MONET +
Trust in digital

TECHNICAL DOCUMENT

Switchio Pay – ECR protocol

SUBMITTED BY	**MONET+,a.s.** Za Dvorem 505, Zlín - Štípa Czech Republic
UPDATED	**5. 11. 2024**
VERSION	**6.10**

CONTENT

CONTENT **1**
1. DOCUMENT VERSION HISTORY **5**
2. APPLICATION VERSION HISTORY **7**
3. INTRODUCTION **8**
3.1. INTENT 8
3.2. REQUEST 8
3.3. RESULT 9
3.3.1. Result structures and examples for each API version 9
Result structure v1 10
Result structure v2 11
Result structure v3 14
Result structure v4 16
Result structure v5 19
Result structure v6 21
4. CONTROL ENDPOINTS **25**
4.1. CLOSING BATCH 25
4.1.1. Request 25
4.1.2. Result 26
4.2. DAUGHTER COMPANIES 27
4.2.1. Request 27
4.2.2. Result 27
4.3. SCANNER 28
4.3.1. Request 28
4.3.2. Result 28
4.4. UPDATE TERMINAL CONFIGURATION 29
4.4.1. Request 29
4.4.2. Result 30
4.5. TOUCH - AUTOMATIC CONFIGURATION UPDATE 31
4.5.1. Broadcast 31
4.5.2. Request 32
4.5.3. Result 32
4.6. SUBTOTAL 33
4.6.1. Request 33
4.6.2. Result 33
4.7. HANDSHAKE 34
4.7.1. Request 34
4.7.2. Result 34
4.8. TOKEN 35
4.8.1. Request 35
4.8.2. Result 36
4.9. GET LAST TRANSACTION 36

4.9.1. Request 36
4.9.2. Result 37
4.10. GET ALL SUCCESSFUL TRANSACTIONS 38
4.10.1. Request 38
4.10.2. Result 39
4.11. CARD VERIFY 40
4.11.1. Request 40
4.11.2. Result 40
4.12. SETTINGS 43
4.12.1. Request 43
4.12.2. Result 43
4.13. OPEN PREAUTHORIZATIONS 44
4.13.1. Request 44
4.13.2. Result 44
5. RETAIL ENDPOINTS 46
5.1. PAYMENT 46
5.1.1. Request 46
5.1.2. Result 47
5.2. REFUND 48
5.2.1. Request 48
5.2.2. Result 49
5.3. PREAUTHORIZATION 50
5.3.1. Request 50
5.3.2. Result 51
5.4. INCREMENT PREAUTHORIZATION 52
5.4.1. Request 52
5.4.2. Result 53
5.5. DECREMENT PREAUTHORIZATION 54
5.5.1. Request 54
5.5.2. Result 54
5.6. COMPLETE PREAUTHORIZATION 55
5.6.1. Request 55
5.6.2. Result 56
5.7. CANCEL PREAUTHORIZATION 57
5.7.1. Request 57
5.7.2. Result 57
5.8. PREAUTHORIZATION - EXTERNAL COMPLETE 58
5.8.1. Request 58
5.8.2. Result 59
5.9. PREAUTHORIZATION - EXTERNAL CANCEL 60
5.9.1. Request 60
5.9.2. Result 60
5.10. GET TRANSACTION 61
5.10.1. Request 61
5.10.2. Result 61
5.11. REVERSAL 62

