

TECHNICAL DOCUMENT

# SWITCHIO PAY – ECR PROTOCOL

---

<b>SUBMITTED BY</b>	<b>MONET+,a.s.</b> Za Dvorem 505, Zlín - Štípa Czech Republic
<b>UPDATED</b>	<b>27. 5. 2025</b>
<b>VERSION</b>	<b>6.15.0</b>

---

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.

# CONTENT

<b>CONTENT</b>	<b>1</b>
<b>1. DOCUMENT VERSION HISTORY</b>	<b>5</b>
<b>2. APPLICATION VERSION HISTORY</b>	<b>8</b>
<b>3. INTRODUCTION</b>	<b>10</b>
3.1. INTENT	10
3.2. REQUEST	11
3.3. RESULT	12
3.3.1. Result structures and examples for each API version	12
Result structure v1	12
Result structure v2	14
Result structure v3	17
Result structure v4	19
Result structure v5	22
Result structure v6	24
Result structure v7	28
<b>4. CONTROL ENDPOINTS</b>	<b>32</b>
4.1. CLOSING BATCH	32
4.1.1. Request	32
4.1.2. Result	33
4.2. DAUGHTER COMPANIES	34
4.2.1. Request	34
4.2.2. Result	34
4.3. SCANNER	35
4.3.1. Request	35
4.3.2. Result	35
4.4. UPDATE TERMINAL CONFIGURATION	36
4.4.1. Request	36
4.4.2. Result	37
4.5. TOUCH - AUTOMATIC CONFIGURATION UPDATE	38
4.5.1. Broadcast	38
4.5.2. Request	39
4.5.3. Result	39
4.6. SUBTOTAL	40
4.6.1. Request	40
4.6.2. Result	40
4.7. HANDSHAKE	41
4.7.1. Request	41
4.7.2. Result	41
4.8. TOKEN	42
4.8.1. Request	42
4.8.2. Result	43

4.9. GET LAST TRANSACTION	43
4.9.1. Request	43
4.9.2. Result	44
4.10. GET ALL SUCCESSFUL TRANSACTIONS	45
4.10.1. Request	45
4.10.2. Result	46
4.11. CARD VERIFY	47
4.11.1. Request	47
4.11.2. Result	47
4.12. SETTINGS	50
4.12.1. Request	50
4.12.2. Result	50
4.13. OPEN PREAUTHORIZATIONS	51
4.13.1. Request	51
4.13.2. Result	51
4.14. INITIALIZE TERMINAL	53
4.14.1. Request	53
4.14.2. Result	53
<b>5. RETAIL ENDPOINTS</b>	<b>55</b>
5.1. PAYMENT	55
5.1.1. Request	55
5.1.2. Result	56
5.2. REFUND	57
5.2.1. Request	57
5.2.2. Result	57
5.3. PREAUTHORIZATION	58
5.3.1. Request	58
5.3.2. Result	59
5.4. INCREMENT PREAUTHORIZATION	59
5.4.1. Request	59
5.4.2. Result	60
5.5. DECREMENT PREAUTHORIZATION	60
5.5.1. Request	60
5.5.2. Result	60
5.6. COMPLETE PREAUTHORIZATION	61
5.6.1. Request	61
5.6.2. Result	62
5.7. CANCEL PREAUTHORIZATION	62
5.7.1. Request	62
5.7.2. Result	62
5.8. PREAUTHORIZATION - EXTERNAL COMPLETE	63
5.8.1. Request	63
5.8.2. Result	64
5.9. PREAUTHORIZATION - EXTERNAL CANCEL	64
5.9.1. Request	64
5.9.2. Result	65

5.10. GET TRANSACTION	65
5.10.1. Request	65
5.10.2. Result	65
5.11. REVERSAL	66
5.11.1. Request	66
5.11.2. Result	66
5.12. MOTO	67
5.12.1. Request	67
5.12.2. Result	68
5.13. TRANSACTION STATUS	68
5.13.1. Request	68
5.13.2. Result	68
5.14. REFERRAL REFUND	69
5.14.1. Request	69
5.14.2. Result	70
5.15. CASHBACK	70
5.15.1. Request	70
5.15.2. Result	70
5.16. QR PAYMENT	71
5.16.1. Request	71
5.16.2. Result	71
5.17. READ CARD	72
5.18. Request	72
5.19. Result	72
5.20. READ CARD PAYMENT	73
5.21. Request	73
5.22. Result	74
<b>6. TRANSPORT ENDPOINTS</b>	<b>75</b>
6.1. TRANSPORT INIT	75
6.1.1. Request	75
6.1.2. Result	75
6.2. TRANSPORT PAYMENT	76
6.2.1. Request	76
6.2.2. Result	77
6.3. TRANSPORT REFUND	79
6.3.1. Request	79
6.3.2. Result	80
6.4. TRANSPORT REVERSAL	83
6.4.1. Request	83
6.4.2. Result	83
6.5. CUSTOMER TAP	84
6.5.1. Request	84
6.5.2. Result	85
6.6. UPLOAD TRANSPORT TRANSACTIONS	86
6.6.1. Request	86
6.6.2. Result	86



6.7. CUSTOMER TAP SERVICE START	87
6.7.1. Request	87
6.7.2. Result	87
6.8. CUSTOMER TAP SERVICE STOP	88
6.8.1. Request	88
6.8.2. Result	88
6.9. GET TAPS	89
6.9.1. Request	89
6.9.2. Result	89
6.10. GET DENYLIST UPDATE	90
6.10.1. Request	90
6.10.2. Result	90
<b>7. APPLICATION DESCRIPTION</b>	<b>91</b>
7.1. Payment	91
7.2. Refund	92
7.3. Dashboard	92
7.4. The closing batch	93
7.5. Settings	94
7.6. Service Settings	95
7.6.1. Key injection	97
<b>ATTACHMENT - ENUMS</b>	<b>99</b>
TerminalErrorCodes	99
PinCommunicationsKey	112
EmvResponseData	112
EmvAdditionalResponseData	112
CvmType	112
CardInputType	113
TransactionType parameter	113
BatchTotals parameter	114







			<ul style="list-style-type: none"> <li>Added new <a href="#">TerminalErrorCodes</a> which are breaking changes             <ul style="list-style-type: none"> <li>READING_CARD_FAILED (8008)</li> <li>SECOND_TAP_CARD_MISMATCH (8009)</li> </ul> </li> <li>Adjusted transport payment request by adding additional fields and support of virtualisation: <a href="#">TRANSPORT PAYMENT</a></li> </ul> <p>Added new endpoints for:</p> <ul style="list-style-type: none"> <li><a href="#">TRANSPORT REFUND</a></li> <li><a href="#">TRANSPORT REVERSAL</a></li> </ul>
<b>6.15.0</b>	2. 6. 2025	M+(FMI)	<p>Added new endpoints for:</p> <ul style="list-style-type: none"> <li><a href="#">MOTO PREAUTHORIZATION</a></li> <li><a href="#">EXTERNAL PREAUTH INCREMENT</a></li> <li><a href="#">EXTERNAL PREAUTH DECREMENT</a></li> </ul>



		<p>New control endpoint</p> <ul style="list-style-type: none"> <li>INITIALIZE TERMINAL</li> </ul>
<b>v7</b>	4.7.0	<p>New version of TransactionResult with new fields:</p> <ul style="list-style-type: none"> <li>licensePlate</li> <li>cid</li> <li>appCryptogram</li> <li>authorizationResponseCode</li> </ul> <p>Support for new operations for Transport mode:</p> <ul style="list-style-type: none"> <li>PAYMENT Operation Extension for Virtualization</li> <li>Adding REVERSAL operation for TRANSPORT ENDPOINT</li> <li>Adding REFUND operation for TRANSPORT ENDPOINT</li> </ul> <p>Support for new operations for Retail mode:</p> <ul style="list-style-type: none"> <li>Adding READ CARD operation for RETAIL ENDPOINTS</li> <li>Adding READ CARD PAYMENT operation for RETAIL ENDPOINTS</li> <li>Adding MOTO PREAUTHORIZATION operation for RETAIL ENDPOINTS</li> <li>Adding EXTERNAL PREAUTH INCREMENT operation for RETAIL ENDPOINTS</li> <li>Adding EXTERNAL PREAUTH DECREMENT operation for RETAIL ENDPOINTS</li> </ul>

## 3. INTRODUCTION

The terminal application implements an interface through which the application can be used by third-party cashier applications on the same device.

In principle, this is an intent with specific parameters that are described in each category of this documentation.

ECR has following versions:

- ECR v1 - An outdated version of the ECR API interface.
- ECR v2 - New cash register implementations should have this version of the API.
- ECR v3 - Is to be used for the new cash registers, it introduces new field tokens and provides a better format of date for serialization.
- ECR v4 - To be used for the new cash registers. ECR V4 defines the new format of API calls, as well as new operations such as CardVerify.
- ECR v5 - Defines new transaction result with fields par and monetToken. Defines new endpoints qr\_payment and preauth\_decrement.
- ECR v6 - Defines new transaction result with optional fields dccOffer, dccResult, externalTransactionId and surcharge.
- ECR v7 - Defines new operations for transport mode (reversal and refund) and extensions to support virtualization. Defines new operations for retail mode read card and read card payment.

The application opens via Intent, which must have the following format described in the next chapter.

### 3.1. INTENT

To perform a selected operation, it is required to create an Intent with specific data.

To version ECR V4

**Action:**

```
act=android.intent.action.SENDTO
```

From version ECR V4

**Action:**

```
act=switchio.pay.ECR
```

**URI:**

See selected operation URI example.

To version ECR V4

**Intent call example:**

```
val intent = Intent ("android.intent.action.SENDTO", uri)  
startActivityForResult(intent = intent, requestCode = 123)
```

From version ECR V4

**Intent call example:**

```
val intent = Intent ("switchio.pay.ECR", uri)
```

```
startActivityForResult(intent = intent, requestCode = 123)
```

It's recommended to use flag `Intent.FLAG_ACTIVITY_REORDER_TO_FRONT` in intent to prevent multiple simultaneous instances of the app.

## 3.2. REQUEST

### To version ECR V4

```
Call
val data = Json.stringify(
    PaymentRequest.serializer(),
    PaymentRequest("1234", 1234, 203, "vs")
)

val intent = Intent("android.intent.action.SENDTO",
    Uri.parse("paya://payment?data=$data"))
startActivityForResult(intent, TERMINAL_REQUEST_CODE)

Input data structure
@Serializable
data class PaymentRequest(
    @SerializedName("TransactionID")
    val transactionId: String,
    @SerializedName("Amount")
    val amount: Int,
    @SerializedName("CurrencyCode")
    val currencyCode: Int,
    @SerializedName("InvoiceNumber")
    val invoiceNumber: String = ""
)
```

### From version ECR V4

```
Call
val data = Json.stringify(
    PaymentRequest.serializer(),
    PaymentRequest("1234", 1234, 203, "vs")
)

val uri = Uri.Builder().apply {
    scheme("app")
    authority("switchiopay")
    path("api/main/v4/payment")
    appendQueryParameter("data", data)
}.build()
val intent = Intent("switchio.pay.ECR", uri)
startActivityForResult(intent, TERMINAL_REQUEST_CODE)

Input data structure
@Serializable
data class PaymentRequest(
    @SerializedName("TransactionID")
    val transactionId: String,
    @SerializedName("Amount")
    val amount: Int,
    @SerializedName("CurrencyCode")
    val currencyCode: Int,
    @SerializedName("InvoiceNumber")
    val invoiceNumber: String = ""
)
```

### 3.3. RESULT

The operation result is returned as a **JSON** in a **Bundle** in the **Intent** data.

**Possible bundle result keys:**

transaction\_result  
data

**Bundle result value type:**

String

#### 3.3.1. Result structures and examples for each API version

Following result structures are valid for most of the requests (all payment endpoints and some control endpoints):

- [Update Terminal Configuration](#)
- [Touch – Automatic Terminal Configuration Update](#)
- [Get Last Transaction](#)
- [Get All Successful Transactions](#)
- [Open Preauthorizations](#)
- [Payment](#)
- [Refund](#)
- [Preauthorization](#)
- [Increment Preauthorization](#)
- [Decrement Preauthorization](#)
- [Complete Preauthorization](#)
- [Cancel Preauthorization](#)
- [Preauthorization External Complete](#)
- [Preauthorization External Cancel](#)
- [Get transaction](#)
- [Reversal](#)
- [MOTO](#)
- [Transaction Status](#)
- [Referral Refund](#)
- [Cashback](#)
- [QR payment](#)

Result structure v1

ENTITY	TYPE	DESCRIPTION
<b>transactionType</b>	string	Transaction types; enums in table <a href="#">TransactionType parameter</a>
<b>responseCode</b>	<a href="#">TerminalError Codes</a>	Code of the response, see <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	Message of the response
<b>sequenceNumber</b>	string	Number of the sequence

<b>authCode</b>	string	Code of the authorization
<b>reversal</b>	boolean	Information if the transaction is a reversal or not
<b>amount</b>	int	Amount of the transaction in cents. 500 is 5.00 euro
<b>invoiceNumber</b>	string	Variable symbol, max length is 20 char
<b>currency</b>	int	Code of the currency
<b>dateTimeTerminal</b>	date	Date and time on terminal, when was the result made
<b>dateTimeServer</b>	date	Date and time on server when, was the result made
<b>aid</b>	string	Application identifier
<b>pan</b>	string	Primary account number
<b>panSequenceNumber</b>	string	Sequence number of the primary account number
<b>expiration</b>	string	Expiration date of the card
<b>conto</b>	string	Card issuer
<b>appLabel</b>	string	Label of application
<b>pinCommunicationsKey</b>	<a href="#">PinCommunicationsKey</a>	Type communication key PIN, enums <a href="#">PinCommunicationsKey</a>
<b>emvResponseData</b>	<a href="#">EmvResponseData</a>	Object of the response <a href="#">EmvResponseData</a>
<b>emvAdditionalResponseData</b>	<a href="#">EmvAdditionalResponseData</a>	Object of the response <a href="#">EmvAdditionalResponseData</a>
<b>aosa</b>	int	Available offline sending amount
<b>ErrorFlag</b>	char	Error flag
<b>emvData</b>	EmvTlv	EMV data
<b>cvmTypeList</b>	List< <a href="#">CvmType</a> >	Type of the TRX. Enum is in <a href="#">CvmType</a>
<b>batchTotals</b>	<a href="#">batchTotals</a>	Object of the <a href="#">BatchTotals parameter</a>

**Result example v1:**

```
{
  "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": "TNEXG001",
    "merchantIdIssuer": "1357",
    "pan": "970348*****3910",
    "panSequenceNumber": "00",
    "responseCode": "OK",
    "responseMessage": "potvrzeno",
    "reversal": false,
    "sequenceNumber": "001001614",
    "terminalIdAcquirer": "M1TNEXG001",
    "terminalIdIssuer": "M1TNEXG001",
    "transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
    "transactionType": "SALE_ONLINE",
    "token": ["fkdfkjbejrhjlb" ] //default - empty list
  }

```

### Result structure v2

ENTITY	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>transactionType</b>	TransactionType	Transaction types; enums in table <a href="#">TransactionType parameter</a>
<b>aid</b>	string	OPTIONAL Application identifier
<b>amount</b>	int	OPTIONAL Amount of the transaction in cents. 500 is 5.00 euro
<b>appLabel</b>	string	OPTIONAL Label of application
<b>authCode</b>	string	Code of the authorization
<b>callReversal</b>	boolean	Information on whether a reversal was generated for this transaction
<b>reversal</b>	boolean	Information if the transaction is a reversal or not
<b>conto</b>	string	OPTIONAL Card issuer
<b>currency</b>	int	OPTIONAL Code of the currency
<b>dateTimeTerminal</b>	string	Date and time on terminal, when was the result made
<b>dateTimeServer</b>	string	OPTIONAL Date and time on server when, was the result made
<b>emvResponseData</b>	<a href="#">EmvResponseD</a>	OPTIONAL

	<a href="#">ata</a>	Object of the response <a href="#">EmvResponseData</a>
<b>expiration</b>	string	OPTIONAL Expiration date of the card
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>responseCode</b>	<a href="#">TerminalErrorCodes</a>	Code of the response, see <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	Message of the response
<b>terminalIdAcquirer</b>	string	OPTIONAL ID of the acquirer's terminal
<b>terminalIdIssuer</b>	string	OPTIONAL ID of the issuer's terminal
<b>merchantIdAcquirer</b>	string	OPTIONAL ID of the merchant's acquirer
<b>merchantIdIssuer</b>	string	OPTIONAL ID of the merchant's issuer
<b>pan</b>	string	OPTIONAL Primary account number
<b>token</b>	string	OPTIONAL Encrypted combination of data
<b>panSequenceNumber</b>	string	OPTIONAL Sequence number of the primary account number
<b>aosa</b>	int	OPTIONAL Available offline sending amount
<b>cardHolderMessage</b>	string	OPTIONAL Message from the server for the cardholder
<b>merchantMessage</b>	string	OPTIONAL Message from the server for the merchant
<b>availableBalance</b>	string	OPTIONAL Balance of the available funds
<b>sequenceNumber</b>	string	Number of the sequence
<b>cvmTypeList</b>	List< <a href="#">CvmType</a> >	Type of the TRX. Enum is in <a href="#">CvmType</a>
<b>cardInputType</b>	<a href="#">CardInputType</a>	OPTIONAL Type of the input. Enum is in the <a href="#">CardInputType</a>
<b>isReversed</b>	boolean	Information whether the transaction was reversed

### Result example v2:

```
{
  "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",
"token": ["fkdfkjbejrhljb" ] //default - empty list
}
```

## Result structure v3

ENTITY	TYPE	DESCRIPTION
<b>aid</b>	string	OPTIONAL Application identifier
<b>amount</b>	int	OPTIONAL Amount of the transaction in cents. 500 is 5.00 euro
<b>aosa</b>	int	OPTIONAL Available offline sending amount
<b>applabel</b>	string	OPTIONAL Label of application
<b>authCode</b>	string	OPTIONAL Code of the authorization
<b>availableBalance</b>	string	OPTIONAL Balance of the available funds
<b>cardHolderMessage</b>	string	OPTIONAL Message from the server for the cardholder
<b>cardInputType</b>	<a href="#">CardInputType</a>	OPTIONAL Type of the input. Enum is in the <a href="#">CardInputType</a>
<b>conto</b>	string	OPTIONAL Card issuer
<b>currency</b>	int	OPTIONAL Code of the currency
<b>cvmTypeList</b>	List< <a href="#">CvmType</a> >	OPTIONAL Type of the TRX. Enum is in <a href="#">CvmType</a>
<b>dateTimeServer</b>	string	OPTIONAL Date and time on server when, was the result made
<b>dateTimeTerminal</b>	string	OPTIONAL Date and time on terminal, when was the result made
<b>emvResponseData</b>	<a href="#">EmvResponseData</a>	OPTIONAL Object of the response <a href="#">EmvResponseData</a>
<b>expiration</b>	string	OPTIONAL Expiration date of the card
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char.
<b>isReversed</b>	boolean	OPTIONAL Information whether the transaction was reversed
<b>merchantIdAcquirer</b>	string	OPTIONAL ID of the merchant's acquirer
<b>merchantIdIssuer</b>	string	OPTIONAL ID of the merchant's issuer

<b>merchantMessage</b>	string	OPTIONAL Message from the server for the merchant
<b>pan</b>	string	OPTIONAL Primary account number
<b>panSequenceNumber</b>	string	OPTIONAL Sequence number of the primary account number
<b>refundId</b>	long	OPTIONAL Refund ID (added since SwitchioPay 3.1.2)
<b>responseCode</b>	<a href="#">TerminalErrorCodes</a>	OPTIONAL Code of the response, see <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	OPTIONAL Message of the response
<b>reversal</b>	boolean	OPTIONAL Information if the transaction is a reversal or not
<b>sequenceNumber</b>	string	OPTIONAL Number of the sequence
<b>terminalIdAcquirer</b>	string	OPTIONAL ID of the acquirer's terminal
<b>terminalIdIssuer</b>	string	OPTIONAL ID of the issuer's terminal
<b>token</b>	string	Encrypted combination of data
<b>transactionId</b>	string	ID of the transaction.
<b>transactionType</b>	string	Transaction types; enums in table <a href="#">TransactionType parameter</a>

