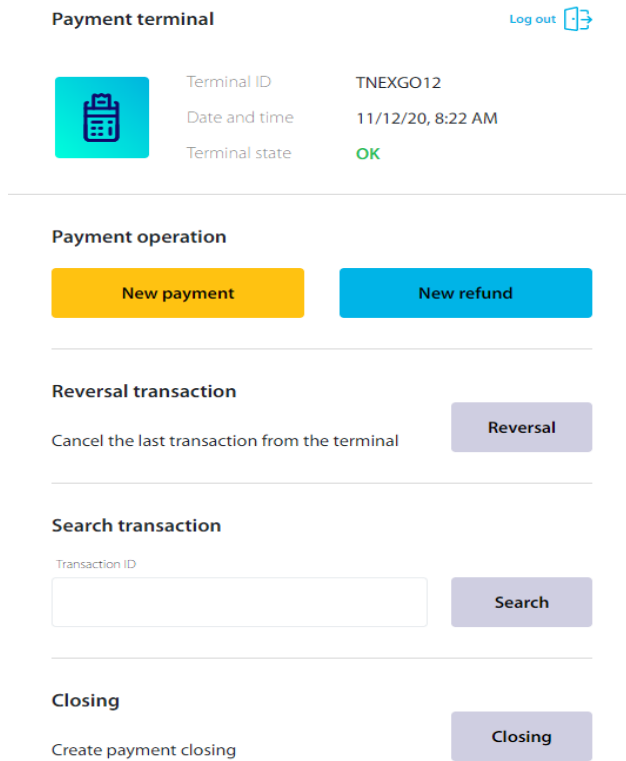
X

Enter the password for the payment terminal.

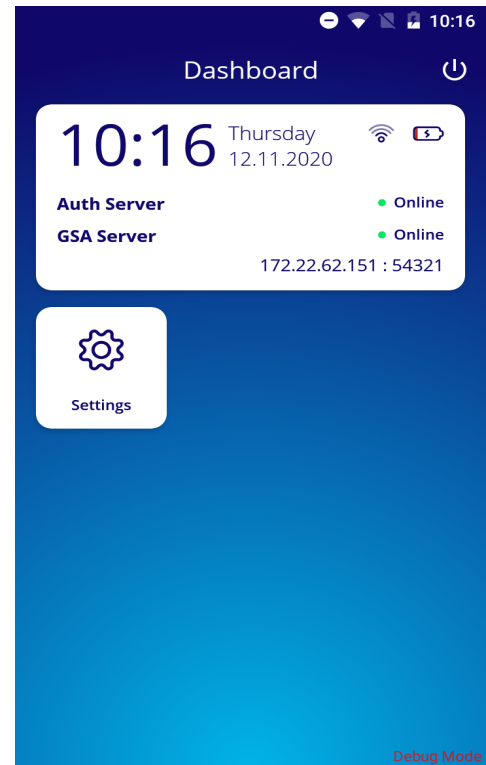
Login

Pic.2 Login II

When you're successfully logged, you will get to the homepage(see pic. 3 - The homepage on the web).



Pic. 3 The homepage on the web



Pic. 4 The terminal's dashboard

Start the Switchio Pay application on the terminal, the *Dashboard* page(see pic. 4 - The terminal's dashboard) will display.

Methods called during the Login process are:

- Info method
- Terminal Status

New payment

Amount

Currency

Variable symbol

Transaction ID
528b7d54-f32e-4dd0-a2b2-ee7e29fb7a5d


Pic. 6 The new payment setting

- Confirmation that the payment is being processed shows up.(see pic.7 - Confirmation that payment being processed)

Tap, Insert or Swipe the card to the terminal.(see pic.8 - Terminal payment process). At this point you are able to cancel the payment on both the terminal or the web cash register.

New payment


Amount: €10.00
Transaction ID: 90c8de04-6572-4061-9535-f9ff2cd823b3



Follow the instructions on the payment terminal to confirm

Pic. 7 Confirmation that the payment being processed

Payment
€ 10.00



Tap, Insert or Swipe Card

Debug Mode

Pic. 8 Terminal payment process

- At the web cash register accept the payment (see pic.9 - Accept payment)

New payment

Amount: **€10.00**
Transaction type: **SALE_ONLINE**
Transaction ID: **8f60dd3f-d42b-4d3c-af32-34b6bb331d1c**


Transaction ID	8f60dd3f-d42b-4d3c-af32-34b6bb331d1c
Transaction type	SALE_ONLINE
AID	A0000000041010
Amount	€10.00
Auth code	754909
Currency	EUR
Date and time of server	"Nov 12, 2020 09:05:40"
Date and time of terminal	"Nov 12, 2020 09:07:04"
MerchantIdAcquirer	TNEXGO12
TerminalIdAcquirer	M1TNEXGO12
MerchantIdIssuer	1357
TerminalIdIssuer	M1TNEXGO12
PAN	557392*****4095
Response code	OK
Response	potvrzeno
CONTO	MASTERCARD
Sequence number	001001286

Reject **Accept**

Pic. 9 *Accept the payment*

New payment

Amount: **€10.00**
Transaction ID: **7387ff90-b23c-405d-9bc3-51005acef685**



Payment was successfully accepted.