### Result example v3:

```
{
  "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": "TNEXG001",
  "merchantIdIssuer": "1357",
}
```

```

"pan": "970348*****3910",
"panSequenceNumber": "00",
"responseCode": "OK",
"responseMessage": "potvrzeno",
"reversal": false,
"sequenceNumber": "001001614",
"terminalIdAcquirer": "M1TNEXG001",
"terminalIdIssuer": "M1TNEXG001",
"transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
"transactionType": "SALE_ONLINE",
"token": ["fkdfkjbekjrhbjb"] //default - empty list
}

```

Result structure v4

PARAMETER	TYPE	DESCRIPTION
<b>aid</b>	string	OPTIONAL Application identifier.
<b>amount</b>	int	OPTIONAL Amount is the amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>aosa</b>	int	OPTIONAL Available offline sending amount.
<b>applLabel</b>	string	OPTIONAL Label of application.
<b>authCode</b>	string	OPTIONAL Code of the authorization.
<b>availableBalance</b>	string	OPTIONAL Balance of the available funds.
<b>brand</b>	string	OPTIONAL
<b>cardInputType</b>	string	OPTIONAL Type of the input. Enum is in the <a href="#">CardInputType</a>
<b>cardHolderMessage</b>	string	OPTIONAL Card issuer.
<b>currency</b>	int	OPTIONAL ISO 4217 currency code.
<b>cvmTypeList</b>	string array	OPTIONAL Type of the transaction. <a href="#">CvmType</a>
<b>dateTimeServer</b>	string	OPTIONAL Date and time on the server when the result was made.
<b>dateTimeTerminal</b>	string	OPTIONAL Date and time on the terminal when the

		result was made.
<b>emvResponseData</b>	object	OPTIONAL Object of the response. <a href="#">EmvResponseData</a>
<b>smartCardScheme</b>	string	OPTIONAL Scheme of the smart card.
<b>authorisationResponseCode</b>	string	OPTIONAL The response code of authorization.
<b>expiration</b>	string	OPTIONAL Expiration date of the card.
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char.
<b>isReversed</b>	boolean	OPTIONAL Information whether the transaction was reversed.
<b>merchantIdAcquirer</b>	string	OPTIONAL ID of the merchant's acquirer.
<b>merchantIdIssuer</b>	string	OPTIONAL ID of the merchant's issuer.
<b>merchantMessage</b>	string	OPTIONAL Message for the merchant
<b>pan</b>	string	OPTIONAL Primary account number.
<b>panSequenceNumber</b>	string	OPTIONAL Sequence number of the primary account number.
<b>refundId</b>	string	OPTIONAL ID of the refund transaction
<b>responseCode</b>	string	OPTIONAL Code of the response. <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	OPTIONAL Message of the response.
<b>reversal</b>	boolean	OPTIONAL Information if the transaction is a reversal or not.
<b>sequenceNumber</b>	string	OPTIONAL Number of the sequence.
<b>terminalIdAcquirer</b>	string	OPTIONAL ID of the acquirer's terminal.
<b>terminalIdIssuer</b>	string	OPTIONAL ID of the issuer's terminal.
<b>tipAmount</b>	integer	OPTIONAL Amount of the tip
<b>transactionId</b>	string	ID of the transaction.

<b>transactionType</b>	string	Transaction type. <a href="#">TransactionTypes</a>
<b>tokens</b>	string array	OPTIONAL Encrypted combination of data.

#### Result example v4:

```
{
  "aid": "A0000000283101",
  "amount": 500,
  "aosa": "null",
  "appLabel": "Air Bank",
  "authCode": "114546",
  "availableBalance": "1000",
  "brand": "MASTERCARD",
  "cardHolderMessage": "Approved",
  "cardInputType": "CLESS",
  "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": "TNEGXG001",
  "merchantIdIssuer": "1357",
  "merchantMessage": "Approved",
  "pan": "970348*****3910",
  "panSequenceNumber": "00",
  "refundId": "1234567890",
  "responseCode": "OK",
  "responseMessage": "potvrzeno",
  "reversal": false,
  "sequenceNumber": "001001614",
  "tipAmount": 100,
  "terminalIdAcquirer": "M1TNEGXG001",
  "terminalIdIssuer": "M1TNEGXG001",
  "transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
  "transactionType": "SALE_ONLINE",
  "tokens": ["fkdfkjbejrhjlb"] //default - empty list
}
```

#### Obtain result example:

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == 123) {
        val response = data?.extras?.getString("transaction_result")
        ...
    }
}
```

## Result structure v5

PARAMETER	TYPE	DESCRIPTION
<b>aid</b>	string	OPTIONAL Application identifier.
<b>amount</b>	int	OPTIONAL Amount is the amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>aosa</b>	int	OPTIONAL Available offline sending amount.
<b>applabel</b>	string	OPTIONAL Label of application.
<b>authCode</b>	string	OPTIONAL Code of the authorization.
<b>availableBalance</b>	string	OPTIONAL Balance of the available funds.
<b>brand</b>	string	OPTIONAL
<b>cardInputType</b>	string	OPTIONAL Type of the input. Enum is in the <a href="#">CardInputType</a>
<b>cardToken</b>	string	OPTIONAL
<b>cardHolderMessage</b>	string	OPTIONAL Card issuer.
<b>currency</b>	int	OPTIONAL ISO 4217 currency code.
<b>cvmTypeList</b>	string array	OPTIONAL Type of the transaction. <a href="#">CvmType</a>
<b>dateTimeServer</b>	string	OPTIONAL Date and time on the server when the result was made.
<b>dateTimeTerminal</b>	string	OPTIONAL Date and time on the terminal when the result was made.
<b>emvResponseData</b>	object	OPTIONAL Object of the response. <a href="#">EmvResponseData</a>
<b>smartCardScheme</b>	string	OPTIONAL Scheme of the smart card.
<b>authorisationResponseCode</b>	string	OPTIONAL The response code of authorization.
<b>expiration</b>	string	OPTIONAL Expiration date of the card.

<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char.
<b>isReversed</b>	boolean	OPTIONAL Information whether the transaction was reversed.
<b>merchantIdAcquirer</b>	string	OPTIONAL ID of the merchant's acquirer.
<b>merchantIdIssuer</b>	string	OPTIONAL ID of the merchant's issuer.
<b>merchantMessage</b>	string	OPTIONAL Message for the merchant
<b>monetToken</b>	string	OPTIONAL
<b>pan</b>	string	OPTIONAL Primary account number.
<b>par</b>	string	OPTIONAL
<b>panSequenceNumber</b>	string	OPTIONAL Sequence number of the primary account number.
<b>refundId</b>	string	OPTIONAL ID of the refund transaction
<b>responseCode</b>	string	OPTIONAL Code of the response. <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	OPTIONAL Message of the response.
<b>reversal</b>	boolean	OPTIONAL Information if the transaction is a reversal or not.
<b>sequenceNumber</b>	string	OPTIONAL Number of the sequence.
<b>terminalIdAcquirer</b>	string	OPTIONAL ID of the acquirer's terminal.
<b>terminalIdIssuer</b>	string	OPTIONAL ID of the issuer's terminal.
<b>tipAmount</b>	integer	OPTIONAL Amount of the tip
<b>transactionId</b>	string	ID of the transaction.
<b>transactionType</b>	string	Transaction type. <a href="#">TransactionTypes</a>
<b>tokens</b>	string array	OPTIONAL Encrypted combination of data.

## Result structure v6

PARAMETER	TYPE	DESCRIPTION
<b>aid</b>	string	OPTIONAL Application identifier.
<b>amount</b>	int	OPTIONAL Amount is the amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point). It is the total amount that includes the surcharge amount as well as the tip amount (if applied on the transaction).
<b>aosa</b>	int	OPTIONAL Available offline sending amount.
<b>applLabel</b>	string	OPTIONAL Label of application.
<b>authCode</b>	string	OPTIONAL Code of the authorization.
<b>availableBalance</b>	string	OPTIONAL Balance of the available funds.
<b>brand</b>	string	OPTIONAL
<b>cardInputType</b>	string	OPTIONAL Type of the input. Enum is in the <a href="#">CardInputType</a>
<b>cardToken</b>	string	OPTIONAL
<b>cardHolderMessage</b>	string	OPTIONAL Card issuer.
<b>currency</b>	int	OPTIONAL ISO 4217 currency code.
<b>cvmTypeList</b>	string array	OPTIONAL Type of the transaction. <a href="#">CvmType</a>
<b>dateTimeServer</b>	string	OPTIONAL Date and time on the server when the result was made.
<b>dateTimeTerminal</b>	string	OPTIONAL Date and time on the terminal when the result was made.
<b>dccOffer</b>	object	OPTIONAL dccOffer object
<b>type</b>	string	Card type. Possible values: <ul style="list-style-type: none"> <li>• VISA</li> <li>• MASTERCARD</li> <li>• UNKNOWN</li> </ul>

<b>currency</b>	string	Name of the DCC currency
<b>amountExponent</b>	int	Amount exponent. Indicates how many decimal places the amount is calculated.
<b>convertedAmount</b>	long	Converted amount
<b>rateExponent</b>	int	DCC rate exponent. Indicates how many decimal places the currency is calculated.
<b>rate</b>	long	Conversion rate
<b>language</b>	string	Language code
<b>ecbRateExponent</b>	int	European Central Bank rate exponent. Indicates how many decimal places the currency is calculated.
<b>ecbRate</b>	long	European Central Bank rate.
<b>markup</b>	int	Markup amount
<b>currencyCode</b>	int	Code of the DCC currency
<b>dccResult</b>	string	OPTIONAL DCC result. Possible values: <ul style="list-style-type: none"> <li>● PERFORMED</li> <li>● NOT_PERFORMED</li> </ul>
<b>emvResponseData</b>	object	OPTIONAL Object of the response. <a href="#">EmvResponseData</a>
<b>smartCardScheme</b>	string	OPTIONAL Scheme of the smart card.
<b>authorisationResponseCode</b>	string	OPTIONAL The response code of authorization.
<b>expiration</b>	string	OPTIONAL Expiration date of the card.
<b>externalTransactionId</b>	string	OPTIONAL Transaction ID in an external system.
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char.
<b>isReversed</b>	boolean	OPTIONAL Information whether the transaction was reversed.
<b>merchantIdAcquirer</b>	string	OPTIONAL ID of the merchant's acquirer.
<b>merchantIdIssuer</b>	string	OPTIONAL ID of the merchant's issuer.
<b>merchantMessage</b>	string	OPTIONAL Message for the merchant
<b>monetToken</b>	string	OPTIONAL

<b>pan</b>	string	OPTIONAL Primary account number.
<b>par</b>	string	OPTIONAL
<b>panSequenceNumber</b>	string	OPTIONAL Sequence number of the primary account number.
<b>refundId</b>	string	OPTIONAL ID of the refund transaction
<b>responseCode</b>	string	OPTIONAL Code of the response. <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	OPTIONAL Message of the response.
<b>reversal</b>	boolean	OPTIONAL Information if the transaction is a reversal or not.
<b>sequenceNumber</b>	string	OPTIONAL Number of the sequence.
<b>surcharge</b>	int	OPTIONAL Surcharged amount
<b>terminalIdAcquirer</b>	string	OPTIONAL ID of the acquirer's terminal.
<b>terminalIdIssuer</b>	string	OPTIONAL ID of the issuer's terminal.
<b>tipAmount</b>	integer	OPTIONAL Amount of the tip
<b>transactionId</b>	string	ID of the transaction.
<b>transactionType</b>	string	Transaction type. <a href="#">TransactionTypes</a>
<b>tokens</b>	string array	OPTIONAL Encrypted combination of data.

**Result example v6:**

```
{
  "aid": "A0000000283101",
  "amount": 500,
  "aosa": "null",
  "appLabel": "Air Bank",
  "authCode": "114546",
  "availableBalance": "1000",
  "brand": "MASTERCARD",
  "cardHolderMessage": "Approved",
  "cardInputType": "CLESS",
  "currency": 203,
  "cvmTypeList": [
    "PIN_ONLINE"
  ],
  "dateTimeServer": "Jul 16, 2020 14:40:50",
  "dateTimeTerminal": "Jul 16, 2020 14:39:37",
  "dccOffer": {
```

```
"type": "VISA",
"currency": "USD",
"amountExponent": 2,
"convertedAmount": 11556691,
"rateExponent": 6,
"rate": 1155680,
"language": "us",
"ecbRateExponent": 0,
"ecbRate": 0,
"markup": 375,
"currencyCode": 840
},
"dccResult": "PERFORMED",
"emvResponseData": {
  "smartCardScheme": "1",
  "authorisationResponseCode": "00"
},
"expiration": "****",
"externalTransactionId": "5043365e4faa-48a3ea7d-1357",
"invoiceNumber": "VS",
"isReversed": false,
"merchantIdAcquirer": "TNEXG001",
"merchantIdIssuer": "1357",
"merchantMessage": "Approved",
"monetToken": "MONETTOKEN",
"pan": "970348*****3910",
"par": "PAR",
"panSequenceNumber": "00",
"refundId": "1234567890",
"responseCode": "OK",
"responseMessage": "potvrzeno",
"reversal": false,
"sequenceNumber": "001001614",
"surcharge": 120,
"tipAmount": 100,
"terminalIdAcquirer": "M1TNEXG001",
"terminalIdIssuer": "M1TNEXG001",
"tokens": ["fkdfkjbejrhljb"],
"transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
"transactionType": "SALE_ONLINE"
}
```

## Result structure v7

PARAMETER	TYPE	DESCRIPTION
<b>aid</b>	string	OPTIONAL Application identifier.
<b>amount</b>	int	OPTIONAL Amount is the amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point). It is the total amount that includes the surcharge amount as well as the tip amount (if applied on the transaction).
<b>aosa</b>	int	OPTIONAL Available offline sending amount.
<b>appCryptogram</b>	string	OPTIONAL Cryptogram returned by the ICC in response of the GENERATE AC command
<b>applLabel</b>	string	OPTIONAL Label of application.
<b>authCode</b>	string	OPTIONAL Code of the authorization.
<b>availableBalance</b>	string	OPTIONAL Balance of the available funds.
<b>brand</b>	string	OPTIONAL
<b>cardInputType</b>	string	OPTIONAL Type of the input. Enum is in the <a href="#">CardInputType</a>
<b>cardToken</b>	string	OPTIONAL
<b>cid</b>	string	OPTIONAL Cryptogram Information Data Values: <ul style="list-style-type: none"> <li>• <b>00 (AAC)</b> - Application Authentication Cryptogram - A cryptogram generated by the card at the end of offline and online declined transactions</li> <li>• <b>40(TC)</b> - Transaction Certificate-the result of card, terminal, and transaction data encrypted by a DES key</li> <li>• <b>80 (ARQC)</b> - Authorization Request. Cryptogram-A cryptogram used for a process called Online Card Authentication.</li> </ul>
<b>cardHolderMessage</b>	string	OPTIONAL Card issuer.
<b>currency</b>	int	OPTIONAL ISO 4217 currency code.
<b>cvmTypeList</b>	string array	OPTIONAL

		Type of the transaction. <a href="#">CvmType</a>
<b>dateTimeServer</b>	string	OPTIONAL Date and time on the server when the result was made.
<b>dateTimeTerminal</b>	string	OPTIONAL Date and time on the terminal when the result was made.
<b>dccOffer</b>	object	OPTIONAL dccOffer object
<b>type</b>	string	Card type. Possible values: <ul style="list-style-type: none"> <li>• VISA</li> <li>• MASTERCARD</li> <li>• UNKNOWN</li> </ul>
<b>currency</b>	string	Name of the DCC currency
<b>amountExponent</b>	int	Amount exponent. Indicates how many decimal places the amount is calculated.
<b>convertedAmount</b>	long	Converted amount
<b>rateExponent</b>	int	DCC rate exponent. Indicates how many decimal places the currency is calculated.
<b>rate</b>	long	Conversion rate
<b>language</b>	string	Language code
<b>ecbRateExponent</b>	int	European Central Bank rate exponent. Indicates how many decimal places the currency is calculated.
<b>ecbRate</b>	long	European Central Bank rate.
<b>markup</b>	int	Markup amount
<b>currencyCode</b>	int	Code of the DCC currency
<b>dccResult</b>	string	OPTIONAL DCC result. Possible values: <ul style="list-style-type: none"> <li>• PERFORMED</li> <li>• NOT_PERFORMED</li> </ul>
<b>driverNumber</b>	string	OPTIONAL ID number of the driver.
<b>emvResponseData</b>	object	OPTIONAL Object of the response. <a href="#">EmvResponseData</a>
<b>smartCardScheme</b>	string	OPTIONAL Scheme of the smart card.
<b>authorisationResponseCode</b>	string	OPTIONAL The response code of authorization.
<b>expiration</b>	string	OPTIONAL Expiration date of the card.
<b>externalTransactionId</b>	string	OPTIONAL

		Transaction ID in an external system.
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char.
<b>isReversed</b>	boolean	OPTIONAL Information whether the transaction was reversed.
<b>merchantIdAcquirer</b>	string	OPTIONAL ID of the merchant's acquirer.
<b>merchantIdIssuer</b>	string	OPTIONAL ID of the merchant's issuer.
<b>merchantMessage</b>	string	OPTIONAL Message for the merchant
<b>monetToken</b>	string	OPTIONAL
<b>odometer</b>	long	OPTIONAL Odometer reading (current mileage value).
<b>pan</b>	string	OPTIONAL Primary account number.
<b>par</b>	string	OPTIONAL
<b>panSequenceNumber</b>	string	OPTIONAL Sequence number of the primary account number.
<b>refundId</b>	string	OPTIONAL ID of the refund transaction
<b>responseCode</b>	string	OPTIONAL Code of the response. <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	OPTIONAL Message of the response.
<b>reversal</b>	boolean	OPTIONAL Information if the transaction is a reversal or not.
<b>sequenceNumber</b>	string	OPTIONAL Number of the sequence.
<b>surcharge</b>	int	OPTIONAL Surcharged amount
<b>terminalIdAcquirer</b>	string	OPTIONAL ID of the acquirer's terminal.
<b>terminalIdIssuer</b>	string	OPTIONAL ID of the issuer's terminal.
<b>tipAmount</b>	integer	OPTIONAL Amount of the tip
<b>transactionId</b>	string	ID of the transaction.
<b>transactionType</b>	string	Transaction type. <a href="#">TransactionTypes</a>

<b>tokens</b>	string array	OPTIONAL Encrypted combination of data.
<b>vehicleCode</b>	string	OPTIONAL Code of the vehicle.
<b>driverNumber</b>	String	OPTIONAL A unique identifier assigned to the driver operating the vehicle during the transaction.
<b>licensePlate</b>	String	OPTIONAL The official registration number of the vehicle, typically displayed on the vehicle's plates.
<b>authorizationResponseCode</b>	String	A two-character code returned by the issuer that indicates the result of the authorization request.

### Result example v7:

```
{
  "aid": "A0000000283101",
  "amount": 500,
  "cashbackAmount": 1500,
  "aosa": "null",
  "appCryptogram": "5F52BA71C7B45441",
  "appLabel": "Air Bank",
  "authCode": "114546",
  "availableBalance": "1000",
  "brand": "MASTERCARD",
  "cardHolderMessage": "Approved",
  "cardInputType": "CLESS",
  "cid": "80",
  "currency": 203,
  "cvmTypeList": ["PIN_ONLINE"],
  "dateTimeServer": "Jul 16, 2020 14:40:50",
  "dateTimeTerminal": "Jul 16, 2020 14:39:37",
  "driverNumber": "123456",
  "emvResponseData": {
    "smartCardScheme": "1",
    "authorisationResponseCode": "00"
  },
  "expiration": "****",
  "invoiceNumber": "VS",
  "isReversed": false,
  "merchantIdAcquirer": "TNEXGO01",
  "merchantIdIssuer": "1357",
  "merchantMessage": "Approved",
  "monetToken": "MONETTOKEN",
  "odometer": 65403,
  "pan": "970348*****3910",
  "par": "PAR",
  "panSequenceNumber": "00",
  "refundId": "1234567890",
  "responseCode": "OK",
  "responseMessage": "potvrzeno",
  "reversal": false,
  "sequenceNumber": "001001614",
  "tipAmount": 100,
}
```

```

"terminalIdAcquirer": "M1TNEXGO01",
"terminalIdIssuer": "M1TNEXGO01",
"tokens": ["fkdfkjbejrhljb"] //default - empty list
"transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
"transactionType": "SALE_ONLINE",
"vehicleCode": "1A21234",
"driverNumber": "DRIVER1",
"licensePlate": "EL100AC",
"authorizationResponseCode": "00",
}

```

## 4. CONTROL ENDPOINTS

### 4.1. CLOSING BATCH

The closing batch request triggers the closing of a batch of transactions for the day. It should be done daily after the shift is finished.

#### 4.1.1. Request

This operation has no request data.

**ECR v1: URI = "paya://closing\_batch"**

**ECR v2: URI = "paya://closing\_batch\_v2"**

**ECR v3: URI = "paya://closing\_batch\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/closure"**

**ECR v5: URI = "app://switchiopay/api/main/v5/closure"**

**ECRv6: URI = "app://switchiopay/api/main/v6/closure"**

**ECRv7: URI = "app://switchiopay/api/main/v7/closure"**

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://closing_batch
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/closure
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/closing\_batch
- parameter value: request json

## 4.1.2. Result

### Result structure:

PARAMETER	TYPE	DESCRIPTION
<b>transactionType</b>	string	Type of the transaction. In this case it is CLOSE BATCH
<b>dateTimeServer</b>	string	OPTIONAL Date and time on the server.
<b>dateTimeTerminal</b>	string	Date and time on the terminal.
<b>responseCode</b>	string	Response code. <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	OPTIONAL
<b>batchTotals</b>	object	
<b>shiftNumber</b>	int	Shift number.
<b>batchNumber</b>	int	Batch number.
<b>debitsCount</b>	int	Count of payment transactions.
<b>debitsAmount</b>	int	Amount of payment transactions.
<b>creditsCount</b>	int	Count of refund transactions.
<b>creditsAmount</b>	int	Amount of refund transactions.
<b>adjsCount</b>	int	Count of adjustments.
<b>adjsAmount</b>	int	Amount of adjustments.

### Result example:

```
{
  "transactionType": "CLOSE_BATCH",
  "dateTimeTerminal": "Jul 16, 2020 14:40:50",
  "dateTimeServer": "Jul 16, 2020 14:41:10",
  "responseCode": "OK",
  "responseMessage": "Success",
  "batchTotals": {
    "shiftNumber": 1,
    "batchNumber": 1,
    "debitsCount": 1000,
    "debitsAmount": 5,
    "creditsCount": 1000,
    "creditsAmount": 5,
    "adjsCount": 1000,
    "adjsAmount": 5
  }
}
```

## 4.2. DAUGHTER COMPANIES

Get list of available daughter companies if virtualization is enabled.

### 4.2.1. Request

This operation has no request data.

**ECR v1: URI = "paya://daughter\_companies"**  
**ECR v2: URI = "paya://daughter\_companies\_v2"**  
**ECR v3: URI = "paya://daughter\_companies\_v3"**  
**ECR v4: URI = "app://switchiopay/api/light/v4/daughter\_companies"**  
**ECR v5: URI = "app://switchiopay/api/light/v5/daughter\_companies"**  
**ECR v6: URI = "app://switchiopay/api/light/v6/daughter\_companies"**  
**ECR v7: URI = "app://switchiopay/api/light/v7/daughter\_companies"**

Bundle result key = "daughter\_companies"

#### Example request for ECR v1:

```
paya://daughter_companies
```

#### Example request for ECR v4:

```
app://switchiopay/api/light/v4/daughter_companies
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v4/daughter\_companies

### 4.2.2. Result

#### Result structure:

PARAMETER	TYPE	DESCRIPTION
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

#### Result example:

```
[
  {
    "daughterCompanyId": "COMPANY1"
  },
  {
    "daughterCompanyId": "COMPANY2"
  },
  {
    "daughterCompanyId": "COMPANY3"
  }
]
```

## 4.3. SCANNER

Scans a QR code and returns a raw unformatted result.

### 4.3.1. Request

This operation has no request data.

**ECR v1: URI = "paya://scan"**

**ECR v2: URI = "paya://scan\_v2"**

**ECR v3: URI = "paya://scan\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/scan"**

**ECR v5: URI = "app://switchiopay/api/main/v5/scan"**

**ECR v6: URI = "app://switchiopay/api/main/v6/scan"**

**ECR v7: URI = "app://switchiopay/api/main/v7/scan"**

Bundle result key = "data"

#### Example request for ECR v1:

```
paya://scan
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/scan
```

Additional info for building uri with `Uri.Builder()`, available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/scan

### 4.3.2. Result

#### Result structure:

```
<string>
```

#### Result example:

```
www.google.com
```

## 4.4. UPDATE TERMINAL CONFIGURATION

Call the TMS and download a new terminal configuration.

### 4.4.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>login</b>	string	Login of the terminal.
<b>password</b>	string	Password of the terminal.
<b>host</b>	string	IP address/hostname of the TMS server.
<b>port</b>	int	Port that the TMS server is listening on.

**ECR v1: URI = "paya://tms\_update"**

**ECR v2: URI = "paya://tms\_update"**

**ECR v3: URI = "paya://tms\_update"**

**ECR v4: URI = "app://switchiopay/api/main/v4/tms\_update"**

**ECR v5: URI = "app://switchiopay/api/config/v5/tms\_update"**

**ECR v6: URI = "app://switchiopay/api/config/v6/tms\_update"**

**ECR v7: URI = "app://switchiopay/api/config/v7/tms\_update"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://tms_update?data={"login":"TERMINAL01", "password": "123456789",
"host":"127.0.0.1", "port":7777}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/tms_update?data={"login":"TERMINAL01",
"password": "123456789", "host":"127.0.0.1", "port":7777}
```

#### Example request for ECR v5:

```
app://switchiopay/api/config/v5/tms_update?data={"login":"TERMINAL01",
"password": "123456789", "host":"127.0.0.1", "port":7777}
```

Additional info for building uri with Uri.Builder(), available for ECR v4:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/tms\_update
- parameter key: data
- parameter value: request json

Additional info for building uri with Uri.Builder(), available for ECR v5:

- scheme: app
- authority: switchiopay
- api: config
- path: /api/config/v5/tms\_update
- parameter key: data



- parameter value: request json

#### 4.4.2. Result

See [result structure](#).

##### Result example:

```
{
  "transactionId": "",
  "transactionType": "UNDEFINED",
  "reversal": false,
  "dateTimeTerminal": "Apr 30, 2023 02:06:01 PM",
  "responseCode": "OK",
  "responseMessage": "TMS MANUAL call was success."
}
```

## 4.5. TOUCH - AUTOMATIC CONFIGURATION UPDATE

Starting with version 2.2.0, Switchio Pay application supports automatic TMS configuration updates. Application periodically checks its configuration state with the TMS and broadcasts the status. In the ECR mode, Switchio Pay application handles only one event by itself - UpdateImmediately. Every other event should be handled by the cash register.

### 4.5.1. Broadcast

Broadcast action = "paya.tmsparam.status"

Broadcast bundle key = "tmsParamStatus"

#### Status:

STATUS	DESCRIPTION
<b>UpToDate</b>	No action needed.
<b>UpdateRequired</b>	Terminal configuration should be updated as soon as possible.
<b>UpdateImmediately</b>	Terminal configuration should be updated immediately. Switchio Pay handles this event by itself, the cash register does not need to implement this. Switchio Pay will not interrupt any payment process with configuration updates.
<b>UpdateAfterClosing</b>	Terminal configuration should be updated after a successful closing.
<b>UpdateAtSpecificTime</b>	Terminal configuration should be updated at the provided time. (hour: <integer>, minute: <integer>)
<b>Undefined</b>	Unknown data received from the server.
<b>UndefinedInvalidData</b>	Invalid data received from the server.

#### Broadcast receiver example:

```
class TmsParamStatusBroadcastReceiver @Inject constructor() : BroadcastReceiver() {
    lateinit var listener: (TmsParamStatus) -> Unit
    override fun onReceive(context: Context, intent: Intent) {
        if (intent.action == ACTION) {
            val tmsParamStatusData = intent.extras?.getString(KEY_STATUS) ?: return
            val tmsParamStatus = Json.decodeFromString(TmsParamStatus.serializer(),
tmsParamStatusData)
            listener(tmsParamStatus)
        }
    }
    companion object {
        private const val ACTION = "paya.tmsparam.status"
        private const val KEY_STATUS = "tmsParamStatus"
    }
}
```

Upon receiving a TMS param status, the cash register should schedule the TMS configuration update. Switchio Pay provides an API for TMS configuration update without the need of providing credentials.

## 4.5.2. Request

This operation has no request data.

**ECR v1: URI = "paya://tms\_auto\_update"**

**ECR v2: URI = "paya://tms\_auto\_update\_v2"**

**ECR v3: URI = "paya://tms\_auto\_update\_v3"**

**ECR v4: URI = "app://switchiopay/api/config/v4/tms\_auto\_update"**

**ECR v5: URI = "app://switchiopay/api/config/v5/tms\_auto\_update"**

**ECR v6: URI = "app://switchiopay/api/config/v6/tms\_auto\_update"**

**ECR v7: URI = "app://switchiopay/api/config/v7/tms\_auto\_update"**

Bundle result key = "transaction\_result"

### Example request for ECR v1:

```
paya://tms_auto_update
```

### Example request for ECR v4:

```
app://switchiopay/api/config/v4/tms_auto_update
```

Additional info for building uri with `Uri.Builder()`, available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: config
- path: /api/config/v4/tms\_auto\_update
- parameter value: request json

## 4.5.3. Result

See [result structure](#).

### Result example:

```
{
  "transactionId": "",
  "transactionType": "UNDEFINED",
  "reversal": false,
  "dateTimeTerminal": "Apr 30, 2023 02:06:01 PM",
  "responseCode": "OK",
  "responseMessage": "TMS MANUAL call was success."
}
```

## 4.6. SUBTOTAL

Returns the current subtotal of the transactions made. It can be done any time of the day and it does not trigger closing.

### 4.6.1. Request

This operation has no request data.

**ECR v1: URI = "paya://subtotal"**

**ECR v2: URI = "paya://subtotal\_v2"**

**ECR v3: URI = "paya://subtotal\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/subtotal"**

**ECR v5: URI = "app://switchiopay/api/main/v5/subtotal"**

**ECR v6: URI = "app://switchiopay/api/main/v6/subtotal"**

**ECR v7: URI = "app://switchiopay/api/main/v7/subtotal"**

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://subtotal
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/subtotal
```

Additional info for building uri with `Uri.Builder()`, available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/subtotal
- parameter value: request json

### 4.6.2. Result

See [Closing batch](#).

## 4.7. HANDSHAKE

Handshake with the authorization server.

### 4.7.1. Request

This operation has no request data.

**ECR v1: URI = "paya://handshake"**  
**ECR v2: URI = "paya://handshake\_v2"**  
**ECR v3: URI = "paya://handshake\_v3"**  
**ECR v4: URI = "app://switchiopay/api/main/v4/handshake"**  
**ECR v5: URI = "app://switchiopay/api/main/v5/handshake"**  
**ECR v6: URI = "app://switchiopay/api/main/v6/handshake"**  
**ECR v7: URI = "app://switchiopay/api/main/v7/handshake"**

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://handshake
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/handshake
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/handshake
- parameter value: request json

### 4.7.2. Result

#### Result structure:

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>transactionType</b>	string	Type of the transaction. In this case, it is HANDSHAKE.
<b>responseCode</b>	string	Code of the response. <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	Message of the response.

#### Result example:

```
{
  "transactionId": "89555389-4f67-4cb0-ae8-544ea11dbd1c",
  "transactionType": "HANDSHAKE",
  "terminalErrorCodes": "OK",
  "responseMessage": "Success"
}
```

## 4.8. TOKEN

ECR protocol supports a token generator running in the background. Application has control of turning on and off the card detection and a token generation via the ECR protocol. Once the token generator is turned on, the token will be broadcasted from the payment application. The client application is able to receive the token by implementing the BroadcastReceiver. After the response from the payment application, start the generator again to generate another token.

Make sure you stop the token generator as well as unregister the broadcast receiver after you are done to prevent excessive battery usage and memory leaks.

### 4.8.1. Request

```
start  
stop
```

**ECR v1: URI = "paya://token"**

**ECR v2: URI = "paya://token\_v2"**

**ECR v3: URI = "paya://token\_v3"**

**ECR v4: URI = "app://switchiopay/api/light/v4/token"**

**ECR v5: URI = "app://switchiopay/api/light/v5/token"**

**ECR v6: URI = "app://switchiopay/api/light/v6/token"**

**ECR v7: URI = "app://switchiopay/api/light/v7/token"**

Bundle result key = "data"

#### Example request for ECR v1:

```
paya://token?action=start  
paya://token?action=stop
```

#### Example request for ECR v4:

```
app://switchiopay/api/light/v4/token?action=start  
app://switchiopay/api/light/v4/token?action=stop
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v4/token
- parameter value: request json

## 4.8.2. Result

### Result structure:

PARAMETER	TYPE	DESCRIPTION
resultCode	int	0 - Success 1 - Unknown error 2 - Unknown card
maskedCardNumber	string	Partially hidden card number
tokens	string array	Encrypted combination of data.
expiration	string	Card expiration date.
cardUid	string	Optional - card chip number

### Result example v1:

```
{
  "resultCode": 0,
  "maskedCardNumber": "970348*****3910",
  "tokens": ["8798F7W98723KDDGF08723HKDJHF897"],
}
```

### Result example v4:

```
{
  "resultCode": 0,
  "maskedCardNumber": "970348*****3910",
  "tokens": ["8798F7W98723KDDGF08723HKDJHF897"],
  "expiration": "****",
  "cardUid": "a1234567890"
}
```

## 4.9. GET LAST TRANSACTION

Request returns last transaction from the terminal.

### 4.9.1. Request

This operation has no request data.

**ECR v1: URI = "paya://get\_last\_transaction"**

**ECR v2: URI = "paya://get\_last\_transaction\_v2"**

**ECR v3: URI = "paya://get\_last\_transaction\_v3"**

**ECR v4: URI = "app://switchiopay/api/light/v4/get\_last\_transaction"**

**ECR v5: URI = "app://switchiopay/api/light/v5/get\_last\_transaction"**

**ECR v6: URI = "app://switchiopay/api/light/v6/get\_last\_transaction"**

**ECR v7: URI = "app://switchiopay/api/light/v7/get\_last\_transaction"**

Bundle result key = "transaction\_result"

### Example request for ECR v1:

```
paya://get_last_transaction
```

**Example request for ECR v4:**

```
app://switchiopay/api/light/v4/get_last_transaction
```

Additional info for building uri with `Uri.Builder()`, available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v4/get\_last\_transaction
- parameter value: request json

**4.9.2. Result**

See [result structure](#).

**Result example:**

```
{
  "aid": "A0000000283101",
  "amount": 500,
  "aosa": "null",
  "appLabel": "Air Bank",
  "authCode": "114546",
  "availableBalance": "1000",
  "brand": "MASTERCARD",
  "cardHolderMessage": "Approved",
  "cardInputType": "CLESS",
  "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": "TNEXG001",
  "merchantIdIssuer": "1357",
  "merchantMessage": "Approved",
  "pan": "970348*****3910",
  "panSequenceNumber": "00",
  "refundId": "1234567890",
  "responseCode": "OK",
  "responseMessage": "potvrzeno",
  "reversal": false,
  "sequenceNumber": "001001614",
  "tipAmount": 100,
  "terminalIdAcquirer": "M1TNEXG001",
  "terminalIdIssuer": "M1TNEXG001",
  "transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
  "transactionType": "SALE_ONLINE",
  "tokens": ["fkdfkjbejrhljb"] //default - empty list
}
```

## 4.10. GET ALL SUCCESSFUL TRANSACTIONS

Application returns list of successful transactions from the terminal.

### 4.10.1. Request

This operation has no request data.

**ECR v1: URI = "paya://get\_transactions"**

**ECR v2: URI = "paya://get\_transactions\_v2"**

**ECR v3: URI = "paya://get\_transactions\_v3"**

**ECR v4: URI = "app://switchiopay/api/light/v4/get\_transactions"**

**ECR v5: URI = "app://switchiopay/api/light/v5/get\_transactions"**

**ECR v6: URI = "app://switchiopay/api/light/v6/get\_transactions"**

**ECR v7: URI = "app://switchiopay/api/light/v7/get\_transactions"**

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://get_transactions
```

#### Example request for ECR v4:

```
app://switchiopay/api/light/v4/get_transactions
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v4/get\_transactions
- parameter value: request json

## 4.10.2. Result

Array of [result structure](#).

### Result example:

```
[
{
  "aid": "A0000000283101",
  "amount": 500,
  "aosa": "null",
  "appLabel": "Air Bank",
  "authCode": "114546",
  "availableBalance": "1000",
  "brand": "MASTERCARD",
  "cardHolderMessage": "Approved",
  "cardInputType": "CLESS",
  "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",
  "merchantMessage": "Approved",
  "pan": "970348*****3910",
  "panSequenceNumber": "00",
  "refundId": "1234567890",
  "responseCode": "OK",
  "responseMessage": "potvrzeno",
  "reversal": false,
  "sequenceNumber": "001001614",
  "tipAmount": 100,
  "terminalIdAcquirer": "M1TNEXGO01",
  "terminalIdIssuer": "M1TNEXGO01",
  "transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
  "transactionType": "SALE_ONLINE",
  "tokens": ["fkdfkjbejrhlib"] //default - empty list
}
]
```

## 4.11. CARD VERIFY

Authorize a transaction with zero amount to verify the card and return the token.

### 4.11.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.

**ECR v1:** URI = "paya://card\_verify"

**ECR v2:** URI = "paya://card\_verify\_v2"

**ECR v3:** URI = "paya://card\_verify\_v3"

**ECR v4:** URI = "app://switchiopay/api/main/v4/card\_verify"

**ECR v5:** URI = "app://switchiopay/api/main/v5/card\_verify"

**ECR v6:** URI = "app://switchiopay/api/main/v6/card\_verify"

**ECR v7:** URI = "app://switchiopay/api/main/v7/card\_verify"

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://card_verify?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/card_verify?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/card\_verify
- parameter key: data
- parameter value: request json

### 4.11.2. Result

#### Result structure:

PARAMETER	TYPE	DESCRIPTION
<b>aid</b>	string	OPTIONAL Application identifier.
<b>appLabel</b>	string	OPTIONAL Label of application.
<b>authCode</b>	string	OPTIONAL Code of the authorization.
<b>cardHolderMessage</b>	string	OPTIONAL

		Message for cardholder.
<b>cardInputType</b>	string	OPTIONAL Type of the input. Enum is in the <a href="#">CardInputType</a>
<b>brand</b>	string	OPTIONAL Card brand.
<b>cvmTypeList</b>	string array	OPTIONAL Type of the transaction. <a href="#">CvmType</a>
<b>dateTimeServer</b>	string	OPTIONAL Date and time on the server when the result was made.
<b>dateTimeTerminal</b>	string	OPTIONAL Date and time on the terminal when the result was made.
<b>expiration</b>	string	OPTIONAL Expiration date of the card.
<b>merchantIdAcquirer</b>	string	OPTIONAL ID of the merchant's acquirer.
<b>merchantIdIssuer</b>	string	OPTIONAL ID of the merchant's issuer.
<b>merchantMessage</b>	string	OPTIONAL Message for merchant.
<b>pan</b>	string	OPTIONAL Primary account number.
<b>panSequenceNumber</b>	string	OPTIONAL Sequence number of the primary account number.
<b>responseCode</b>	string	OPTIONAL Code of the response. <a href="#">TerminalErrorCodes</a>
<b>responseMessage</b>	string	OPTIONAL Message of the response.
<b>sequenceNumber</b>	string	OPTIONAL Number of the sequence.
<b>terminalIdAcquirer</b>	string	OPTIONAL ID of the acquirer's terminal.
<b>terminalIdIssuer</b>	string	OPTIONAL ID of the issuer's terminal.
<b>transactionId</b>	string	ID of the transaction.
<b>transactionType</b>	string	Transaction type. <a href="#">TransactionTypes</a>
<b>cardToken</b>	string	OPTIONAL Encrypted combination of data.
<b>par</b>	string	OPTIONAL Card PAR number
<b>transactionReference</b>	string	OPTIONAL Transaction reference number

**Result example for v5:**

```
{
  "transactionId": "transactionId",
  "transactionType": "CARD_VERIFY",
  "aid": "A1000000000003",
  "appLabel": "MASTERCARD",
  "authCode": "auth",
  "brand": "Mastercard",
  "dateTimeTerminal": "May 13, 2023 02:28:56 PM",
  "dateTimeServer": "May 13, 2023 02:28:56 PM",
  "expiration": "***",
  "responseCode": "OK",
  "responseMessage": "OK",
  "terminalIdAcquirer": "MONETPLUS_ACQ",
  "terminalIdIssuer": "MONETPLUS_ISS",
  "merchantIdAcquirer": "MONETPLUS_ACQ",
  "merchantIdIssuer": "MONETPLUS_ISS",
  "pan": "111122223334444",
  "panSequenceNumber": "01",
  "merchantMessage": "Have a nice day",
  "sequenceNumber": "00100200",
  "cvmTypeList": [
    "NO"
  ],
  "cardInputType": "CLESS",
  "cardToken": "cardtoken",
  "par": "par",
  "transactionReference": "transactionReference"
}
```

## 4.12. SETTINGS

This request opens Settings in the Switchio Pay application.

### 4.12.1. Request

**ECR v1: URI = "paya://settings"**

**ECR v2: URI = "paya://settings\_v2"**

**ECR v3: URI = "paya://settings\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/settings"**

**ECR v5: URI = "app://switchiopay/api/main/v5/settings"**

**ECR v6: URI = "app://switchiopay/api/main/v6/settings"**

**ECR v7: URI = "app://switchiopay/api/main/v7/settings"**

#### Example request for ECR v1:

```
paya://settings
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/settings
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/settings
- parameter key: data
- parameter value: request json

### 4.12.2. Result

As a result of this request, Switchio Pay application opens. When Switchio Pay is closed, the 3rd party cash register application opens automatically.