New payment **Back**

Pic. 10 *The payment confirmation*

5. The payment is confirmed (see pic. 10 – The payment confirmation).

Methods called during the payment process are:

- Start payment
- Status
- Result
- Confirm

7.2.2. Refund

The refund setup:

1. Tap the *refund* button (see pic. 11 – Tap the refund button)

Payment terminal
Log out

Terminal ID
TNEXGO12
Date and time
11/12/20, 8:22 AM
Terminal state
OK

Payment operation

New payment

New refund

Reversal transaction

Cancel the last transaction from the terminal

Reversal

Search transaction

Transaction ID

Search

Closing

Create payment closing

Closing

Pic. 11 Tap the refund button

2. Enter the *amount* of the refund. Pick the *currency* from the scroll down button option. Entering the *Variable symbol* is optional. Tap the *Send refund* button (see pic. 6 – The refund setting)

New refund

Amount Currency

Variable symbol

Transaction ID
9703337c-5d11-43da-a1d1-9cd9adfc661b


Pic. 12 The Refund setting

- Confirmation that the refund is being processed shows up.(see pic.13 – Confirmation that refund is being processed)

Tap, Insert or Swipe the card to the terminal.(see pic.14 – Terminal refund process). At this point you are able to cancel the payment on both the terminal or the web cash register.

New refund


Amount: €10.00
Transaction ID: 8d63c550-31ad-4052-b14f-9352b46e8b7a



Follow the instructions on the payment terminal to confirm

Pic. 13 Confirmation that refund is being processed

Refund
€ 10.00



Tap, Insert or Swipe Card

Debug Mode

Pic. 14 Terminal refund process

- At the web cash register accept the payment(see pic.15 – Accept the refund)

New refund

Amount: **€10.00**
Transaction type: **RETURN_ONLINE**
Transaction ID: **f1e41238-52d8-4da4-872a-de6a1050cfc2**

Transaction ID	f1e41238-52d8-4da4-872a-de6a1050cfc2
Transaction type	RETURN_ONLINE
AID	A0000000041010
Amount	€10.00
Auth code	340743
Currency	EUR
Date and time of server	"Nov 12, 2020 08:28:35"
Date and time of terminal	"Nov 12, 2020 08:29:58"
MerchantIdAcquirer	TNEXGO12
TerminalIdAcquirer	M1TNEXGO12
MerchantIdIssuer	1357
TerminalIdIssuer	M1TNEXGO12
PAN	557392*****4095
Response code	OK
Response	potvrzeno
CONTO	MASTERCARD
Sequence number	001001233

Reject Accept

Pic. 15 *Accept the refund*

New refund

Amount: **€10.00**
Transaction ID: **47c322fb-8956-4d8c-8685-42e6d5aa5d4d**



Payment was successfully accepted.

New refund Back

Pic. 16 *The refund confirmation*

5. The payment is confirmed (see pic. 16 – The refund confirmation).

Methods called during the refund process are:

- Start refund
- Status
- Result
- Confirm

7.2.3. Reversal

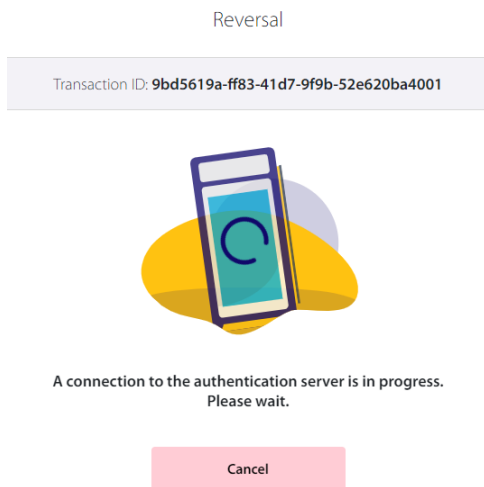
The reversal setup:

1. Tap the *Reversal* button (see pic.17 – Tap the reversal button).

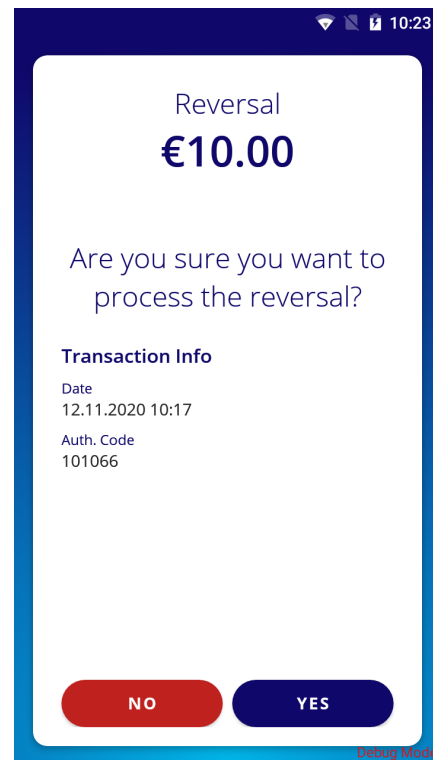
The screenshot shows the MONET+ Payment terminal interface. At the top, there's a 'Payment terminal' section with a 'Log out' button and a terminal icon. Below this, terminal details are listed: Terminal ID (TNEXGO12), Date and time (11/12/20, 8:22 AM), and Terminal state (OK). The 'Payment operation' section contains 'New payment' and 'New refund' buttons. The 'Reversal transaction' section has the instruction 'Cancel the last transaction from the terminal' and a 'Reversal' button, which is highlighted with a red circle and a red arrow. Below this is the 'Search transaction' section with a 'Transaction ID' input field and a 'Search' button. The 'Closing' section at the bottom has a 'Create payment closing' label and a 'Closing' button.