## 4.13. OPEN PREAUTHORIZATIONS

Gets a list of all open pre-authorized transactions.

### 4.13.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v1: URI = "paya://preauth\_transactions"**

**ECR v2: URI = "paya://preauth\_transactions\_v2"**

**ECR v3: URI = "paya://preauth\_transactions\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/preauth\_transactions"**

**ECR v5: URI = "app://switchiopay/api/main/v5/preauth\_transactions"**

**ECR v6: URI = "app://switchiopay/api/main/v6/preauth\_transactions"**

**ECR v7: URI = "app://switchiopay/api/main/v7/preauth\_transactions"**

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://preauth_transactions
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/preauth_transactions
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/preauth\_transactions
- parameter value: request json

### 4.13.2. Result

Result is a list of [result structure](#).

#### Result example for v5:

```
[
{
  "transactionId": "37a0382e-cc83-44c1-b94e-732ad74380a5",
  "transactionType": "PREAUTHORIZATION",
  "aid": "A0000000043010",
  "amount": 14900,
  "appLabel": "MASTERCARD",
  "authCode": "123456",
  "reversal": false,
  "brand": "MASTERCARD",
  "currency": 203,
  "dateTimeTerminal": "Oct 09, 2023 02:43:06 PM",
  "dateTimeServer": "Oct 09, 2023 02:43:06 PM",
  "emvResponseData": {
    "smartCardScheme": 0,

```



```
    "authorisationResponseCode": "3030",
    "issuerAuthenticationData": "FFFF"
  },
  "expiration": "****",
  "invoiceNumber": "",
  "responseCode": "OK",
  "responseMessage": "OK",
  "terminalIdAcquirer": "TDEM01",
  "terminalIdIssuer": "terminalIdIssuer",
  "merchantIdAcquirer": "M1TDEM01",
  "merchantIdIssuer": "merchantIdIssuer",
  "pan": "541333*****9130",
  "tokens": [
    "FFFF"
  ],
  "panSequenceNumber": "01",
  "aosa": 14901,
  "cardHolderMessage": "Authorized",
  "merchantMessage": "authorized",
  "availableBalance": "14902",
  "sequenceNumber": "001001001",
  "cvmTypeList": [
    "PIN_ONLINE"
  ],
  "cardInputType": "CLESS",
  "refundId": "refundId",
  "par": "1234567890",
  "cardToken": "token"
}
]
```

## 4.14. INITIALIZE TERMINAL

Initialize the terminal before the first start or force initialization to wipe out the app storage and download a new configuration.

### 4.14.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>isForceInit</b>	boolean	OPTIONAL true to initialize the terminal even if it has been already initialized before, false to initialize only if necessary, default is false

Bundle result key = "data"

#### Example request for ECR v6:

```
app://switchiopay/api/config/v6/init?data={"isForceInit":true}
```

Additional info for building uri with Uri.Builder(), available for ECR v6:

scheme: app  
authority: switchiopay  
api: config  
path: /api/config/v6  
endpoint: init

### 4.14.2. Result

PARAMETER	TYPE	DESCRIPTION
<b>Result</b>	string	Overall initialization result Possible values: OK - Initialization successful ERROR - Initialization failed
<b>ttkStatus</b>	string	Possible values: OK - Master key OK NOT_FOUND - The master key has not been found. The master key needs to be injected into the terminal
<b>configurationStatus</b>	string	Possible values: OK - Configuration OK ALREADY_INITIALIZED - Configuration has been already initialized, skipping initialization UNKNOWN - Configuration status is not known because of other error, for example missing master key CREDENTIALS_ERROR - Configuration has not been downloaded because of invalid TMS credentials DOWNLOAD_ERROR - Configuration downloading failed FAILED - Downloaded configuration is corrupted
<b>configurationDownload Message</b>	string	Result message of the configuration download.

<b>configurationInit</b>		
<b>Message</b>	string	Result message of the configuration Init status.
<b>tid</b>	string	Terminal ID
<b>handshakeStatus</b>	string	Possible values: OK - Authorization server handshake successful FAILED - Authorization server handshake failed NOT_PERFORMED - Handshake has been skipped because of another error or the terminal is in the transport mode
<b>handshakeMessage</b>	string	Result message of the handshake status.

#### Definition of Result structure for v6:

```
{
  "result":<Result>,
  "appInitInfo":{
    "ttkStatus":<TtkStatus>,
    "configurationStatus":<ConfigurationStatus>,
    "configurationDownloadMessage":<string>,
    "configurationInitMessage":<string>,
    "tid":<string>,
    "handshakeStatus":<HandshakeStatus>,
    "handshakeMessage":<string>
  }
}
```

#### Result example for v6:

```
{
  "result":"OK",
  "appInitInfo":{
    "ttkStatus":"OK",
    "configurationStatus":"OK",
    "configurationDownloadMessage":"OK",
    "configurationInitMessage":"OK",
    "tid":"TPAX0001",
    "handshakeStatus":"OK",
    "handshakeMessage":"OK"
  }
}
```

## 5. RETAIL ENDPOINTS

### 5.1. PAYMENT

Perform a payment transaction with specified amount, currency, invoice number and optional tip.

#### 5.1.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>amount</b>	int	Amount is the amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>tipAmount</b>	int	OPTIONAL Amount of the tip.
<b>currencyCode</b>	int	Code of the currency. ISO 4217 currency code
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v1: URI = "paya://payment"**

**ECR v2: URI = "paya://payment\_v2"**

**ECR v3: URI = "paya://payment\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/payment"**

**ECR v5: URI = "app://switchiopay/api/main/v5/payment"**

**ECR v6: URI = "app://switchiopay/api/main/v6/payment"**

**ECR v7: URI = "app://switchiopay/api/main/v7/payment"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://payment?data={"transactionId": "88bb2366-716d-11ed-a1eb-0242ac120002", "amount": 1200, "tipAmount": 300, "currencyCode": 978, "invoiceNumber": "123456"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/payment?data={"transactionId": "88bb2366-716d-11ed-a1eb-0242ac120002", "amount": 1200, "tipAmount": 300, "currencyCode": 978, "invoiceNumber": "123456"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/payment



- parameter key: data
- parameter value: request json

### 5.1.2. Result

See [result structure](#) and [result examples](#).

## 5.2. REFUND

Perform a refund transaction with specified amount, currency and invoice number.

### 5.2.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>amount</b>	int	Amount is the amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>currencyCode</b>	int	Code of the currency. ISO 4217 currency code
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v1: URI = "paya://refund"**

**ECR v2: URI = "paya://refund\_v2"**

**ECR v3: URI = "paya://refund\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/refund"**

**ECR v5: URI = "app://switchiopay/api/main/v5/refund"**

**ECR v6: URI = "app://switchiopay/api/main/v6/refund"**

**ECR v7: URI = "app://switchiopay/api/main/v7/refund"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://refund?data={"transactionId": "88bb2366-716d-11ed-a1eb-0242ac120002", "amount": 1200, "tipAmount": 300, "currencyCode": 978, "invoiceNumber": "123456"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/refund?data={"transactionId": "88bb2366-716d-11ed-a1eb-0242ac120002", "amount": 1200, "tipAmount": 300, "currencyCode": 978, "invoiceNumber": "123456"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/refund
- parameter key: data
- parameter value: request json

### 5.2.2. Result

See [result structure](#) and [result examples](#).



### 5.3.2. Result

See [result structure](#) and [result examples](#).

## 5.4. INCREMENT PREAUTHORIZATION

Increment the pre-authorization transaction amount.

### 5.4.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>originalTransactionId</b>	string	ID of the original transaction that is being incremented.
<b>incrementAmount</b>	int	Increment amount is the amount that the transaction is being incremented by.
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v1: URI = "paya://preauth\_increment"**

**ECR v2: URI = "paya://preauth\_increment\_v2"**

**ECR v3: URI = "paya://preauth\_increment\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/preauth\_increment"**

**ECR v5: URI = "app://switchiopay/api/main/v5/preauth\_increment"**

**ECR v6: URI = "app://switchiopay/api/main/v6/preauth\_increment"**

**ECR v7: URI = "app://switchiopay/api/main/v7/preauth\_increment"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://preauth_increment?data={"transactionId": "89555389-4f67-4cb0-ae8-544ea11dbd1c",
"originalTransactionId": "88bb2366-716d-11ed-a1eb-0242ac120002", "incrementAmount": 1200,
"invoiceNumber": "123456"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/preauth_increment?data={"transactionId":
"89555389-4f67-4cb0-ae8-544ea11dbd1c", "originalTransactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002", "incrementAmount": 1200, "invoiceNumber": "123456"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/preauth\_increment
- parameter key: data
- parameter value: request json

## 5.4.2. Result

See [result structure](#) and [result examples](#).

## 5.5. DECREMENT PREAUTHORIZATION

Decrements the preauthorized transaction amount.

### 5.5.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>originalTransactionId</b>	string	ID of the original transaction that is being incremented.
<b>decrementAmount</b>	int	Decrement amount is the amount that the transaction is being decremented by.
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v5: URI = "app://switchiopay/api/main/v5/preauth\_decrement"**

**ECR v6: URI = "app://switchiopay/api/main/v6/preauth\_decrement"**

**ECR v7: URI = "app://switchiopay/api/main/v7/preauth\_decrement"**

Bundle result key = "transaction\_result"

#### Example request for ECR v5:

```
app://switchiopay/api/main/v5/preauth_decrement?data={"transactionId":
"89555389-4f67-4cb0-ae8-544ea11dbd1c", "originalTransactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002", "decrementAmount": 1200, "invoiceNumber": "123456"}
```

Additional info for building uri with Uri.Builder(), available for ECR v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v5/
- endpoint: preauth\_decrement
- parameter key: data
- parameter value: request json

## 5.5.2. Result

See [result structure](#) and [result examples](#).

## 5.6. COMPLETE PREAUTHORIZATION

To complete a pre authorization you need to provide transactionId of the corresponding successful pre-auth transaction. You also need to specify the final transaction amount. If any of the provided values are invalid or the transaction with the specified transactionId is not found in the Switchio Pay application database, the error result will be returned.

### 5.6.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>originalTransactionId</b>	string	ID of the original transaction that is being incremented.
<b>finalAmount</b>	int	Final amount is the sum of the incremental transaction and the original transaction.
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v1: URI = "paya://preauth\_complete"**

**ECR v2: URI = "paya://preauth\_complete\_v2"**

**ECR v3: URI = "paya://preauth\_complete\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/preauth\_complete"**

**ECR v5: URI = "app://switchiopay/api/main/v5/preauth\_complete"**

**ECR v6: URI = "app://switchiopay/api/main/v6/preauth\_complete"**

**ECR v7: URI = "app://switchiopay/api/main/v7/preauth\_complete"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://preauth_complete?data={"transactionId": "89555389-4f67-4cb0-ae8-544ea11dbd1c",
"originalTransactionId": "88bb2366-716d-11ed-a1eb-0242ac120002", "finalAmount": 1200,
"invoiceNumber": "123456"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/preauth_complete?data={"transactionId":
"89555389-4f67-4cb0-ae8-544ea11dbd1c", "originalTransactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002", "finalAmount": 1200, "invoiceNumber": "123456"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/preauth\_complete
- parameter key: data
- parameter value: request json

## 5.6.2. Result

See [result structure](#) and [result examples](#).

## 5.7. CANCEL PREAUTHORIZATION

In order to cancel a preauth transaction, transactionId of the corresponding successful preauth transaction is needed.

### 5.7.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>originalTransactionId</b>	string	ID of the original transaction that is being incremented.
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v1:** URI = "paya://preauth\_cancel"

**ECR v2:** URI = "paya://preauth\_cancel\_v2"

**ECR v3:** URI = "paya://preauth\_cancel\_v3"

**ECR v4:** URI = "app://switchiopay/api/main/v4/preauth\_cancel"

**ECR v5:** URI = "app://switchiopay/api/main/v5/preauth\_cancel"

**ECR v6:** URI = "app://switchiopay/api/main/v6/preauth\_cancel"

**ECR v7:** URI = "app://switchiopay/api/main/v7/preauth\_cancel"

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://preauth_cancel?data={"transactionId": "89555389-4f67-4cb0-ae8-544ea11dbd1c",
"originalTransactionId": "88bb2366-716d-11ed-a1eb-0242ac120002"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/preauth_cancel?data={"transactionId":
"89555389-4f67-4cb0-ae8-544ea11dbd1c", "originalTransactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/preauth\_cancel
- parameter key: data
- parameter value: request json

### 5.7.2. Result

See [result structure](#) and [result examples](#).

## 5.8. PREAUTHORIZATION – EXTERNAL COMPLETE

Complete a preauth transaction which was opened on a different terminal. Authorization data is available on the original transaction receipt.

### 5.8.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>originalTransactionData</b>	object	
<b>terminalId</b>	string	ID of the terminal.
<b>authCode</b>	string	Code of the authorization.
<b>seqNumber</b>	string	Number of the sequence.
<b>finalAmount</b>	int	Final amount is the sum of the incremental transaction and the original transaction.
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v3: URI = "paya://preauth\_external\_complete\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/preauth\_external\_complete"**

**ECR v5: URI = "app://switchiopay/api/main/v5/preauth\_external\_complete"**

**ECR v6: URI = "app://switchiopay/api/main/v6/preauth\_external\_complete"**

**ECR v7: URI = "app://switchiopay/api/main/v7/preauth\_external\_complete"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v3:

```
paya://preauth_external_complete_v3?data={"transactionId":"3e761c9c-7ab1-4aea-9bc0-fc29546f9e77","originalTransactionData":{"terminalId":"TERMINALID","authCode":"987654","seqNumber":"123456"},"finalAmount":1200,"invoiceNumber":"123456789"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/preauth_external_complete?data={"transactionId":"3e761c9c-7ab1-4aea-9bc0-fc29546f9e77","originalTransactionData":{"terminalId":"TERMINALID","authCode":"987654","seqNumber":"123456"},"finalAmount":1200,"invoiceNumber":"123456789"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/preauth\_external\_complete
- parameter key: data



- parameter value: request json

### 5.8.2. Result

See [result structure](#) and result example for [v4](#) and [v5](#).

## 5.9. PREAUTHORIZATION – EXTERNAL INCREMENT

Perform an increment on a preauth transaction which was opened on a different terminal. Authorization data is available on the original transaction receipt.

Available in Switchio Pay version 4.7.0 and higher.

### 5.9.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>originalTransactionData</b>	object	
<b>terminalId</b>	string	ID of the terminal.
<b>authCode</b>	string	Code of the authorization.
<b>seqNumber</b>	string	Number of the sequence.
<b>incrementAmount</b>	int	Increment amount is the amount that the transaction is being incremented by.
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v7: URI = "app://switchiopay/api/main/v7/preauth\_external\_increment"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v7:

```
app://switchiopay/api/main/v7/preauth_external_increment?data={"transactionId":"3e761c9c-7ab1-4aea-9bc0-fc29546f9e77","originalTransactionData":{"terminalId":"TERMINALID","authCode":"987654","seqNumber":"123456"},"incrementAmount":1200,"invoiceNumber":"123456789"}
```

### 5.9.2. Result

See [result structure](#) and [result example for v7](#).

## 5.10. PREAUTHORIZATION – EXTERNAL DECREMENT

Perform an decrement on a preauth transaction which was opened on a different terminal. Authorization data is available on the original transaction receipt.

Available in Switchio Pay version 4.7.0 and higher.

### 5.10.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>originalTransactionData</b>	object	
<b>terminalId</b>	string	ID of the terminal.
<b>authCode</b>	string	Code of the authorization.
<b>seqNumber</b>	string	Number of the sequence.
<b>decrementAmount</b>	int	Decrement amount is the amount that the transaction is being decremented by.
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v7: URI = "app://switchiopay/api/main/v7/preauth\_external\_decrement"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v7:

```
app://switchiopay/api/main/v7/preauth_external_decrement?data={"transactionId":"3e761c9c-7ab1-4aea-9bc0-fc29546f9e77","originalTransactionData":{"terminalId":"TERMINALID","authCode":"987654","seqNumber":"123456"},"decrementAmount":1200,"invoiceNumber":"123456789"}
```

### 5.10.2. Result

See [result structure](#) and [result example for v7](#).

## 5.11. PREAUTHORIZATION – EXTERNAL CANCEL

Cancel a preauth transaction created on a different POS.

### 5.11.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.
<b>originalTransactionData</b>		
<b>terminalId</b>	string	ID of the terminal
<b>authCode</b>	string	Code of the authorization.
<b>seqNumber</b>	string	Number of the sequence.

**ECR v3: URI = "paya://preauth\_external\_cancel\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/preauth\_external\_cancel"**

**ECR v5: URI = "app://switchiopay/api/main/v5/preauth\_external\_cancel"**

**ECR v6: URI = "app://switchiopay/api/main/v6/preauth\_external\_cancel"**

**ECR v7: URI = "app://switchiopay/api/main/v7/preauth\_external\_cancel"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v3:

```
paya://preauth_external_cancel_v3?data={"transactionId":"b4f945cc-eee9-43ac-bd59-3f0105cf0d19", "originalTransactionData":{"terminalId":"TERMINALID", "authCode":"987654", "seqNumber":"123456"}}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/preauth_external_cancel?data={"transactionId":"b4f945cc-eee9-43ac-bd59-3f0105cf0d19", "originalTransactionData":{"terminalId":"TERMINALID", "authCode":"987654", "seqNumber":"123456"}}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/preauth\_external\_cancel
- parameter key: data
- parameter value: request json

### 5.11.2. Result

See [result structure](#) and [result examples](#).

## 5.12. GET TRANSACTION

Request gets a transaction from the terminal based on the transactionId.

### 5.12.1. Request

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction.

**ECR v1:** URI = "paya://get\_transaction"

**ECR v2:** URI = "paya://get\_transaction\_v2"

**ECR v3:** URI = "paya://get\_transaction\_v3"

**ECR v4:** URI = "app://switchiopay/api/light/v4/get\_transaction"

**ECR v5:** URI = "app://switchiopay/api/light/v5/get\_transaction"

**ECR v6:** URI = "app://switchiopay/api/light/v6/get\_transaction"

**ECR v7:** URI = "app://switchiopay/api/light/v7/get\_transaction"

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://get_transaction?data={"transactionId": "89555389-4f67-4cb0-ae8-544ea11dbd1c" }
```

#### Example request for ECR v4:

```
app://switchiopay/api/light/v4/get_transaction?data={"transactionId": "89555389-4f67-4cb0-ae8-544ea11dbd1c" }
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v4/get\_transaction
- parameter key: data
- parameter value: request json

### 5.12.2. Result

See [result structure](#) and [result examples](#).

## 5.13. REVERSAL

This request performs transaction reversal. There are two options: (1) reverse transaction based on originalTransactionId (in this case, it is necessary to enter request data specified below); and (2) reverse the last transaction (this option does not require any request data).

### 5.13.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the reverse transaction.
<b>originalTransactionId</b>	string	ID of the transaction that is being reversed.
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v1: URI = "paya://reversal"**

**ECR v2: URI = "paya://reversal\_v2"**

**ECR v3: URI = "paya://reversal\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/reversal"**

**ECR v5: URI = "app://switchiopay/api/main/v5/reversal"**

**ECR v6: URI = "app://switchiopay/api/main/v6/reversal"**

**ECR v7: URI = "app://switchiopay/api/main/v7/reversal"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://reversal?data={"transactionId":
"89555389-4f67-4cb0-ae8-544ea11dbd1c", "originalTransactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/reversal?data={"transactionId":
"89555389-4f67-4cb0-ae8-544ea11dbd1c", "originalTransactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/reversal
- parameter key: data
- parameter value: request json

### 5.13.2. Result

See [result structure](#) and [result examples](#).

## 5.14. MOTO

Start a MOTO transaction. The card related data is entered manually in the payment application.

### 5.14.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>amount</b>	int	The amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, <b>amount</b> = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>currencyCode</b>	int	ISO 4217 currency code.
<b>invoiceNumber</b>	string	OPTIONAL Number of the invoice.
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v1: URI = "paya://moto\_payment"**

**ECR v2: URI = "paya://moto\_payment\_v2"**

**ECR v3: URI = "paya://moto\_payment\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/moto\_payment"**

**ECR v5: URI = "app://switchiopay/api/main/v5/moto\_payment"**

**ECR v6: URI = "app://switchiopay/api/main/v6/moto\_payment"**

**ECR v7: URI = "app://switchiopay/api/main/v7/moto\_payment"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://moto_payment?data={"transactionId": "88bb2366-716d-11ed-a1eb-0242ac120002", "amount": 1200, "currencyCode": 978, "invoiceNumber": "123456"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/moto_payment?data={"transactionId": "88bb2366-716d-11ed-a1eb-0242ac120002", "amount": 1200, "currencyCode": 978, "invoiceNumber": "123456"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/moto\_payment
- parameter key: data
- parameter value: request json



## 5.14.2. Result

See [result structure](#) and [result examples](#).

## 5.15. MOTO PREAUTHORIZATION

Start a MOTO Preauthorization. The card related data is entered manually in the payment application.

Available in Switchio Pay version 4.7.0 and higher.

### 5.15.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>amount</b>	int	The amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, <b>amount</b> = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>currencyCode</b>	int	ISO 4217 currency code.
<b>invoiceNumber</b>	string	OPTIONAL Number of the invoice.
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v7: URI = "app://switchiopay/api/main/v7/moto\_preauth"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v7:

```
app://switchiopay/api/main/v7/moto_preauth?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002", "amount": 1200, "currencyCode": 978, "invoiceNumber":
"123456" }
```

### 5.15.2. Result

See [result structure](#) and [result example for v7](#).

## 5.16. TRANSACTION STATUS

Returns status of selected transaction represented by transactionId.

### 5.16.1. Request

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction.

**ECR v1:** URI = "paya://transaction\_status"

**ECR v2:** URI = "paya://transaction\_status\_v2"

**ECR v3:** URI = "paya://transaction\_status\_v3"

**ECR v4:** URI = "app://switchiopay/api/main/v4/transaction\_status"

**ECR v5:** URI = "app://switchiopay/api/main/v5/transaction\_status"

**ECR v6:** URI = "app://switchiopay/api/main/v6/transaction\_status"

**ECR v7:** URI = "app://switchiopay/api/main/v7/transaction\_status"

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://transaction_status?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/transaction_status?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/transaction\_status
- parameter key: data
- parameter value: request json

### 5.16.2. Result

See [result structure](#) and [result examples](#).

## 5.17. REFERRAL REFUND

Authorize refund with refundId as a reference to the original transaction. Refund transaction is declined if the original transaction is not found. RefundId has to be 18 digits long.

### 5.17.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>amount</b>	int	The amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, <b>amount</b> = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>currencyCode</b>	int	ISO 4217 currency code.
<b>refundId</b>	string	ID of the refund.
<b>invoiceNumber</b>	string	OPTIONAL Number of the invoice.
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled

**ECR v3: URI = "paya://referral\_refund\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/referral\_refund"**

**ECR v5: URI = "app://switchiopay/api/main/v5/referral\_refund"**

**ECR v6: URI = "app://switchiopay/api/main/v6/referral\_refund"**

**ECR v7: URI = "app://switchiopay/api/main/v7/referral\_refund"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v3:

```
paya://referral_refund_v3?data={"transactionId": "88bb2366-716d-11ed-a1eb-0242ac120002",
"amount": 1200, "currencyCode": 978, "refundId": "23456395834576345389", "invoiceNumber":
"123456"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/referral_refund?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002", "amount": 1200, "currencyCode": 978, "refundId":
"23456395834576345389", "invoiceNumber": "123456"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/referral\_refund
- parameter key: data
- parameter value: request json

## 5.17.2. Result

See [result structure](#) and [result examples](#).

## 5.18. CASHBACK

This method allows customers to withdraw cash from a merchant during a card transaction.

**NOTE:** Only domestic cards can be used for the CASHBACK operation. This means that the terminal country code needs to be the same as the payment card country code. If the card country code differs from the terminal country code, the result of the transaction returns DOMESTIC\_CARD\_ONLY\_ERROR - see [TerminalErrorCodes](#) for more detail.

### 5.18.1. Request

**ECR v1: URI = "paya://cashback"**

**ECR v2: URI = "paya://cashback\_v2"**

**ECR v3: URI = "paya://cashback\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/cashback"**

**ECR v5: URI = "app://switchiopay/api/main/v5/cashback"**

**ECR v6: URI = "app://switchiopay/api/main/v6/cashback"**

**ECR v7: URI = "app://switchiopay/api/main/v7/cashback"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://cashback?data={"transactionId":"fe94a152-50db-488a-a04d-78407fed195d",  
"saleAmount":14900,"cashbackAmount":5,"currencyCode":203,"daughterCompanyId":  
"M1TDEMOD01", "invoiceNumber":"Invoice number"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/cashback?data={"transactionId":"fe94a152-50db-4  
88a-a04d-78407fed195d", "saleAmount":14900,"cashbackAmount":5,"currencyCode":2  
03,"daughterCompanyId":"M1TDEMOD01", "invoiceNumber":"Invoice number"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/cashback
- parameter key: data
- parameter value: request json

### 5.18.2. Result

See [result structure](#) and [result examples](#).

## 5.19. QR PAYMENT

Initiates a QR payment transaction. Transaction is completed by QR code scanning - see [SCANNER](#).

### 5.19.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>amount</b>	int	Amount is the amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>currencyCode</b>	int	Code of the currency. ISO 4217 currency code
<b>invoiceNumber</b>	string	OPTIONAL Variable symbol, max length is 20 char
<b>daughterCompanyId</b>	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

**ECR v5: URI = "app://switchiopay/api/main/v5/qr\_payment"**

**ECR v6: URI = "app://switchiopay/api/main/v6/qr\_payment"**

Bundle result key = "transaction\_result"

#### Example request for ECR v5:

```
app://switchiopay/api/main/v5/qr_payment?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002", "amount": 1200, "currencyCode":
978, "invoiceNumber": "123456"}
```

### 5.19.2. Result

See [result structure](#) and [result examples](#).

## 5.20. READ CARD

Read card data and store card data in terminal database for Contactless and Magstripe transactions.

### 5.21. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>amount</b>	int	Amount is the amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>currencyCode</b>	int	Code of the currency. ISO 4217 currency code

**ECR v7: URI = "app://switchiopay/api/main/v7/read\_card"**

Bundle result key = "transaction\_result"

#### Example request for ECR v7:

```
app://switchiopay/api/main/v7/read_card?data={"transactionId":"92751f62-17e0-46c5-a3fb-cd49ce1bc190", "amount":10000, "currencyCode":203}
```

Additional info for building uri with Uri.Builder(), available for ECR v7:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v7
- endpoint: read\_card

### 5.22. Result

See [result structure](#) and [result examples](#).

### 5.23. READ CARD PAYMENT

Succeeding operation after read\_card. Authorize transaction using previously obtained card data.

Perform a payment transaction with specified amount, currency, products, service level. Use stored card data for CLESS and Magstripe.

### 5.24. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>amount</b>	int	Amount is the amount of the transaction in a supported currency in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>currencyCode</b>	int	Code of the currency. ISO 4217 currency code
<b>cardInputType</b>	string	Possible values: <ul style="list-style-type: none"> <li>• CLESS - contactless</li> <li>• MAGNETIC - magstripe</li> <li>• CONTACT - put inside terminal</li> </ul>
<b>serviceLevel</b>	string	Type of sale. Possible values: <ul style="list-style-type: none"> <li>• SELF_SERVE - POS is unattended</li> <li>• FULL_SERVE - POS is attended</li> </ul>
<b>productList</b>	string array	
<b>code</b>	string	Type of sold product.
<b>measureUnit</b>	string	Type of measurement. Example: <ul style="list-style-type: none"> <li>• LITRES</li> <li>• UNITS</li> <li>• NO_UNIT</li> </ul>
<b>quantity</b>	string	OPTIONAL Number of sold product unit.
<b>unitPrice</b>	string	OPTIONAL Price per unit of measure.
<b>amount</b>	string	Monetary value of purchased product. The decimal point is implied by the optional currency code. The default value is two fractional decimal digits.
<b>taxCode</b>	char	Type of VAT included in the amount. Amended to alphanumeric to provide more potencial codes.
<b>additionalProductCode</b>	string	OPTIONAL Code identifies the product. It is optional.

**ECR v7: URI = "app://switchiopay/api/main/v7/read\_card\_payment"**

Bundle result key = "transaction\_result"

**Example request for ECR v7:**

```
app://switchiopay/api/main/v7/read_card_payment?data={"transactionId":  
"88bb2366-716d-11ed-a1eb-0242ac120002","amount": 1200,"currencyCode": 978,"cardInputType":  
"CLESS","serviceLevel": "SELF_SERVE","productList": [{"code": "123","measureUnit":  
"LITRES","quantity": "2.1","unitPrice": "0.45","amount": "1212","taxCode": "C"}, {"code":  
"34","measureUnit": "UNITS","quantity": "3.2","unitPrice": "1","amount": "32","taxCode":  
"P","additionalProductCode": "443q"}]}
```

Additional info for building uri with `Uri.Builder()`, available for ECR v7:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v7/
- endpoint: read\_card\_payment
- parameter key: data
- parameter value: request json

**5.25. Result**

See [result structure](#) and [result examples](#).

## 6. TRANSPORT ENDPOINTS

### 6.1. TRANSPORT INIT

Initialize transport mode and denylist. Network connection is required.

#### 6.1.1. Request

This operation has no request data.

**ECR v1: URI = "paya://init\_transport"**

**ECR v2: URI = "paya://init\_transport\_v2"**

**ECR v3: URI = "paya://init\_transport\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/init\_transport"**

**ECR v5: URI = "app://switchiopay/api/main/v5/init\_transport"**

**ECR v6: URI = "app://switchiopay/api/main/v6/init\_transport"**

**ECR v7: URI = "app://switchiopay/api/main/v7/init\_transport"**

Bundle result key = "result"

#### Example request for ECR v1:

```
paya://init_transport
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/init_transport
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/init\_transport
- parameter value: request json

#### 6.1.2. Result

##### Result structure:

PARAMETER	TYPE	DESCRIPTION
resultCode	int	0 - Success 1 - Internal error 2 - Download error
message	string	Result message.

##### Result example:

```
{
  "resultCode": 0,
  "message": "Success"
}
```

## 6.2. TRANSPORT PAYMENT

Requesting transport payment data from Switchio Pay.

### 6.2.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>transportAmount</b>	int	OPTIONAL, available from v4.5.0 Transaction amount shown on the terminal/validator display when the card is tapped. The amount is sent in the minimum currency unit. For example, if the currency is Kč 10,00, amount = „1000“; EUR 1,50 = „150“ (without decimal point).
<b>transportCurrencyCode</b>	int	OPTIONAL, available from v4.5.0 Currency code visualized on the terminal/validator display when the card is tapped.
<b>processingMode</b>	string	OPTIONAL, available from v4.7.0; values: Aggregated / KFT. Just resend it to Transport server
<b>InvoiceNumber</b>	string	Variable symbol, max length is 20 char
<b>virtualTerminalId</b>	string	OPTIONAL, available from v4.7.0 daughterCompanyId, required only when virtualisation is enabled
<b>transportData</b>	string	base64 encoded string data from validator

**ECR v1: URI = "paya://payment\_transport"**

**ECR v2: URI = "paya://payment\_transport\_v2"**

**ECR v3: URI = "paya://payment\_transport\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/payment\_transport"**

**ECR v5: URI = "app://switchiopay/api/main/v5/payment\_transport"**

**ECR v6: URI = "app://switchiopay/api/main/v6/payment\_transport"**

**ECR v7: URI = "app://switchiopay/api/main/v7/payment\_transport"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v1:

```
paya://payment_transport?data={"transactionId":"3ffed32e-6243-430e-905b-f0c5552feeff",
"invoiceNumber":"1234", "transportData":"data"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/payment_transport?data={"transactionId":"3ffed32e-6243-430e-905b-f0c5552feeff", "transportAmount":10053, "transportCurrencyCode":203,
"invoiceNumber":"1234", "transportData":"data"}
```

#### Example request for ECR v7:

```
app://switchiopay/api/main/v7/payment_transport?data={"transactionId":"3ffed32e-6243-430e-905b-f0c5552feeff", "transportAmount":10053, "transportCurrencyCode":203,
"processingMode":"KFT", "invoiceNumber":"1234", "virtualTerminalId":"M1TDEMOD01",
"transportData":"data"}
```



Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/payment\_transport
- parameter key: data
- parameter value: request json

## 6.2.2. Result

### Result structure:

PARAMETER	TYPE	DESCRIPTION
<b>processingResultCode</b>	int	Operation result code
<b>message</b>	string	String describing cause of the failure
<b>fareTransactionRequest</b>	fareTransactionRequest	OPTIONAL fareTransactionRequest object
<b>device</b>	device	device object
<b>type</b>	string	EMV reader brand
<b>reader_id</b>	string	Card reader ID
<b>virtual_reader_id</b>	string	Virtual reader ID if virtualization is enabled and is in request
<b>reader_sn</b>	string	Card reader serial number
<b>deny_list_version</b>	string	Current version of the deny list
<b>transaction</b>	transaction	transaction object
<b>token</b>	string	Encrypted combination of data
<b>id</b>	string	Transaction ID
<b>dtm</b>	string	Date and time of the transaction
<b>result</b>	string	Token deny list result
<b>masked_pan</b>	string	Masked PAN of the card
<b>fare_data</b>	string	base64 encoded string data from validator
<b>emv_data</b>	emv_data	emv_data object
<b>tags</b>	array	
<b>encrypted</b>	encrypted	encrypted object
<b>encryption_method</b>	string	Method used for encryption
<b>encrypted_data</b>	string	Encrypted data

**Result example:**

```
{
  "processingResultCode": 0,
  "message": "ok",
  "fareTransactionRequest": {
    "device": {
      "type": "NEXGO",
      "reader_id": "THOSKBB999",
      "reader_sn": "N600W102315",
      "deny_list_version": "D002592310120002"
    },
    "transaction": {
      "token": "119D45E2D7E32E1AE676CF7B196F790B7F4C3FD69C026427C9C4A9AC3B006C1E50",
      "id": "5e9c5819-0ad3-494a-a61d-acf7b632d1ad",
      "dtm": "2023-10-12T12:28:53Z",
      "result": "NO_DENYLIST",
      "masked_pan": "541333*****4111"
    },
    "fare_data": "eyJCdXMiOjF9",
    "emv_data": {
      "tags": [
        {
          "key": "9F24",
          "value": "353030314142434445464748494A4B4C4D4E4F50515253545556575859"
        },
        {
          "key": "9F06",
          "value": "A0000000041010"
        },
        {
          "key": "5F24",
          "value": "2512"
        }
      ],
      "encrypted": {
        "encryption_method": "MNSP",
        "encrypted_data":
"00787F637B727209096D54AC173A2DE33A098C2001BB67DE20605FB79528E5AAFB12DA2FD535C08E40AA8CBF25
682226FA63228FD6A5F37A98BF26F5193405E748433336E09A132B07AC39D9911740A741C936CDF63201499BA30
E7A0F1098E2F9778D59B9949988259FB231CFE6DBD75B528FD0EA9A9E4EB8B76D2620E4F7AA4FDCEFC5C07188F5
30525559F2D62BCA08053A36B49DD22432131AF3275F4B30F6800744D94344BBDCB7ED8021E7FDC8FC655CE7AAD
E7D9F2F34226459DEA6B00FA3BC4D9E1ADAF851053BF94825257C11B79BC749EBD00A9674614C8392CEFE3FD910
8C91E7F87025147F979BAAB23B0B4D1D47A083D20A67BFFDC3D42B89648EED81B0625CAA90873063DD0EFC311F2
B03E1EAE447F796969C02C8C9A62BE5A8F53688F4F76FA3AC531BBB13EF5575F623969041870EF07E6F9164A4AC
440979CCDF1C1FEA76A434D344318A47AADFAB26B27FA358400FB4C7CE75D20359FF672BAF035C7645F87FE3519
FAD008914840E3465737AAD4DAAD1"
      }
    }
  }
}
```

### 6.3. TRANSPORT REFUND

Requesting transport refund data from Switchio Pay. Available from v4.7.0.

#### 6.3.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the transaction.
<b>transportAmount</b>	int	Amount of transaction
<b>transportCurrency Code</b>	int	Currency code
<b>InvoiceNumber</b>	string	Optional, Id of transaction that is being reversed.
<b>additionalText</b>	string	Optional, Option to use for customer email address
<b>virtualTerminalId</b>	string	Optional, daughterCompanyId, required only when virtualisation is enabled

**ECR v7: URI = ""app://switchiopay/api/main/v7/refund\_transport**

Parameter key = "data"

Bundle result key = "transaction\_result"

**Example request for ECR v7:**

```
app://switchiopay/api/main/v7/refund_transport?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002","amount": 1200,"currencyCode": 978,"invoiceNumber":
"123456", "additionalText": "customer@gmail.com", "virtualTerminalId":"M1TDEM01"}
```

Additional info for building uri with Uri.Builder(), available for ECR v7:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v7/refund\_transport
- parameter key: data
- parameter value: request json

### 6.3.2. Result

**Result structure:**

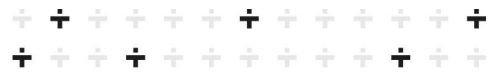
PARAMETER	TYPE	DESCRIPTION
<b>processingResultCode</b>	int	Operation result code
<b>message</b>	string	String describing cause of the failure
<b>fareTransactionRequest</b>	fareTransactionRequest	OPTIONAL fareTransactionRequest object
<b>device</b>	device	device object
<b>type</b>	string	Type of the device
<b>validator_id</b>	string	Logical id of the validator
<b>validator_sn</b>	string	Validator serial number
<b>reader_id</b>	string	EMV reader logical ID
<b>reader_sn</b>	string	EMV reader serial number
<b>virtual_reader_id</b>	string	ID of the virtual reader, optional
<b>deny_list_version</b>	string	Version of deny list in the device in the moment of transaction
<b>transaction</b>	transaction	transaction object
<b>token</b>	string	Encrypted combination of data
<b>masked_pan</b>	string	Masked PAN value in format 6+4 (optional)
<b>id</b>	string	Transaction ID on the device
<b>dtm</b>	string	Date and time of the transaction, preferred ISO 8601 format, time zone is local time zone of the device
<b>result</b>	string	Transaction result = OK, FAIL, DENYLIST etc...
<b>mode</b>	string	Processing mode, optional, defaults to MTT, possible values MTT/KFT
<b>detail_result</b>	string	Detail result = response code from transaction processing or some processing details
<b>fare_data</b>	string	BASE64 encoded string data from validator. Data holding information about the trip, zones, amount - project specific
<b>emv_data</b>	emv_data	List of clear EMV tags - project specific
<b>tags</b>	array	
<b>key</b>	string	
<b>value</b>	string	
<b>encrypted</b>	encrypted	List of encrypted EMV tags - project specific

<b>encryption_method</b>	string	Method used for encryption (DUKPT)
<b>encryption_seed</b>	string	Data needed for key derivation
<b>encrypted_data</b>	string	Encrypted sensitive EMV data

### Result example:

```
{
  "processingResultCode": 0,
  "message": "ok",
  "fareTransactionRequest": {
    "device": {
      "type": <string>,
      // type of the device
      "validator_id": <string>,
      // logical id of the validator
      "validator_sn": <string>,
      // validator serial number
      "reader_id": <string>,
      // emv reader logical id
      "reader_sn": <string>,
      // emv reader serial number
      "virtual_reader_id": <string>,
      // id of the virtual reader, optional
      "deny_list_version": <string>
      // version of deny list in the device in the moment of transaction
    },
    "transaction": {
      "token": <string>,
      // card token calculated by the device
      "masked_pan": <string>,
      // masked pan value in format 6+4 (optional)
      "id": <string>,
      // transaction id on the device
      "dtm": <string>,
      // date and time of the transaction, preferred ISO 8601 format,
      //time zone is local time zone of the device
      "result": <string>,
      // transaction result = OK, FAIL, DENYLIST etc...
      "mode": <string>,
      //processing mode, optional, defaults to MTT, possible values MTT/KFT
      "detail_result": <string>
      // detail result = response code from transaction processing
      //or some processing details
    },
    "fare_data": <string>,
    // data holding information about the trip, zones, amount - project specific
    // BASE64 encoded binary TLV structure

    //list of clear emv tags - project specific
    "emv_data": {
      "tags": [
        {
          "key": <string>,
          "value": <string>
        }
      ]
    }
  }
}
```



```
    ],  
    //list of encrypted emv tags - project specific  
    "encrypted":{  
        "encryption_method": <string>,  
        // DUKPT  
        "encryption_seed": <string>,  
        // data needed for key derivation  
        "encrypted_data": <string>  
        // encrypted sensitive emv data  
    }  
}  
}
```

## 6.4. TRANSPORT REVERSAL

This request performs transaction reversal. Reverse transaction based on originalTransactionId (it is necessary to enter request data specified below).