Pic. 17 Tap the reversal button

2. Confirmation that the reverse of the last payment is being processed shows up.(see pic.18 – Confirmation that the reverse of the last payment is being processed)
Press the *YES* button on the terminal.(see pic.19 – Terminal reversal process to confirm). At this point you are able to cancel the payment on both the terminal(with the *NO* button) or the web cash register.

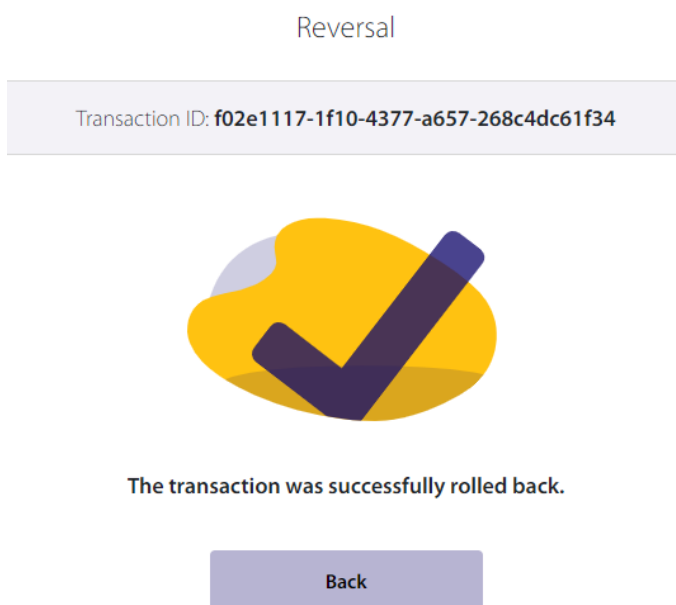


Pic. 18 Confirmation that the reverse of the last payment is being processed



Pic. 19 Terminal reversal process to confirm

3. The payment is confirmed (see pic. 20 – The reversal confirmation).




Pic. 20 The reversal confirmation


Methods called during the reversal process are:

- Start reverse
- Status
- Confirm

The closing setup:

- Payment terminal

Log out 



Terminal ID

TNEXGO12

Date and time

11/12/20, 8:22 AM

Terminal state

OK

Payment operation

New payment

New refund

Reversal transaction

Cancel the last transaction from the terminal

Reversal

Search transaction

Transaction ID

Search

Closing

Create payment closing

Closing

Pic. 21 Tap the closing button

- Closing

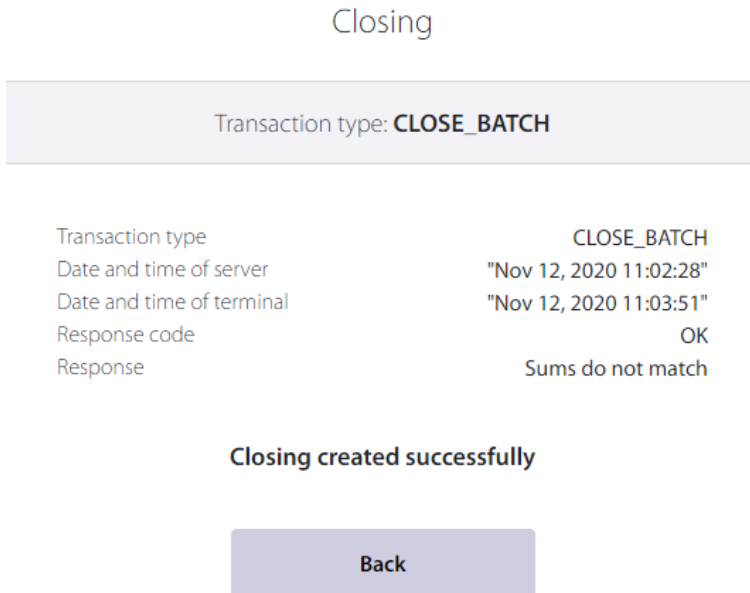
Are you sure you want to create a payment closing?

Back Create

The image shows a confirmation dialog box. At the top, the title 'Closing' is centered. Below it is a large, abstract graphic consisting of a yellow blob with a dark blue pill-shaped element on top. The question 'Are you sure you want to create a payment closing?' is centered below the graphic. At the bottom, there are two buttons: 'Back' on the left and 'Create' on the right.