### 6.4.1. Request

PARAMETER	TYPE	DESCRIPTION
<b>transactionId</b>	string	ID of the reverse transaction.
<b>originalTransactionId</b>	string	ID of the transaction that is being reversed.
<b>additionalText</b>	string	Optional, Option to use for customer email address
<b>virtualTerminalId</b>	string	Optional, daughterCompanyId, required only when virtualisation is enabled

**ECR v7: URI = "app://switchiopay/api/main/v7/reversal\_transport"**

Parameter key = "data"

Bundle result key = "transaction\_result"

#### Example request for ECR v7:

```
app://switchiopay/api/main/v7/reversal_transport?data={"transactionId":
"89555389-4f67-4cb0-ae8-544ea11dbd1c", "originalTransactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002", "additionalText": "customer@gmail.com",
"virtualTerminalId": "M1TDEM01" }
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v7/reversal\_transport
- parameter key: data
- parameter value: request json

### 6.4.2. Result

#### Result structure:

PARAMETER	TYPE	DESCRIPTION
<b>resultCode</b>	int	Operation result code, enums: <ul style="list-style-type: none"> <li>• CODE_SUCCESS = 0</li> <li>• CODE_INVALID_REQUEST = 1</li> </ul>
<b>message</b>	string	String describing cause of the failure

#### Result example:

```
{
"resultCode": 0,
"message": "ok",
}
```

## 6.5. CUSTOMER TAP

This gets information about a tap based on the transaction ID.

### 6.5.1. Request

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction.

**ECR v1: URI = "paya://customer\_tap"**

**ECR v2: URI = "paya://customer\_tap\_v2"**

**ECR v3: URI = "paya://customer\_tap\_v3"**

**ECR v4: URI = "app://switchiopay/api/main/v4/customer\_tap"**

**ECR v5: URI = "app://switchiopay/api/main/v5/customer\_tap"**

**ECR v6: URI = "app://switchiopay/api/main/v6/customer\_tap"**

**ECR v7: URI = "app://switchiopay/api/main/v7/customer\_tap"**

Parameter key = "data"

Bundle result key = "data"

#### Example request for ECR v1:

```
paya://customer_tap?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

#### Example request for ECR v4:

```
app://switchiopay/api/main/v4/customer_tap?data={"transactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: main
- path: /api/main/v4/customer\_tap
- parameter key: data
- parameter value: request json

## 6.5.2. Result

### Result structure:

PARAMETER	TYPE	DESCRIPTION
<b>processingResultCode</b>	int	0 - Success 3 - Card read error 4 - Encryption error 5 - Timeout 8 - Internal error
<b>message</b>	string	Result message.
<b>response</b>	object	Response object with the result.
<b>result</b>	string	Possible values: <ul style="list-style-type: none"> <li>• OK</li> <li>• FAIL</li> </ul>
<b>resultMessage</b>	string	OPTIONAL String describing the cause of the failure.
<b>code</b>	string	Response code from terminal; see <a href="#">TerminalErrorCodes</a> table.

### Result example:

```
{
  "processingResultCode": 0,
  "message": "OK",
  "response": {
    "result": "OK",
    "resultMessage": "result message",
    "code": "code"}
}
```

## 6.6. UPLOAD TRANSPORT TRANSACTIONS

The Terminal periodically uploads transactions. This API allows the cash register to force upload all stored transactions in the terminal.

### 6.6.1. Request

This operation has no request data. Result of the upload operation is broadcasted.

**ECR v1: URI = "paya://upload\_transport\_transactions"**  
**ECR v2: URI = "paya://upload\_transport\_transactions\_v2"**  
**ECR v3: URI = "paya://upload\_transport\_transactions\_v3"**  
**ECR v4: URI = "app://switchiopay/api/light/v4/upload\_transport\_transactions"**  
**ECR v5: URI = "app://switchiopay/api/light/v5/upload\_transport\_transactions"**  
**ECR v6: URI = "app://switchiopay/api/light/v6/upload\_transport\_transactions"**  
**ECR v7: URI = "app://switchiopay/api/light/v7/upload\_transport\_transactions"**

Bundle result key = "result"

#### Example request for ECR v1:

```
paya://upload_transport_transactions
```

#### Example request for ECR v4:

```
app://switchiopay/api/light/v4/upload_transport_transactions
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v4/upload\_transport\_transactions
- parameter value: request json

### 6.6.2. Result

#### Result structure:

PARAMETER	TYPE	DESCRIPTION
<b>resultCode</b>	int	0 - Success 1 - Upload error 2 - Configuration error 4 - No transaction found
<b>message</b>	string	Result message.

#### Result broadcast action:

```
switchiopay.transport.upload
```

Broadcast action = "paya.tmsparam.status"

Broadcast bundle key = "tmsParamStatus"

#### Result example:

```
{
  "resultCode": 4
  "message": "no transactions found"
}
```

```
}
```

## 6.7. CUSTOMER TAP SERVICE START

This endpoint starts a service in the background that makes it possible to open a client's application that receives data with information about the taps made.

### 6.7.1. Request

**ECR v1:** URI = "paya://customer\_tap\_service\_start"

**ECR v2:** URI = "paya://customer\_tap\_service\_start\_v2"

**ECR v3:** URI = "paya://customer\_tap\_service\_start\_v3"

**ECR v4:** URI = "app://switchiopay/api/light/v4/customer\_tap\_service\_start"

**ECR v5:** URI = "app://switchiopay/api/light/v5/customer\_tap\_service\_start"

**ECR v6:** URI = "app://switchiopay/api/light/v6/customer\_tap\_service\_start"

**ECR v7:** URI = "app://switchiopay/api/light/v7/customer\_tap\_service\_start"

Bundle result key = "data"

#### Example request for ECR v1:

```
paya://customer_tap_service_start
```

#### Example request for ECR v4:

```
app://switchiopay/api/light/v4/customer_tap_service_start
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v4/customer\_tap\_service\_start
- parameter value: request json

### 6.7.2. Result

See [Result structure](#).

#### Result broadcast action:

```
switchio.pay.customertap
```

Broadcast action = "switchio.pay.customertap"

Broadcast bundle key = "data"

#### Result example:

```
{  
  "resultCode": 4,  
  "message": "no transactions found"  
}
```

## 6.8. CUSTOMER TAP SERVICE STOP

Stop the customer tap service and the client's application ceases to receive information about card taps.

### 6.8.1. Request

**ECR v1: URI = "paya://customer\_tap\_service\_stop"**

**ECR v2: URI = "paya://customer\_tap\_service\_stop\_v2"**

**ECR v3: URI = "paya://customer\_tap\_service\_stop\_v3"**

**ECR v4: URI = "app://switchiopay/api/light/v4/customer\_tap\_service\_stop"**

**ECR v5: URI = "app://switchiopay/api/light/v5/customer\_tap\_service\_stop"**

**ECR v6: URI = "app://switchiopay/api/light/v6/customer\_tap\_service\_stop"**

**ECR v7: URI = "app://switchiopay/api/light/v7/customer\_tap\_service\_stop"**

#### Example request for ECR v1:

```
paya://customer_tap_service_stop
```

#### Example request for ECR v4:

```
app://switchiopay/api/light/v4/customer_tap_service_stop
```

Additional info for building uri with `Uri.Builder()`, available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v4/customer\_tap\_service\_stop
- parameter value: request json

### 6.8.2. Result

As a result of this request, the customer tap service stops and the customer's application does not receive data about taps any more.

## 6.9. GET TAPS

Gets information about card taps that were not sent to the server.

### 6.9.1. Request

This operation has no request data. Result of the get taps operation is broadcasted. If there are no available taps, the result list will be empty.

**ECR v5: URI = "app://switchiopay/api/light/v5/get\_taps"**

**ECR v6: URI = "app://switchiopay/api/light/v6/get\_taps"**

**ECR v7: URI = "app://switchiopay/api/light/v7/get\_taps"**

Bundle result key = "get\_taps"

#### Example request for ECR v5:

```
app://switchiopay/api/light/v5/get_taps
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v5/get\_taps
- parameter value: request json

### 6.9.2. Result

#### Result structure:

PARAMETER	TYPE	DESCRIPTION
<b>token</b>	string	Card token
<b>id</b>	string	ID of the tap
<b>dttm</b>	string	Date and time of the tap
<b>result</b>	string	Possible values: <ul style="list-style-type: none"> <li>• OK</li> <li>• FAIL</li> <li>• EXPIRED</li> <li>• DENYLISTED</li> <li>• NOTIFY_DENYLIST</li> <li>• NO_DENYLIST</li> </ul>
<b>detail_result</b>	string	OPTIONAL
<b>masked_pan</b>	string	OPTIONAL Partially hidden PAN

#### Result example:

```
[
  {
    "token": "token12345",
    "id": "7d1c083d-19f3-4d7b-aa14-f615ed500a85",
    "dttm": "2023-09-05T08:49:55Z",
    "result": "OK",
    "detail_result": "result",
  }
]
```

```
"masked_pan": "1234****1234"  
}  
]
```

## 6.10. GET DENYLIST UPDATE

Method for getting the date of last denylist update. Date is in format "yyyy-MM-dd HH:mm:ss".

### 6.10.1. Request

This operation has no request data. Result of the get taps is broadcasted, if there are no available taps the result list will be empty.

**ECR v5: URI = "app://switchiopay/api/light/v5/get\_denylist\_last\_update"**

**ECR v6: URI = "app://switchiopay/api/light/v6/get\_denylist\_last\_update"**

**ECR v7: URI = "app://switchiopay/api/light/v7/get\_denylist\_last\_update"**

Bundle result key = "get\_denylist\_last\_update"

#### Example request for ECR v5:

```
app://switchiopay/api/light/v5/get_denylist_last_update
```

Additional info for building uri with Uri.Builder(), available for ECR v4, v5:

- scheme: app
- authority: switchiopay
- api: light
- path: /api/light/v5/get\_denylist\_last\_update
- parameter value: request json

### 6.10.2. Result

#### Result structure:

```
<string>
```

#### Result example:

```
"2023-09_06 10:01:02"
```

## 7. APPLICATION DESCRIPTION

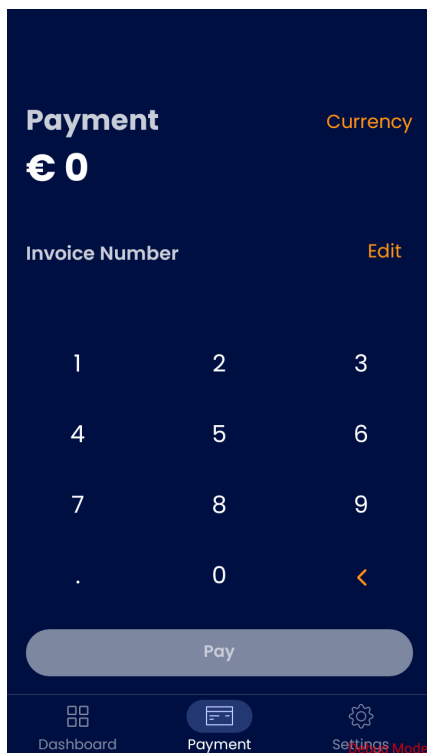
Certified app for Android terminal for EMV card payment. It supports two modes, standalone and external cash register mode.

The payment application primarily serves to complete operations triggered by third party cash registers using the ECR protocol.

The external cash register mode is blocked from invoking payment functions directly from the payment application. Instead, it uses the ECR protocol, and payment operations are triggered by a third-party cash register with the given parameters. After completion of the operation, the result or data related to the result of the transaction are returned to the cash register. When you start the application from the system menu of the application, the Dashboard is displayed, ie a menu with basic functions such as shutter and settings.

Standalone mode supports all terminal payment operations without the need for a third party cash register. The start screen in this mode is payment.

### 7.1. Payment



The operator can make a payment by entering the amount or variable symbol. When entering the variable symbol, the system keyboard is displayed, the numeric keyboard is displayed after clicking on the displayed amount. After ECR protocol supports running of the token generator in the background. Application has control of turning on and off the card detection and a token generation via the ECR protocol. Once the token generator is turned on, the token will be broadcasted from the payment application. The Client application is able to receive the token by implementing the BroadcastReceiver. After the response from the payment application, start the generator again to generate another token.

Make sure you stop the token generator as well as unregister the broadcast receiver after you are done to prevent excessive battery usage and memory leaks.

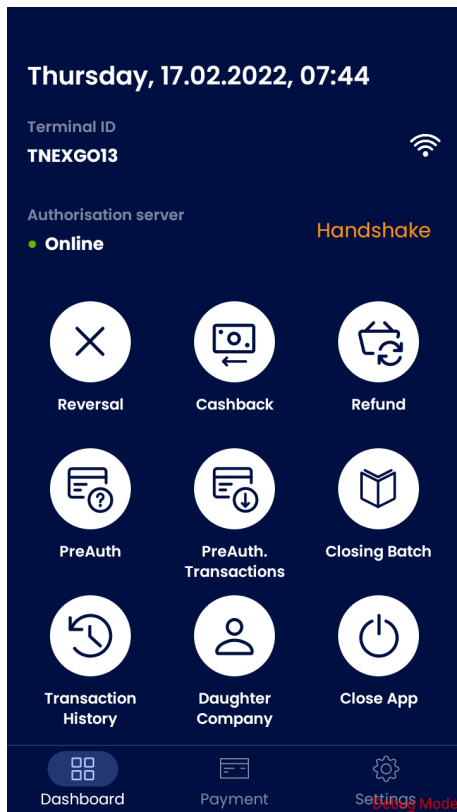
By filling in the necessary data and pressing the "Pay" button, you will be prompted to apply your credit card and the standard payment process will take place.

The button in the upper left corner gives the operator access to the dashboard, where he has settings, closing date and all other terminal functions.

## 7.2. Refund

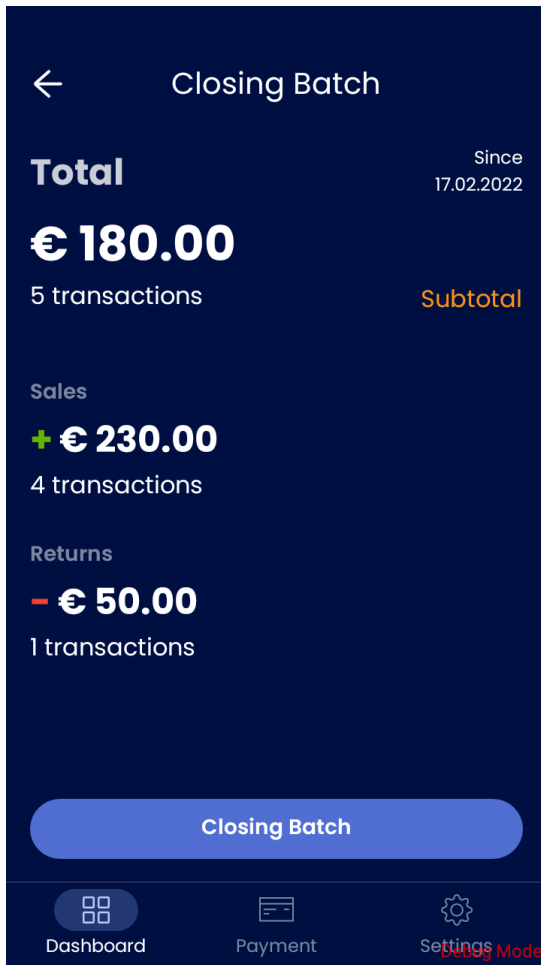
The operation flow of the refund transaction is basically the same as for standard payment.

## 7.3. Dashboard



The dashboard has the function of the main application menu. Displays basic terminal status, wifi signal, battery status, server status, date and time. It also contains links to settings, the closing date and other terminal functions, depending on the mode and configuration of the application. The dashboard is also where you can exit the app by pressing the button in the top right corner.

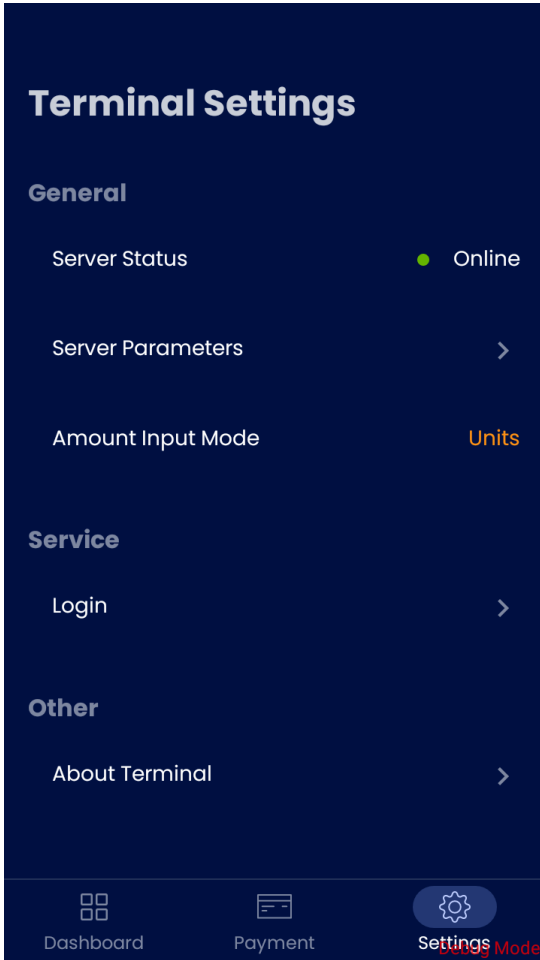
### 7.4. The closing batch



The closing batch function allows the operator to view the current subtotal, print a block with a subtotal, and also create the closing batch itself.

In the closing batch screen, each color represents a different type of transaction amounts / counts. Green stands for payments, red for refunds and blue represents total amounts and counts.

## 7.5. Settings



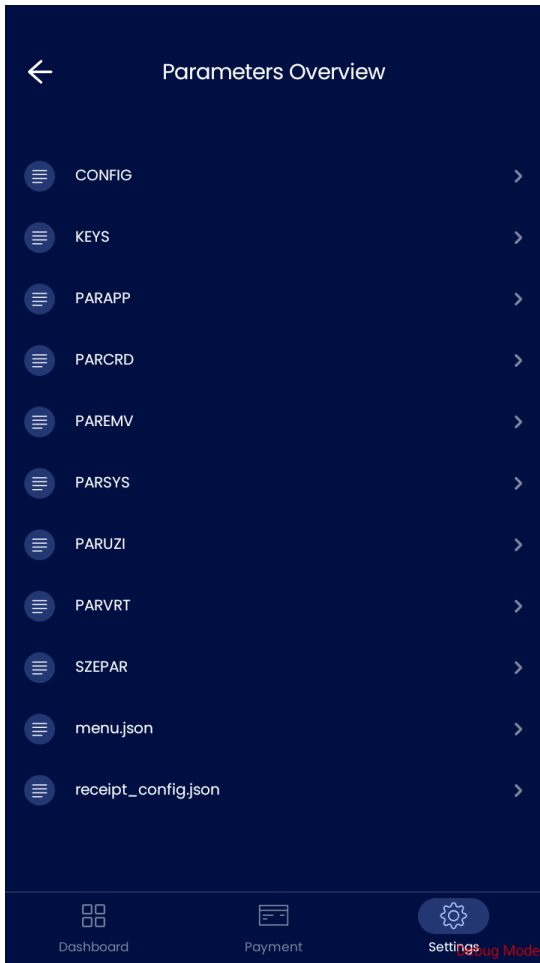
The terminal settings include tools for downloading parameters from the server, editing them, a service menu for testing all terminal functions without the need for an intermediate application protocol, loading keys to the terminal, and more.

The Server Parameters menu contains settings for downloading parameters. They can be edited by a service technician in the Service menu. The default service PIN is set to 99999999.

After logging into the service technician, all terminal functions are made available in the Dashboard menu.

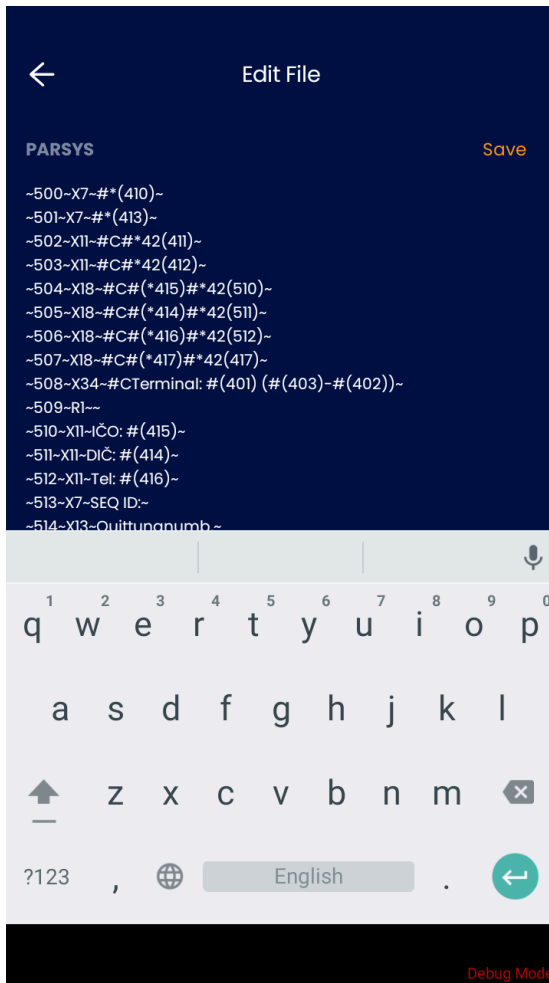
Tapping Server Status runs the handshake process resulting in one of Online, Offline or Unknown status.

## 7.6. Service Settings

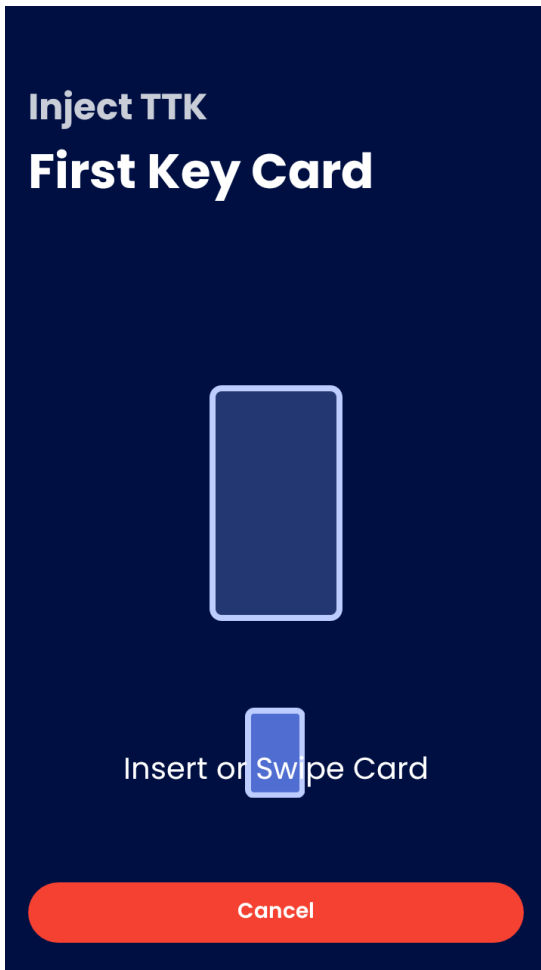


Service settings offer tools to setup or change configuration files by authorized technicians.

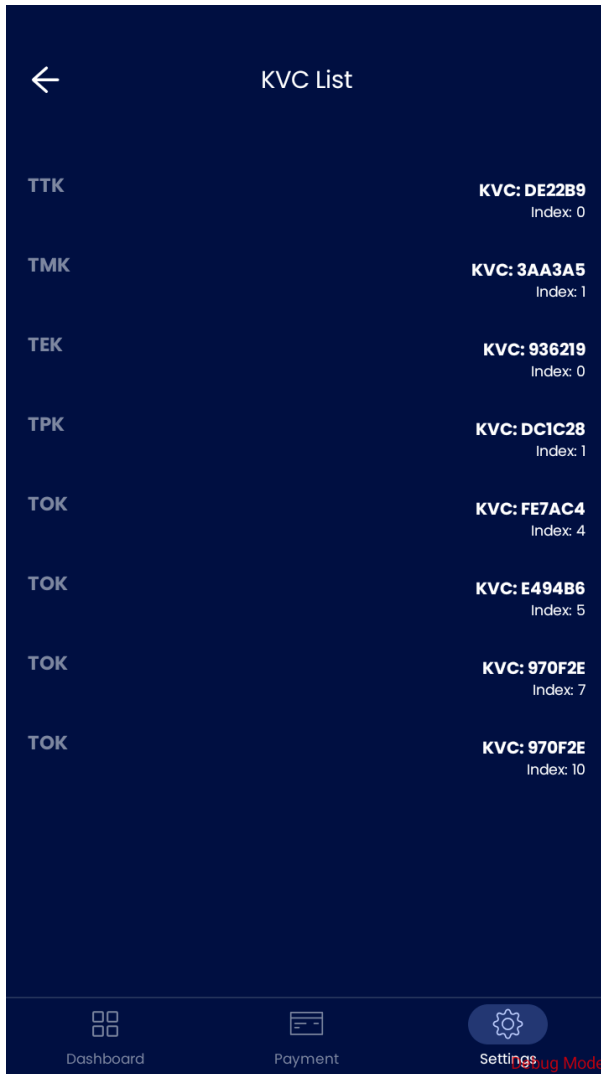
Parameters menu allows the technician to edit terminal parameters locally.



### 7.6.1. Key injection



Service menu contains the wizard to inject terminal keys using two TTK cards. The wizard takes the technician through the complete key injection setup.



Service menu also allows the technician to view the KCVs of the injected keys as well as their positions.

# ATTACHMENT – ENUMS

## TerminalErrorCodes

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	✗
9	NO_PAPER	No paper available	✗
10	UNKNOWN_MERCHANT	Unknown merchant	✗
11	UNKNOWN	Transaction has not been completed	✗
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.	✓

<b>1008</b>	ApprovedNegativeCardFileHitOK	Approved negative card file hit OK.	✓
<b>1009</b>	ApprovedCommercialStatus	Approved commercial status.	✓
<b>1010</b>	ApprovedForPartialAmount	Approved for partial amount.	✓
<b>1050</b>	DeclinedGeneral	General decline.	✗
<b>1050</b>	DeclinedProductGroup	Product group not allowed.	✗
<b>1050</b>	DeclinedCardBlocked	The card is blocked.	✗
<b>1050</b>	DeclinedCardBlockedByRange	The card is blocked by range.	✗
<b>1050</b>	DeclinedAccountNotAllowedOnStation	The use of an account card is not allowed on the given gas station.	✗
<b>1050</b>	DeclinedProductGroupForStation	Product group not allowed for the given gas station.	✗
<b>1050</b>	DeclinedCardDailyTxCountLimitExceeded	The card has exceeded the daily maximum number of transactions.	✗
<b>1050</b>	DeclinedCardWeeklyTxCountLimitExceeded	The card has exceeded the weekly maximum number of transactions.	✗
<b>1050</b>	DeclinedCardMonthlyTxCountLimitExceeded	The card has exceeded the monthly maximum number of transactions.	✗
<b>1050</b>	DeclinedCardDailyFuelLimitExceeded	The card has exceeded the daily fuel limit.	✗
<b>1050</b>	DeclinedCardWeeklyFuelLimitExceeded	The card has exceeded the weekly fuel limit.	✗
<b>1050</b>	DeclinedCardMonthlyFuelLimitExceeded	The card has exceeded the monthly fuel limit.	✗
<b>1050</b>	DeclinedCardDailyMoneyLimitExceeded	The card has exceeded the daily money limit.	✗
<b>1050</b>	DeclinedCardWeeklyMoneyLimitExceeded	The card has exceeded the weekly money limit.	✗
<b>1050</b>	DeclinedCardMonthlyMoneyLimitExceeded	The card has exceeded its monthly cash limit.	✗
<b>1050</b>	DeclinedCardTxPriceLimitExceeded	The card has exceeded the maximum cash limit per transaction.	✗
<b>1050</b>	DeclinedCardDailyOtherLimitExceeded	The card has exceeded the daily money limit on dry goods.	✗
<b>1050</b>	DeclinedCardWeeklyOtherLimitExceeded	The card has exceeded the weekly money limit on dry goods.	✗
<b>1050</b>	DeclinedCardMonthlyOtherLimitExceeded	The card has exceeded the monthly money limit on dry goods.	✗
<b>1050</b>	DeclinedCardTxFuelAmountLimitExceeded	The card has exceeded the maximum amount of fuel per	✗

		transaction.	
<b>1050</b>	DeclinedCardTxOtherPriceLimitExceeded	The card has exceeded the maximum price limit for dry goods per transaction.	<b>X</b>
<b>1051</b>	DeclinedExpiredCard	Expired card.	<b>X</b>
<b>1052</b>	DeclinedNumberOfPINAttemptsExceeded	Number of PIN tries exceeded.	<b>X</b>
<b>1053</b>	DeclinedNoSharingAllowed	No sharing allowed.	<b>X</b>
<b>1054</b>	DeclinedNoSecurityModule	No security module.	<b>X</b>
<b>1055</b>	DeclinedInvalidTransaction	Invalid transaction.	<b>X</b>
<b>1056</b>	DeclinedNotSupportedByInstitution	Transaction not supported by institution.	<b>X</b>
<b>1057</b>	DeclinedLostOrStolenCard	Lost or stolen card.	<b>X</b>
<b>1058</b>	DeclinedInvalidCardStatus	Invalid card status.	<b>X</b>
<b>1059</b>	DeclinedRestrictedStatus	Restricted status.	<b>X</b>
<b>1060</b>	DeclinedAccountNotFoundInCardholderDatabase	Account not found in cardholder database.	<b>X</b>
<b>1061</b>	DeclinedPositiveBalanceAccountRecordNotFound	Positive balance account record not found.	<b>X</b>
<b>1061</b>	DeclinedPositiveBalanceAccountRecordNotFound	Positive balance account update error.	<b>X</b>
<b>1063</b>	DeclinedInvalidAuthorizationTypeInInstitutionDatabase	Invalid authorization type in institution database.	<b>X</b>
<b>1064</b>	DeclinedBadTrackInformation	Bad track information.	<b>X</b>
<b>1065</b>	DeclinedAdjustmentNotAllowedInInstitutionDatabase	Adjustment not allowed in the institution database.	<b>X</b>
<b>1066</b>	DeclinedInvalidCreditCardAdvanceIncrement	Invalid credit card advance increment.	<b>X</b>
<b>1067</b>	DeclinedInvalidTransactionDate	Invalid transaction date.	<b>X</b>
<b>1068</b>	DeclinedTransactionLogFileError	Transaction log file error.	<b>X</b>
<b>1069</b>	DeclinedBadMessageEdit	Bad message edit.	<b>X</b>
<b>1070</b>	DeclinedNoInstitutionDatabaseRecord	No institution database record.	<b>X</b>
<b>1071</b>	DeclinedInvalidRoutingToHostApplication	Invalid routing to host application.	<b>X</b>
<b>1072</b>	DeclinedCardOnNationalNegativeFile	Card on national negative file.	<b>X</b>
<b>1073</b>	DeclinedInvalidRoutingAuthorizationService	Invalid routing authorization service.	<b>X</b>

<b>1074</b>	DeclinedUnableToAuthorize	Unable to authorize.	<b>X</b>
<b>1074</b>	DeclinedUnableToAuthorizeRepeat	Unable to authorize, retry transaction.	<b>X</b>
<b>1075</b>	DeclinedInvalidPANLength	Invalid PAN length.	<b>X</b>
<b>1076</b>	DeclinedInsufficientFundsInPositiveBalanceAccount	Insufficient funds in a positive balance account.	<b>X</b>
<b>1077</b>	DeclinedPreauthorizationFull	Preauthorization full.	<b>X</b>
<b>1078</b>	DeclinedDuplicateTransactionReceived	Duplicate transaction received.	<b>X</b>
<b>1079</b>	DeclinedMaximumOnlineRefundReached	Maximum online refund reached.	<b>X</b>
<b>1080</b>	DeclinedMaximumOfflineRefundReached	Maximum offline refund reached.	<b>X</b>
<b>1081</b>	DeclinedMaximumCreditPerRefundReached	Maximum credit per refund reached.	<b>X</b>
<b>1082</b>	DeclinedMaximumNumberOfTimesUsed	Maximum number of times used.	<b>X</b>
<b>1083</b>	DeclinedMaximumRefundCreditReached	Maximum refund credit reached.	<b>X</b>
<b>1084</b>	DeclinedCustomerSelectedNegativeCardFileReason	Customer selected negative card file reason.	<b>X</b>
<b>1085</b>	DeclinedInquiryNotAllowedNoBalances	Inquiry not allowed—no balances.	<b>X</b>
<b>1086</b>	DeclinedOverFloorLimit	Over floor limit.	<b>X</b>
<b>1087</b>	DeclinedMaximumNumberRefundCreditsReached	Maximum number refund credits reached.	<b>X</b>
<b>1088</b>	DeclinedPlaceCall	Place call.	<b>X</b>
<b>1089</b>	DeclinedCardStatusIsInactiveOrClosed	Card status equals 0 (inactive) or 9 (closed).	<b>X</b>
<b>1090</b>	DeclinedReferralFileFull	Referral file full.	<b>X</b>
<b>1091</b>	DeclinedProblemAccessingNegativeCardFile	Problem accessing negative card file.	<b>X</b>
<b>1092</b>	DeclinedAdvanceLessThanMinimum	Advance less than minimum.	<b>X</b>
<b>1093</b>	DeclinedDelinquent	Delinquent.	<b>X</b>
<b>1094</b>	DeclinedOverLimitTableOrExceedsAmountAvailable	Over limit table or exceeds the amount available.	<b>X</b>
<b>1095</b>	DeclinedAmountOverMaximum	Amount over maximum.	<b>X</b>
<b>1096</b>	DeclinedPINRequired	PIN required.	<b>X</b>
<b>1097</b>	DeclinedMod10Check	Mod 10 check.	<b>X</b>

<b>1098</b>	DeclinedForcePost	Force post.	<b>X</b>
<b>1099</b>	DeclinedCouldNotAccessPositiveBalanceAccountInDatabase	Could not access positive balance account in database	<b>X</b>
<b>1100</b>	ReferralUnableToProcessTransaction	Unable to process a transaction.	<b>X</b>
<b>1101</b>	ReferralUnableToAuthorizeIssueCall	Unable to authorize—issue call.	<b>X</b>
<b>1102</b>	ReferralCall	Call.	<b>X</b>
<b>1103</b>	ReferralProblemAccessingNegativeCardFile	Problem accessing negative card file.	<b>X</b>
<b>1104</b>	ReferralProblemAccessingCardholderAccount	Problem accessing cardholder account.	<b>X</b>
<b>1105</b>	ReferralCardNotSupported	Card not supported.	<b>X</b>
<b>1105</b>	ReferralCardNotFound	Card not found.	<b>X</b>
<b>1105</b>	ReferralCardReplenishmentSupported	Card replenishment not supported.	<b>X</b>
<b>1106</b>	ReferralAmountOverMaximum	Amount over maximum.	<b>X</b>
<b>1107</b>	ReferralOverDailyLimit	Over daily limit.	<b>X</b>
<b>1108</b>	ReferralCardAuthorizationParametersNotFound	Card authorization parameters not found.	<b>X</b>
<b>1109</b>	ReferralAdvanceLessThanMinimum	Advance less than minimum.	<b>X</b>
<b>1110</b>	ReferralNumberTimesUsed	Number times used.	<b>X</b>
<b>1111</b>	ReferralDelinquent	Delinquent.	<b>X</b>
<b>1112</b>	ReferralOverLimitTable	Over limit table	<b>X</b>
<b>1113</b>	ReferralTimeout	Timeout.	<b>X</b>
<b>1115</b>	ReferralTransactionLogFileFull	Transaction log file full.	<b>X</b>
<b>1120</b>	ReferralProblemAccessingCardholderUsageAccumulationData	Problem accessing cardholder usage accumulation data.	<b>X</b>
<b>1121</b>	ProblemAccessingAdministrativeCardData	Problem accessing administrative card data.	<b>X</b>
<b>1122</b>	ReferralUnableToValidatePIN	Unable to validate PIN; security module is down.	<b>X</b>
<b>1130</b>	ReferralAuthorizationRequestCryptogram	Authorization request cryptogram (ARQC) referral.	<b>X</b>
<b>1131</b>	ReferralCardVerificationResults	Card verification results (CVR) referral.	<b>X</b>
<b>1132</b>	ReferralTerminalVerificationResults	Terminal verification results (TVR) referral.	<b>X</b>
<b>1133</b>	ReferralReasonOnlineCode	Reason online code referral.	<b>X</b>