Pic. 22 Tap the Create button

3. The confirmation of closing is displayed (see pic.23 – The confirmation of the closing).



Pic. 23 *The confirmation of the closing*

Methods called during the closing process are:

- Start closing
- Status
- Result

8. APPENDIX 1: ENUMS

8.1. Status parameter

STATUS
WaitForCard
WaitForPin
Connecting
Connected
Disconnected
Sending
Receiving
Finished

8.2. CardInputType parameter

CardInputType
MAGNETIC
MAGNETIC_FALLBACK
CONTACT
CLESS
CLESS_MAGSTRIPE

8.3. CvmTypeList parameter

CvmTypeList
NO
SIGNATURE
PIN_ONLINE
PIN_OFFLINE
CUSTOMER_DEVICE

8.4. TransactionType parameter

CvmType
SALE_ONLINE
SALE_OFFLINE
PREAUTHORIZATION
PREAUTHORIZATION_COMPLETION
PREAUTHORIZATION_INCREMENT
RETURN_ONLINE
RETURN_OFFLINE
MOTO_SALE
CLOSE_BATCH
SUBTOTAL_BATCH
HANDSHAKE
UNDEFINED

8.5. ResultCode parameter

ResultCode			
ID	Response Code	Description (Response Message)	Result
0	OK	All processing is successful	✓
1	KERNEL_FAILED	Initialization of kernel failed	✗
2	KERNEL_NOT_ACCESS	Kernel isn't accessible	✗
3	TEXT_KERNEL_NOT_INIT	Kernel not initialized	✗
4	BLUETOOTH_NOT_SUPPORTED	Bluetooth not supported	✗
5	PINBLOCK_TIMEOUT	Timeout for read pinblock	✗
6	PINBLOCK_USER_CANCELLED	User cancel a pinblock	✗
7	CANCELLED	Default user cancel	✗
8	FAILED	Default failure	✗
501	SCA_PIN_REQUIRED	Passcode and biometric capable. It is not possible to insert a contact card, it is just contactless.	✗
1000	SPDH_DEFINE		-
1000	ApprovedBalancesAvailable	Approved balances available.	✓
1001	ApprovedNoBalancesAvailable	Approved no balances available.	✓
1002	ApprovedCountryClubStatus	Approved country club status.	✓
1003	Approved	Approved (maybe more identification is required).	✓
1004	ApprovedPendingIdentification	Approved pending identification (sign paper draft is required).	✓
1005	ApprovedBlind	Approved blind.	✓
1006	ApprovedVIPStatus	Approved VIP status.	✓
1007	ApprovedAdministrativeTransaction	Approved administrative transaction.	✓
1008	ApprovedNegativeCardFileHitOK	Approved negative card file hit OK.	✓

1009	ApprovedCommercialStatus	Approved commercial status.	✓
1010	ApprovedForPartialAmount	Approved for partial amount.	✓
1050	DeclinedGeneral	General decline.	X
1050	DeclinedProductGroup	Product group not allowed.	X
1050	DeclinedCardBlocked	The card is blocked.	X
1050	DeclinedCardBlockedByRange	The card is blocked by range.	X
1050	DeclinedAccountNotAllowedOnStation	The use of an account card is not allowed on the given gas station.	X
1050	DeclinedProductGroupForStation	Product group not allowed for the given gas station.	X
1050	DeclinedCardDailyTxCountLimitExceeded	The card has exceeded the daily maximum number of transactions.	X
1050	DeclinedCardWeeklyTxCountLimitExceeded	The card has exceeded the weekly maximum number of transactions.	X
1050	DeclinedCardMonthlyTxCountLimitExceeded	The card has exceeded the monthly maximum number of transactions.	X
1050	DeclinedCardDailyFuelLimitExceeded	The card has exceeded the daily fuel limit.	X
1050	DeclinedCardWeeklyFuelLimitExceeded	The card has exceeded the weekly fuel limit.	X
1050	DeclinedCardMonthlyFuelLimitExceeded	The card has exceeded the monthly fuel limit.	X
1050	DeclinedCardDailyMoneyLimitExceeded	The card has exceeded the daily money limit.	X
1050	DeclinedCardWeeklyMoneyLimitExceeded	The card has exceeded the weekly money limit.	X
1050	DeclinedCardMonthlyMoneyLimitExceeded	The card has exceeded its monthly cash limit.	X
1050	DeclinedCardTxPriceLimitExceeded	The card has exceeded the maximum cash limit per transaction.	X
1050	DeclinedCardDailyOtherLimitExceeded	The card has exceeded the daily money limit on dry goods.	X
1050	DeclinedCardWeeklyOtherLimitExceeded	The card has exceeded the weekly money limit on dry goods.	X
1050	DeclinedCardMonthlyOtherLimitExceeded	The card has exceeded the monthly money limit on dry goods.	X

1050	DeclinedCardTxFuelAmountLimitExceeded	The card has exceeded the maximum amount of fuel per transaction.	X
1050	DeclinedCardTxOtherPriceLimitExceeded	The card has exceeded the maximum price limit for dry goods per transaction.	X
1051	DeclinedExpiredCard	Expired card.	X
1052	DeclinedNumberOfPINtriesExceeded	Number of PIN tries exceeded.	X
1053	DeclinedNoSharingAllowed	No sharing allowed.	X
1054	DeclinedNoSecurityModule	No security module.	X
1055	DeclinedInvalidTransaction	Invalid transaction.	X
1056	DeclinedNotSupportedByInstitution	Transaction not supported by institution.	X
1057	DeclinedLostOrStolenCard	Lost or stolen card.	X
1058	DeclinedInvalidCardStatus	Invalid card status.	X
1059	DeclinedRestrictedStatus	Restricted status.	X
1060	DeclinedAccountNotFoundInCardholderDatabase	Account not found in cardholder database.	X
1061	DeclinedPositiveBalanceAccountRecordNotFound	Positive balance account record not found.	X
1061	DeclinedPositiveBalanceAccountRecordNotFound	Positive balance account update error.	X
1063	DeclinedInvalidAuthorizationTypeInInstitutionDatabase	Invalid authorization type in institution database.	X
1064	DeclinedBadTrackInformation	Bad track information.	X
1065	DeclinedAdjustmentNotAllowedInInstitutionDatabase	Adjustment not allowed in the institution database.	X
1066	DeclinedInvalidCreditCardAdvanceIncrement	Invalid credit card advance increment.	X
1067	DeclinedInvalidTransactionDate	Invalid transaction date.	X
1068	DeclinedTransactionLogFileError	Transaction log file error.	X
1069	DeclinedBadMessageEdit	Bad message edit.	X
1070	DeclinedNoInstitutionDatabaseRecord	No institution database record.	X
1071	DeclinedInvalidRoutingToHostApplication	Invalid routing to host application.	X
1072	DeclinedCardOnNationalNegativeFile	Card on national negative file.	X

	e		
1073	DeclinedInvalidRoutingAuthorizationService	Invalid routing authorization service.	X
1074	DeclinedUnableToAuthorize	Unable to authorize.	X
1074	DeclinedUnableToAuthorizeRepeat	Unable to authorize, retry transaction.	X
1075	DeclinedInvalidPANLength	Invalid PAN length.	X
1076	DeclinedInsufficientFundsInPositiveBalanceAccount	Insufficient funds in a positive balance account.	X
1077	DeclinedPreauthorizationFull	Preauthorization full.	X
1078	DeclinedDuplicateTransactionReceived	Duplicate transaction received.	X
1079	DeclinedMaximumOnlineRefundReached	Maximum online refund reached.	X
1080	DeclinedMaximumOfflineRefundReached	Maximum offline refund reached.	X
1081	DeclinedMaximumCreditPerRefundReached	Maximum credit per refund reached.	X
1082	DeclinedMaximumNumberOfTimesUsed	Maximum number of times used.	X
1083	DeclinedMaximumRefundCreditReached	Maximum refund credit reached.	X
1084	DeclinedCustomerSelectedNegativeCardFileReason	Customer selected negative card file reason.	X
1085	DeclinedInquiryNotAllowedNoBalances	Inquiry not allowed—no balances.	X
1086	DeclinedOverFloorLimit	Over floor limit.	X
1087	DeclinedMaximumNumberRefundCreditsReached	Maximum number refund credits reached.	X
1088	DeclinedPlaceCall	Place call.	X
1089	DeclinedCardStatusIsInactiveOrClosed	Card status equals 0 (inactive) or 9 (closed).	X
1090	DeclinedReferralFileFull	Referral file full.	X
1091	DeclinedProblemAccessingNegativeCardFile	Problem accessing negative card file.	X
1092	DeclinedAdvanceLessThanMinimum	Advance less than minimum.	X
1093	DeclinedDelinquent	Delinquent.	X

1094	DeclinedOverLimitTableOrExceedsAmountAvailable	Over limit table or exceeds the amount available.	X
1095	DeclinedAmountOverMaximum	Amount over maximum.	X
1096	DeclinedPINRequired	PIN required.	X
1097	DeclinedMod10Check	Mod 10 check.	X
1098	DeclinedForcePost	Force post.	X
1099	DeclinedCouldNotAccessPositiveBalanceAccountInDatabase	Could not access positive balance account in database	X
1100	ReferralUnableToProcessTransaction	Unable to process a transaction.	X
1101	ReferralUnableToAuthorizeIssueCall	Unable to authorize—issue call.	X
1102	ReferralCall	Call.	X
1103	ReferralProblemAccessingNegativeCardFile	Problem accessing negative card file.	X
1104	ReferralProblemAccessingCardholderAccount	Problem accessing cardholder account.	X
1105	ReferralCardNotSupported	Card not supported.	X
1105	ReferralCardNotFound	Card not found.	X
1105	ReferralCardReplenishmentSupported	Card replenishment not supported.	X
1106	ReferralAmountOverMaximum	Amount over maximum.	X
1107	ReferralOverDailyLimit	Over daily limit.	X
1108	ReferralCardAuthorizationParametersNotFound	Card authorization parameters not found.	X
1109	ReferralAdvanceLessThanMinimum	Advance less than minimum.	X
1110	ReferralNumberTimesUsed	Number times used.	X
1111	ReferralDelinquent	Delinquent.	X
1112	ReferralOverLimitTable	Over limit table	X
1113	ReferralTimeout	Timeout.	X
1115	ReferralTransactionLogFileFull	Transaction log file full.	X
1120	ReferralProblemAccessingCardholderUsageAccumulationData	Problem accessing cardholder usage accumulation data.	X
1121	ProblemAccessingAdministrativeCardData	Problem accessing administrative card data.	X
1122	ReferralUnableToValidatePIN	Unable to validate PIN; security	X