<b>1134</b>	ReferralFallback	Fallback referral.	<b>X</b>
<b>1150</b>	ServiceMerchantNotOnFile	Merchant not on file.	<b>X</b>
<b>1150</b>	ServiceMerchantTerminalDisabled	Blocked terminal.	<b>X</b>
<b>1150</b>	ServiceMerchantAcquirerDisabled	Blocked terminal owner.	<b>X</b>
<b>1200</b>	TransactionInvalidAccount	Invalid account.	<b>X</b>
<b>1201</b>	TransactionIncorrectPIN	Incorrect PIN.	<b>X</b>
<b>1202</b>	TransactionCashAdvancelLessThanMinimum	Cash advance is less than minimum.	<b>X</b>
<b>1203</b>	TransactionAdministrativeCardNeeded	Administrative card needed.	<b>X</b>
<b>1204</b>	TransactionEnterLesserAmount	Enter a lesser amount (This code can also be used when the transaction amount exceeds the retailer ceiling limits.).	<b>X</b>
<b>1205</b>	TransactionInvalidAdvanceAmount	Invalid advance amount.	<b>X</b>
<b>1206</b>	TransactionCardholderAuthorizationDataNotFound	Cardholder authorization data not found.	<b>X</b>
<b>1207</b>	TransactionInvalidTransactionDate	Invalid transaction date.	<b>X</b>
<b>1208</b>	TransactionInvalidExpirationDate	Invalid expiration date.	<b>X</b>
<b>1209</b>	TransactionInvalidTransactionCode	Invalid transaction code.	<b>X</b>
<b>1251</b>	TransactionCashBackExceedsDailyLimit	Cash back exceeds daily limit.	<b>X</b>
<b>1400</b>	TransactionAuthorizationRequestCryptogramFailure	Authorization request cryptogram (ARQC) failure.	<b>X</b>
<b>1401</b>	TransactionHardwareSecurityModuleParameterError	Hardware security module parameter error.	<b>X</b>
<b>1402</b>	TransactionHardwareSecurityModuleFailure	Hardware security module failure.	<b>X</b>
<b>1403</b>	TransactionIntegratedCircuitCardKeyInformationNotFound	Integrated circuit card key information not found.	<b>X</b>
<b>1404</b>	TransactionApplicationTransactionCounterCheckFailure	Application transaction counter (ATC) check failure.	<b>X</b>
<b>1405</b>	TransactionCardVerificationResultsDecline	Card verification results (CVR) decline.	<b>X</b>
<b>1406</b>	TransactionTerminalVerificationResultsDecline	Terminal verification results (TVR) decline.	<b>X</b>
<b>1407</b>	TransactionReasonOnlineCodeDecline	Reason online code decline.	<b>X</b>
<b>1408</b>	TransactionFallbackDecline	Fallback decline.	<b>X</b>

<b>1500</b>	ProductDenied	Unauthorized product.	<b>X</b>
<b>1557</b>	SuspectedFraudCallWag	WAG antifraud. Suspected fraud, call WAG	<b>X</b>
<b>1558</b>	LockedCardSendSMS	WAG antifraud. Card locked, for unlock card send SMS to WAG	<b>X</b>
<b>1800</b>	TransactionFormatError	Format error.	<b>X</b>
<b>1801</b>	TransactionInvalidData	Invalid data.	<b>X</b>
<b>1802</b>	TransactionInvalidEmployeeNumber	Invalid employee number.	<b>X</b>
<b>1809</b>	TransactionInvalidCloseTransaction	Invalid close transaction.	<b>X</b>
<b>1810</b>	TransactionTimeout	Transaction timeout.	<b>X</b>
<b>1811</b>	TransactionSystemError	System error.	<b>X</b>
<b>1811</b>	SuperiorAuthorizationSystemError	Authorization failed on the master authorization system.	<b>X</b>
<b>1820</b>	TransactionInvalidTerminalIdentifier	Invalid terminal identifier.	<b>X</b>
<b>1821</b>	TransactionInvalidResponseLength	Invalid response length.	<b>X</b>
<b>1870</b>	MailDelivered	Mail delivered.	✓
<b>1871</b>	MailStored	Mail stored.	✓
<b>1880</b>	MailOrDownloadMessageHasBeenReceivedInItsEntirety	Mail or download message has been received in its entirety.	✓
<b>1881</b>	MailOrDownloadMessageReceivedSuccessfullyAndNoMoreData	Mail or download message received successfully and there is more data for this mail message.	✓
<b>1882</b>	DownloadAborted	Download aborted (call for service).	<b>X</b>
<b>1878</b>	DeclineIncorrectPINLengthError	Incorrect PIN length error.	<b>X</b>
<b>1889</b>	DeclineMACCommunicationsKeySynchronizationError	MAC communications key (KMAC) synchronization error.	<b>X</b>
<b>1898</b>	DeclineInvalidMAC	Invalid MAC.	<b>X</b>
<b>1899</b>	DeclineSequenceErrorResync	Sequence error—resync.	<b>X</b>
<b>1900</b>	POSCaptureNumberOfPINAttemptsExceeded	Number of PIN tries exceeded.	<b>X</b>
<b>1901</b>	POSCaptureExpiredCard	Expired card.	<b>X</b>
<b>1902</b>	POSCaptureNegativeCardFileCaptureCode	Negative card file capture code.	<b>X</b>
<b>1903</b>	POSCaptureCardStatusIsStolen	Card status is 3 (stolen).	<b>X</b>
<b>1904</b>	POSCaptureAdvanceLessThanMinimum	Advance less than minimum.	<b>X</b>

<b>1905</b>	POSCaptureNumberTimesUsedExceeded	Number times used exceeded.	<b>X</b>
<b>1906</b>	POSCaptureDelinquent	Delinquent.	<b>X</b>
<b>1907</b>	POSCaptureOverLimitTable	Over limit table.	<b>X</b>
<b>1908</b>	POSCaptureAmountOverMaximum	Amount over maximum.	<b>X</b>
<b>1909</b>	POSCaptureCapture	Capture.	✓
<b>1910</b>	POSCaptureAuthorizationRequestCryptogramCapture	Authorization request cryptogram (ARQC) capture.	✓
<b>1911</b>	POSCaptureCardVerificationResultsCapture	Card verification results (CVR) capture.	✓
<b>1912</b>	POSCaptureTerminalVerificationResultsCapture	Terminal verification results (TVR) capture.	✓
<b>1950</b>	DeclinedAdministrativeCardNotFound	Administrative card not found.	<b>X</b>
<b>1951</b>	DeclinedAdministrativeCardNotAllowed	Administrative card not allowed.	<b>X</b>
<b>1959</b>	DeclinedAdministrativeTransactionsNotSupported	Administrative transactions not supported.	<b>X</b>
<b>1952</b>	ApprovedAdministrativeRequestInWindow	Approved administrative request—in window.	✓
<b>1953</b>	ApprovedAdministrativeRequestOutOfWindow	Approved administrative request—out of window.	✓
<b>1954</b>	ApprovedAdministrativeRequestAnytime	Approved administrative request—anytime.	✓
<b>1955</b>	ChargebackCustomerFileUpdated	Chargeback—customer file updated.	✓
<b>1956</b>	ChargebackCustomerFileUpdatedAcquirerNotFound	Chargeback—customer file updated—acquirer not found.	<b>X</b>
<b>1957</b>	ChargebackIncorrectPrefixNumber	Chargeback—incorrect prefix number.	<b>X</b>
<b>1958</b>	ChargebackIncorrectResponseCodeOrCardPrefixConfiguration	Chargeback—incorrect response code or card prefix configuration.	<b>X</b>
<b>1960</b>	ChargebackApprovedCustomerFileNotUpdated	Chargeback—approved customer file not updated.	<b>X</b>
<b>1961</b>	ChargebackApprovedCustomerFileNotUpdatedAcquirerNotFound	Chargeback—approved customer file not updated, acquirer not found.	<b>X</b>
<b>1962</b>	ChargebackAcceptedIncorrectDestination	Chargeback—accepted, incorrect destination.	<b>X</b>
<b>1963</b>	NoCardTimeout	No card timeout	<b>X</b>
<b>1964</b>	InvalidSignature	Invalid signature	<b>X</b>

<b>2000</b>	PAX_DEFINE	ReturnCodes by PAX	-
<b>2001</b>	ICC_RESET_ERR	IC cardreset is failed	<b>X</b>
<b>2002</b>	ICC_CMD_ERR	IC card command is failed	<b>X</b>
<b>2003</b>	ICC_BLOCK	IC card is blocked	<b>X</b>
<b>2004</b>	EMV_RSP_ERR	Status Code returned by IC card is not 9000	<b>X</b>
<b>2005</b>	EMV_APP_BLOCK	The application selected is blocked	<b>X</b>
<b>2006</b>	EMV_NO_APP	There is no AID matched between ICC and terminal	<b>X</b>
<b>2007</b>	EMV_USER_CANCEL	The current operation or transaction was canceled by user	<b>X</b>
<b>2008</b>	EMV_TIME_OUT	User's operation is timeout	<b>X</b>
<b>2009</b>	EMV_DATA_ERR	Data error is found	<b>X</b>
<b>2010</b>	EMV_NOT_ACCEPT	Transaction is not accepted	<b>X</b>
<b>2011</b>	EMV_DENIAL	Transaction is denial	<b>X</b>
<b>2012</b>	EMV_KEY_EXP	Certification Authority Public Key is Expired	<b>X</b>
<b>2013</b>	EMV_NO_PINPAD	PIN enter is required, but PIN pad is not present or not working	<b>X</b>
<b>2014</b>	EMV_NO_PASSWORD	PIN enter is required, PIN pad is present, but there is no PIN entered	<b>X</b>
<b>2015</b>	EMV_SUM_ERR	Checksum of CAPK is error	<b>X</b>
<b>2016</b>	EMV_NOT_FOUND	Appointed Data element can't be found	<b>X</b>
<b>2017</b>	EMV_NO_DATA	The length of the appointed Data Element is 0	<b>X</b>
<b>2018</b>	EMV_OVERFLOW	Memory is overflow	<b>X</b>
<b>2019</b>	NO_TRANS_LOG	There is no Transaction log	<b>X</b>
<b>2020</b>	RECORD_NOTEXIST	Appointed log is not existed	<b>X</b>
<b>2021</b>	LOGITEM_NOTEXIST	Appointed Label is not existed in current log record	<b>X</b>
<b>2022</b>	ICC_RSP_6985	Status Code returned by IC card for GPO is 6985	<b>X</b>
<b>2023</b>	CLSS_USE_CONTACT	Must use other interface for the transaction	<b>X</b>
<b>2024</b>	EMV_FILE_ERR	There is file error found	<b>X</b>
<b>2025</b>	CLSS_TERMINATE	Must terminate the transaction	<b>X</b>

<b>2026</b>	CLSS_FAILED	Contactless transaction is failed	<b>X</b>
<b>2027</b>	CLSS_DECLINE	Transaction should be declined	<b>X</b>
<b>2028</b>	CLSS_TRY_ANOTHER_CARD	Try another card (DPAS Only)	<b>X</b>
<b>2030</b>	CLSS_PARAM_ERR	Parameter is error = EMV_PARAM_ERR	<b>X</b>
<b>2031</b>	CLSS_WAVE2_OVERSEA	International transaction (for VISA AP payWaveLevel2 IC card use)	<b>X</b>
<b>2033</b>	CLSS_WAVE2_US_CARD	US card (for VISA AP payWaveL2 IC card use)	<b>X</b>
<b>2034</b>	CLSS_WAVE3_INS_CARD	Need to use IC card for the transaction (for VISA payWaveIC card use)	<b>X</b>
<b>2035</b>	CLSS_RESELECT_APP	Select the next AID in candidate list	<b>X</b>
<b>2036</b>	CLSS_CARD_EXPIRED	IC card is expired	<b>X</b>
<b>2037</b>	EMV_NO_APP_PPSE_ERR	Select PPSE command returns error code, or there is no matching Application	<b>X</b>
<b>2038</b>	CLSS_USE_VSDC	Switch to Contactless PBOC	<b>X</b>
<b>2039</b>	CLSS_CVMDECLINE	DVM result is decline	<b>X</b>
<b>2040</b>	CLSS_REFERER_CONSUMER_DEVICE	Status Code returned by IC card is 6986, please see phone	<b>X</b>
<b>2041</b>	CLSS_LAST_CMD_ERR	Terminal did not receive the response of the last command of Read Record. (qPBOC only)	<b>X</b>
<b>2042</b>	CLSS_API_ORDER_ERR	APIs are called in wrong order. Please call Clss_GetDebugInfo_xxx to get error codes.	<b>X</b>
<b>2043</b>	CLSS_TORN_CARDNUM_ERR	Torn log's PAN is different with the reselect card's pan	<b>X</b>
<b>2044</b>	CLSS_TRON_AID_ERR	Torn log's AID is different with the reselect card's AID	<b>X</b>
<b>2045</b>	CLSS_TRON_AMT_ERR	Torn log's amount is different with the current transaction's amount	<b>X</b>
<b>2046</b>	CLSS_CARD_EXPIRED_REQ_ONLINE	Card expired and the transaction should go online	<b>X</b>
<b>2047</b>	CLSS_FILE_NOT_FOUND	ICC return 6A82 (File not found) in response to the SELECT command	<b>X</b>
<b>2048</b>	CLSS_TRY_AGAIN	Try again	<b>X</b>
<b>2200</b>	CLSS_PAYMENT_NOT_ACCEPT	Payment Type Not Accepted for flash	<b>X</b>
<b>2301</b>	CLSS_INSERTED_ICCCARD	IC card is detected during	<b>X</b>

		contactless transaction	
<b>2302</b>	CLSS_SWIPED_MAGCARD	Magnetic stripe card is detected during contactless transaction	<b>X</b>
<b>3000</b>	MONEMV_CLS_KERNEL	Monet kernel	-
<b>3256</b>	MONEMV_CLS_KERNEL_STATUS_OK	The function is correctly executed.	✓
<b>3257</b>	MONEMV_CLS_KERNEL_STATUS_SERVICE_NOT_AVAILABLE	The kernel to be accessed is not loaded in the terminal.	<b>X</b>
<b>3258</b>	MONEMV_CLS_KERNEL_STATUS_DATABASE_ERROR	A database error occurred.	<b>X</b>
<b>3259</b>	MONEMV_CLS_KERNEL_STATUS_INVALID_INPUT_DATA	One of the provided data is incorrect.	<b>X</b>
<b>3260</b>	MONEMV_CLS_KERNEL_STATUS_NOT_SUPPORTED	The called function is not supported by the kernel.	<b>X</b>
<b>3261</b>	MONEMV_CLS_KERNEL_STATUS_LACK_OF_MEMORY	There is not enough memory to complete the transaction.	<b>X</b>
<b>3262</b>	MONEMV_CLS_KERNEL_STATUS_COMMUNICATION_ERROR	A communication error occurred with the contactless card.	<b>X</b>
<b>3263</b>	MONEMV_CLS_KERNEL_STATUS_MISSING_INPUT_DATA	A mandatory data is missing to perform the transaction.	<b>X</b>
<b>3264</b>	MONEMV_CLS_KERNEL_STATUS_ICC_MISSING_DATA	A mandatory ICC data is missing to perform the transaction.	<b>X</b>
<b>3265</b>	MONEMV_CLS_KERNEL_STATUS_ICC_INVALID_DATA	The data returned by the card is not correctly formatted.	<b>X</b>
<b>3266</b>	MONEMV_CLS_KERNEL_STATUS_ICC_REDUNDANT_DATA	Card returned redundant data.	<b>X</b>
<b>3267</b>	MONEMV_CLS_KERNEL_STATUS_ICC_DATA_FORMAT_ERROR	The card response is not correctly formatted (parsing error, empty response, etc).	<b>X</b>
<b>3268</b>	MONEMV_CLS_KERNEL_STATUS_TERMINAL_MISSING_DATA	A mandatory terminal data is missing to perform the transaction.	<b>X</b>
<b>3269</b>	MONEMV_CLS_KERNEL_STATUS_CARD_BLOCKED	A Card is blocked and the transaction cannot be performed.	<b>X</b>
<b>3270</b>	MONEMV_CLS_KERNEL_STATUS_APPLICATION_BLOCKED	The application in the card is blocked.	<b>X</b>
<b>3271</b>	MONEMV_CLS_KERNEL_STATUS_REMOVE_AID	The AID shall be removed from the candidate list.	<b>X</b>
<b>3272</b>	MONEMV_CLS_KERNEL_STATUS_UNKNOWN_SW	Unexpected response code from the ICC.	<b>X</b>
<b>3273</b>	MONEMV_CLS_KERNEL_STATUS_CONDITION_OF_USE_NOT_SATISFIED	Conditions of use not satisfied" in the GPO response.	<b>X</b>
<b>3274</b>	MONEMV_CLS_KERNEL_STATUS_OFFLINE_APPROVED	Transaction succeeded and is offline approved.	✓

<b>3275</b>	MONEMV_CLS_KERNEL_STATUS_OFFLINE_DECLINED	Transaction is offline declined.	<b>X</b>
<b>3276</b>	MONEMV_CLS_KERNEL_STATUS_ONLINE_AUTHORISATION	Transaction succeeded and needs an online authorisation to be completed.	-
<b>3277</b>	MONEMV_CLS_KERNEL_STATUS_CANCELLED	Transaction has been cancelled.	<b>X</b>
<b>3278</b>	MONEMV_CLS_KERNEL_STATUS_USE_CONTACT_INTERFACE	The transaction has to be conducted over another interface (chip or swipe for example)	<b>X</b>
<b>3279</b>	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.	<b>X</b>
<b>3280</b>	MONEMV_CLS_KERNEL_STATUS_CONTINUE	The transaction flow must continue.	-
<b>3281</b>	MONEMV_CLS_KERNEL_STATUS_SUSPEND	The transaction flow must be stopped.	<b>X</b>
<b>3282</b>	MONEMV_CLS_KERNEL_STATUS_INTERNAL_ERROR	A kernel internal error occurred.	<b>X</b>
<b>3283</b>	MONEMV_CLS_KERNEL_STATUS_LIBRARY_INTERFACE_ERROR	An error occurred into the kernel interface (linked to the custom application).	<b>X</b>
<b>3284</b>	MONEMV_CLS_KERNEL_STATUS_EXPIRED_CERTIFICATE	The certificate used for Offline data authentication is expired.	<b>X</b>
<b>3285</b>	MONEMV_CLS_KERNEL_STATUS_REVOKED_CERTIFICATE	The certificate used for Offline data authentication is revoked.	<b>X</b>
<b>3286</b>	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 ...")	<b>X</b>
<b>3287</b>	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) ...")	<b>X</b>
<b>3383</b>	MONEMV_CLS_KERNEL_STATUS_UNKNOWN	Unknown	<b>X</b>
<b>4000</b>	FEITIAN_READER	Feitian reader	-
<b>8005</b>	DOMESTIC_CARD_ONLY_ERROR	This error code is used for the Cashback operation and indicates that for this operation, only domestic cards can be used. This means that the terminal country code has to be the same as the card country code. If the card country code differs from the	<b>X</b>



		terminal country code, this error occurs.	
<b>8008</b>	READING_CARD_FAILED	Error when card was not processed correctly	<b>X</b>
<b>8009</b>	SECOND_TAP_CARD_MISMATCH	Error occurred when a different card was used for the required second tap of the transaction.	<b>X</b>
<b>21474 83647</b>	UNDEFINED	Undefined	-

## PinCommunicationsKey

```
public class PinCommunicationsKey {
    private PinCommunicationsKey.KeyLength keyLength;
    private byte[] key;
}

public static enum KeyLength {
    SINGLE(16),
    DOUBLE(32),
    TRIPLE(48);
}
}
```

## EmvResponseData

PARAMETER	TYPE	DESCRIPTION
smartCardScheme	int	Scheme of the smart card
authorisationResponseCode	string	The response code of authorization.
issuerAuthenticationData	string	Information whether is transaction confirmed or not

## EmvAdditionalResponseData

```
public class EmvAdditionalResponseData {
    private Integer smartCardScheme;
    private String issuerScriptData;
}
```

## CvmType

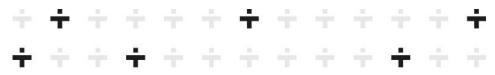
CvmTypeList
NO
SIGNATURE
PIN_ONLINE
PIN_OFFLINE
CONSUMER_DEVICE

## CardInputType

CardInputType
MAGNETIC
MAGNETIC_FALLBACK
CONTACT
CLESS
CLESS_MAGSTRIPE
MANUAL

## TransactionType parameter

TransactionType
SALE_ONLINE
SALE_OFFLINE
QR_SALE
TOKEN
READ_CARD
READ_CARD_PAYMENT
CASHBACK_ONLINE
PREAUTHORIZATION
PREAUTHORIZATION_INCREMENT
PREAUTHORIZATION_DECREMENT
PREAUTHORIZATION_COMPLETION
EXTERNAL_PREAUTHORIZATION_INCREMENT
EXTERNAL_PREAUTHORIZATION_DECREMENT
EXTERNAL_PREAUTHORIZATION_COMPLETION
MOTO_SALE
MOTO_PREAUTHORIZATION
RETURN_ONLINE
RETURN_OFFLINE
REFFERAL_RETURN
CLOSE_BATCH



AUTO_CLOSE_BATCH
SUBTOTAL_BATCH
HANDSHAKE
TRANSACTION_STATUS
CARD_VERIFY
PURCHASE_ADJUSTMENT
UNDEFINED

## BatchTotals parameter

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