		module is down.	
1130	ReferralAuthorizationRequestCryptogram	Authorization request cryptogram (ARQC) referral.	X
1131	ReferralCardVerificationResults	Card verification results (CVR) referral.	X
1132	ReferralTerminalVerificationResults	Terminal verification results (TVR) referral.	X
1133	ReferralReasonOnlineCode	Reason online code referral.	X
1134	ReferralFallback	Fallback referral.	X
1150	ServiceMerchantNotOnFile	Merchant not on file.	X
1150	ServiceMerchantTerminalDisabled	Blocked terminal.	X
1150	ServiceMerchantAcquirerDisabled	Blocked terminal owner.	X
1200	TransactionInvalidAccount	Invalid account.	X
1201	TransactionIncorrectPIN	Incorrect PIN.	X
1202	TransactionCashAdvancesLessThanMinimum	Cash advance is less than minimum.	X
1203	TransactionAdministrativeCardNeeded	Administrative card needed.	X
1204	TransactionEnterLesserAmount	Enter a lesser amount (This code can also be used when the transaction amount exceeds the retailer ceiling limits.).	X
1205	TransactionInvalidAdvanceAmount	Invalid advance amount.	X
1206	TransactionCardholderAuthorizationDataNotFound	Cardholder authorization data not found.	X
1207	TransactionInvalidTransactionDate	Invalid transaction date.	X
1208	TransactionInvalidExpirationDate	Invalid expiration date.	X
1209	TransactionInvalidTransactionCode	Invalid transaction code.	X
1251	TransactionCashBackExceedsDailyLimit	Cash back exceeds daily limit.	X
1400	TransactionAuthorizationRequestCryptogramFailure	Authorization request cryptogram (ARQC) failure.	X
1401	TransactionHardwareSecurityModuleParameterError	Hardware security module parameter error.	X
1402	TransactionHardwareSecurityModuleFailure	Hardware security module failure.	X
1403	TransactionIntegratedCircuitCardKey	Integrated circuit card key	X

	yInformationNotFound	information not found.	
1404	TransactionApplicationTransactionCounterCheckFailure	Application transaction counter (ATC) check failure.	X
1405	TransactionCardVerificationResultsDecline	Card verification results (CVR) decline.	X
1406	TransactionTerminalVerificationResultsDecline	Terminal verification results (TVR) decline.	X
1407	TransactionReasonOnlineCodeDecline	Reason online code decline.	X
1408	TransactionFallbackDecline	Fallback decline.	X
1500	ProductDenied	Unauthorized product.	X
1557	SuspectedFraudCallWag	WAG antifraud. Suspected fraud, call WAG	X
1558	LockedCardSendSMS	WAG antifraud. Card locked, for unlock card send SMS to WAG	X
1800	TransactionFormatError	Format error.	X
1801	TransactionInvalidData	Invalid data.	X
1802	TransactionInvalidEmployeeNumber	Invalid employee number.	X
1809	TransactionInvalidCloseTransaction	Invalid close transaction.	X
1810	TransactionTimeout	Transaction timeout.	X
1811	TransactionSystemError	System error.	X
1811	SuperiorAuthorizationSystemError	Authorization failed on the master authorization system.	X
1820	TransactionInvalidTerminalIdentifier	Invalid terminal identifier.	X
1821	TransactionInvalidResponseLength	Invalid response length.	X
1870	MailDelivered	Mail delivered.	✓
1871	MailStored	Mail stored.	✓
1880	MailOrDownloadMessageHasBeenReceivedInItsEntirety	Mail or download message has been received in its entirety.	✓
1881	MailOrDownloadMessageReceivedSuccessfullyAndNoMoreData	Mail or download message received successfully and there is more data for this mail message.	✓
1882	DownloadAborted	Download aborted (call for service).	X
1878	DeclineIncorrectPINLengthError	Incorrect PIN length error.	X
1889	DeclineMACCommunicationsKeySy	MAC communications key (KMAC)	X

	nchronizationError	synchronization error.	
1898	DeclineInvalidMAC	Invalid MAC.	X
1899	DeclineSequenceErrorResync	Sequence error—resync.	X
1900	POSCaptureNumberOfPINAttemptsExceeded	Number of PIN tries exceeded.	X
1901	POSCaptureExpiredCard	Expired card.	X
1902	POSCaptureNegativeCardFileCaptureCode	Negative card file capture code.	X
1903	POSCaptureCardStatusIsStolen	Card status is 3 (stolen).	X
1904	POSCaptureAdvanceLessThanMinimum	Advance less than minimum.	X
1905	POSCaptureNumberTimesUsedExceeded	Number times used exceeded.	X
1906	POSCaptureDelinquent	Delinquent.	X
1907	POSCaptureOverLimitTable	Over limit table.	X
1908	POSCaptureAmountOverMaximum	Amount over maximum.	X
1909	POSCaptureCapture	Capture.	✓
1910	POSCaptureAuthorizationRequestCryptogramCapture	Authorization request cryptogram (ARQC) capture.	✓
1911	POSCaptureCardVerificationResultsCapture	Card verification results (CVR) capture.	✓
1912	POSCaptureTerminalVerificationResultsCapture	Terminal verification results (TVR) capture.	✓
1950	DeclinedAdministrativeCardNotFound	Administrative card not found.	X
1951	DeclinedAdministrativeCardNotAllowed	Administrative card not allowed.	X
1959	DeclinedAdministrativeTransactionsNotSupported	Administrative transactions not supported.	X
1952	ApprovedAdministrativeRequestInWindow	Approved administrative request—in window.	✓
1953	ApprovedAdministrativeRequestOutOfWindow	Approved administrative request—out of window.	✓
1954	ApprovedAdministrativeRequestAnytime	Approved administrative request—anytime.	✓
1955	ChargebackCustomerFileUpdated	Chargeback—customer file updated.	✓

1956	ChargebackCustomerFileUpdatedAcquirerNotFound	Chargeback—customer file updated—acquirer not found.	X
1957	ChargebackIncorrectPrefixNumber	Chargeback—incorrect prefix number.	X
1958	ChargebackIncorrectResponseCodeOrCardPrefixConfiguration	Chargeback—incorrect response code or card prefix configuration.	X
1960	ChargebackApprovedCustomerFileNotUpdated	Chargeback—approved customer file not updated.	X
1961	ChargebackApprovedCustomerFileNotUpdatedAcquirerNotFound	Chargeback—approved customer file not updated, acquirer not found.	X
1962	ChargebackAcceptedIncorrectDestination	Chargeback—accepted, incorrect destination.	X
1963	NoCardTimeout	No card timeout	X
1964	InvalidSignature	Invalid signature	X
2000	PAX_DEFINE	ReturnCodes by PAX	–
2001	ICC_RESET_ERR	IC cardreset is failed	X
2002	ICC_CMD_ERR	IC card command is failed	X
2003	ICC_BLOCK	IC card is blocked	X
2004	EMV_RSP_ERR	Status Code returned by IC card is not 9000	X
2005	EMV_APP_BLOCK	The application selected is blocked	X
2006	EMV_NO_APP	There is no AID matched between ICC and terminal	X
2007	EMV_USER_CANCEL	The current operation or transaction was canceled by user	X
2008	EMV_TIME_OUT	User's operation is timeout	X
2009	EMV_DATA_ERR	Data error is found	X
2010	EMV_NOT_ACCEPT	Transaction is not accepted	X
2011	EMV_DENIAL	Transaction is denial	X
2012	EMV_KEY_EXP	Certification Authority Public Key is Expired	X
2013	EMV_NO_PINPAD	PIN enter is required, but PIN pad is not present or not working	X
2014	EMV_NO_PASSWORD	PIN enter is required, PIN pad is present, but there is no PIN entered	X

2015	EMV_SUM_ERR	Checksum of CAPK is error	X
2016	EMV_NOT_FOUND	Appointed Data element can't be found	X
2017	EMV_NO_DATA	The length of the appointed Data Element is 0	X
2018	EMV_OVERFLOW	Memory is overflow	X
2019	NO_TRANS_LOG	There is no Transaction log	X
2020	RECORD_NOTEXIST	Appointed log is not existed	X
2021	LOGITEM_NOTEXIST	Appointed Label is not existed in current log record	X
2022	ICC_RSP_6985	Status Code returned by IC card for GPO is 6985	X
2023	CLSS_USE_CONTACT	Must use other interface for the transaction	X
2024	EMV_FILE_ERR	There is file error found	X
2025	CLSS_TERMINATE	Must terminate the transaction	X
2026	CLSS_FAILED	Contactless transaction is failed	X
2027	CLSS_DECLINE	Transaction should by declined	X
2028	CLSS_TRY_ANOTHER_CARD	Try another card (DPAS Only)	X
2030	CLSS_PARAM_ERR	Parameter is error = EMV_PARAM_ERR	X
2031	CLSS_WAVE2_OVERSEA	International transaction (for VISA AP payWaveLevel2 IC card use)	X
2033	CLSS_WAVE2_US_CARD	US card (for VISA AP payWaveL2 IC card use)	X
2034	CLSS_WAVE3_INS_CARD	Need to use IC card for the transaction (for VISA payWaveIC card use)	X
2035	CLSS_RESELECT_APP	Select the next AID in candidate list	X
2036	CLSS_CARD_EXPIRED	IC card is expired	X
2037	EMV_NO_APP_PPSE_ERR	Select PPSE command returns error code, or these is no matching Application	X
2038	CLSS_USE_VSDC	Switch to Contactless PBOC	X
2039	CLSS_CVMDECLINE	DVM result is decline	X
2040	CLSS_REFER_CONSUMER_DEVICE	Status Code returned by IC card is	X

		6986, please see phone	
2041	CLSS_LAST_CMD_ERR	Terminal did not receive the response of the last command of Read Record. (qPBOC only)	X
2042	CLSS_API_ORDER_ERR	APIs are called in wrong order. Please call Clss_GetDebugInfo_xxx to get error codes.	X
2043	CLSS_TORN_CARDNUM_ERR	Torn log's PAN is different with the reselect card's pan	X
2044	CLSS_TRON_AID_ERR	Torn log's AID is different with the reselect card's AID	X
2045	CLSS_TRON_AMT_ERR	Torn log's amount is different with the current transaction's amount	X
2046	CLSS_CARD_EXPIRED_REQ_ONLINE	Card expired and the transaction should go online	X
2047	CLSS_FILE_NOT_FOUND	ICC return 6A82 (File not found) in response to the SELECT command	X
2048	CLSS_TRY_AGAIN	Try again	X
2200	CLSS_PAYMENT_NOT_ACCEPT	Payment Type Not Accepted for flash	X
2301	CLSS_INSERTED_ICCCARD	IC card is detected during contactless transaction	X
2302	CLSS_SWIPED_MAGCARD	Magnetic stripe card is detected during contactless transaction	X
3000	MONEMV_CLS_KERNEL	Monet kernel	-
3256	MONEMV_CLS_KERNEL_STATUS_OK	The function is correctly executed.	✓
3257	MONEMV_CLS_KERNEL_STATUS_SERVICE_NOT_AVAILABLE	The kernel to be accessed is not loaded in the terminal.	X
3258	MONEMV_CLS_KERNEL_STATUS_DATABASE_ERROR	A database error occurred.	X
3259	MONEMV_CLS_KERNEL_STATUS_INVALID_INPUT_DATA	One of the provided data is incorrect.	X
3260	MONEMV_CLS_KERNEL_STATUS_NOT_SUPPORTED	The called function is not supported by the kernel.	X
3261	MONEMV_CLS_KERNEL_STATUS_LACK_OF_MEMORY	There is not enough memory to complete the transaction.	X
3262	MONEMV_CLS_KERNEL_STATUS_COMMUNICATION_ERROR	A communication error occurred with the contactless card.	X
3263	MONEMV_CLS_KERNEL_STATUS_MISSING_INPUT_DATA	A mandatory data is missing to perform the transaction.	X

3264	MONEMV_CLS_KERNEL_STATUS_ICC_MISSING_DATA	A mandatory ICC data is missing to perform the transaction.	X
3265	MONEMV_CLS_KERNEL_STATUS_ICC_INVALID_DATA	The data returned by the card is not correctly formatted.	X
3266	MONEMV_CLS_KERNEL_STATUS_ICC_REDUNDANT_DATA	Card returned redundant data.	X
3267	MONEMV_CLS_KERNEL_STATUS_ICC_DATA_FORMAT_ERROR	The card response is not correctly formatted (parsing error, empty response, etc).	X
3268	MONEMV_CLS_KERNEL_STATUS_TERM_MISSING_DATA	A mandatory terminal data is missing to perform the transaction.	X
3269	MONEMV_CLS_KERNEL_STATUS_CARD_BLOCKED	A Card is blocked and the transaction cannot be performed.	X
3270	MONEMV_CLS_KERNEL_STATUS_APPLICATION_BLOCKED	The application in the card is blocked.	X
3271	MONEMV_CLS_KERNEL_STATUS_REMOVE_AID	The AID shall be removed from the candidate list.	X
3272	MONEMV_CLS_KERNEL_STATUS_UNKNOWN_SW	Unexpected response code from the ICC.	X
3273	MONEMV_CLS_KERNEL_STATUS_CONDITION_OF_USE_NOT_SATISFIED	Conditions of use not satisfied" in the GPO response.	X
3274	MONEMV_CLS_KERNEL_STATUS_OFFLINE_APPROVED	Transaction succeeded and is offline approved.	✓
3275	MONEMV_CLS_KERNEL_STATUS_OFFLINE_DECLINED	Transaction is offline declined.	X
3276	MONEMV_CLS_KERNEL_STATUS_ONLINE_AUTHORISATION	Transaction succeeded and needs an online authorisation to be completed.	-
3277	MONEMV_CLS_KERNEL_STATUS_CANCELLED	Transaction has been cancelled.	X
3278	MONEMV_CLS_KERNEL_STATUS_USE_CONTACT_INTERFACE	The transaction has to be conducted over another interface (chip or swipe for example)	X
3279	MONEMV_CLS_KERNEL_STATUS_NOT_ALLOWED	This function cannot be called as no transaction is in process with this kernel or cancellation is not allowed according to the transaction progress.	X
3280	MONEMV_CLS_KERNEL_STATUS_CONTINUE	The transaction flow must continue.	-
3281	MONEMV_CLS_KERNEL_STATUS_SUSPEND	The transaction flow must be stopped.	X

3282	MONEMV_CLS_KERNEL_STATUS_INTERNAL_ERROR	A kernel internal error occurred.	X
3283	MONEMV_CLS_KERNEL_STATUS_LIB_INTERFACE_ERROR	An error occurred into the kernel interface (linked to the custom application).	X
3284	MONEMV_CLS_KERNEL_STATUS_EXPIRED_CERTIFICATE	The certificate used for Offline data authentication is expired.	X
3285	MONEMV_CLS_KERNEL_STATUS_REVOKED_CERTIFICATE	The certificate used for Offline data authentication is revoked.	X
3286	MONEMV_CLS_KERNEL_STATUS_CARD_UNKNOWN	The contactless card accessed isn't the same as at the beginning of the transaction. This error occurs for a new card presentation as Issuer Script Processing ...")	X
3287	MONEMV_CLS_KERNEL_STATUS_MOBILE	A mobile phone GPO response has been detected. This error allows to restart a selection application cycle with new user message (for mobile phone) ...")	X
3383	MONEMV_CLS_KERNEL_STATUS_UNKNOWN	Unknown	X
4000	FEITIAN_READER	Feitian reader	-
2147483647	UNDEFINED	Undefined	-

8.6. SLIP

dateTimeTerminal

DD.MM.RRRR hh:mm

Ticket number

00001

Merchant_name

Merchant_unit_name

Adress

Merchant ID:

12345678

merchantIdIssuer

Terminal ID:

ABCD1234

terminalIdIssuer

Internal MID:

EFGH4321

merchantIdAcquirer

Internal TID:

KLMN0001

terminalIdAcquirer

ÚČTENKA PRO OBCHODNÍKA

transactionType

PRODEJ

cardInputType

1,00 Eur

amount

pan

541234*****1234 / 01 (L) MASTERCARD

conto

aid

A0000000041010 Debit MasterCard

sequenceNumber

Seq. ID: 001001099

authCode

Auth.Code: 402010

cvmTypeList

No Pin

* No Refund *