5.11.1. Request 62
5.11.2. Result 62
5.12. MOTO 63
5.12.1. Request 63
5.12.2. Result 64
5.13. TRANSACTION STATUS 65
5.13.1. Request 65
5.13.2. Result 65
5.14. REFERRAL REFUND 66
5.14.1. Request 66
5.14.2. Result 67
5.15. CASHBACK 68
5.15.1. Request 68
5.15.2. Result 68
5.16. QR PAYMENT 69
5.16.1. Request 69
5.16.2. Result 69
6. TRANSPORT ENDPOINTS 69
6.1. TRANSPORT INIT 70
6.1.1. Request 70
6.1.2. Result 70
6.2. TRANSPORT PAYMENT 71
6.2.1. Request 71
6.2.2. Result 72
6.3. CUSTOMER TAP 74
6.3.1. Request 74
6.3.2. Result 75
6.4. UPLOAD TRANSPORT TRANSACTIONS 76
6.4.1. Request 76
6.4.2. Result 76
6.5. CUSTOMER TAP SERVICE START 77
6.5.1. Request 77
6.5.2. Result 77
6.6. CUSTOMER TAP SERVICE STOP 78
6.6.1. Request 78
6.6.2. Result 78
6.7. GET TAPS 79
6.7.1. Request 79
6.7.2. Result 79
6.8. GET DENYLIST UPDATE 80
6.8.1. Request 80
6.8.2. Result 80
7. APPLICATION DESCRIPTION 81
7.1. Payment 81
7.2. Refund 82
7.3. Dashboard 82

7.4. The closing batch 83
7.5. Settings 84
7.6. Service Settings 85
7.6.1. Key injection 87
ATTACHMENT - ENUMS **89**
TerminalErrorCodes 89
PinCommunicationsKey 101
EmvResponseData 101
EmvAdditionalResponseData 101
CvmType 101
CardInputType 102
TransactionType parameter 102
BatchTotals parameter 103

1. DOCUMENT VERSION HISTORY

VERSION	DATE	AUTHOR	DESCRIPTION
1.0	29.10.2019	RTA	1st version of the document
2.0	13.01.2020	GBI	Documentation revision
3.0	07.01.2022	PZI	Request check: MOTO Payment, Last Transaction, Get Successful Transactions + added Reversal by ID
4.0	05.04.2022	PZI	ECR V3 since SwitchioPay 3.1.0
4.1	20.04.2022	GBI	Revision of the new chapter of ECR v3
4.2	11.07.2022	GBI AAD	• Added tables with field descriptions for individual operations • Added examples of individual calls in JSON
4.3	05.09.2022	XRTA AAD	added sections to ECR v3 • PreAuth External Complete • PreAuth External Cancel • PreAuth Increment • PreAuth Transaction • Closing batch • Daughter Companies • Scanner • new Transaction Type parameters deleted sections • Transport Reversal • Transport Refund
5.0	14.12.2022	M+(AAD)	Added ECR v4.
5.01	01.02.2023	M+(AAD)	Added two result lines to Card Verify ECR v4
5.02	01.03.2023	M+(AAD)	Added cardUid to Token - ECR v4
5.03	30.04.2023	M+(AAD) XRTA	Updated Result-V4 Added Configuration Update V4 result example
5.04	16.05.2023	XRTA	Fixed CardVerify v4 response
5.05	22.05.2023	M+(AAD)	Added OPEN PREAUTHORIZATION result example
5.06	07.06.2023	XPZI	Changed daughter_companies from main to light API
5.07	24.07.2023	M+(AAD)	Added ECR v5 - Result • monetToken

			• par
5.08	19.9.2023	M+(AAD)	Added Get taps v5, Get Denylist last update
5.09	23.11.2023	M+(KLA)	Restructured documentation and new information added: • Sections are now organized according to the endpoints (not API versions); • Added new endpoints: cashback, customer tap service start and customer tap service stop; • Completed information about all endpoints with examples and URIs for different versions.
5.10	23.2.2024	M+(KLA)	Updated the tms_update endpoint information: • v5 URI updated • added v5 request example • added info for Uri.Builder() - v5
6.00	10. 10. 2024	M+(KLA)	Added new endpoints: • DECREMENT PREAUTHORIZATION • QR PAYMENT Added new endpoint versions for v6. Added new Transaction result for v6 - Result structure v6. Minor changes and corrections
6.10	5. 11. 2024	M+(KLA)	Updated: • Transport payment request with new optional fields • Added new TerminalErrorCode

2. APPLICATION VERSION HISTORY

ENDPOINT VERSION	APPLICATION VERSION	DESCRIPTION OF CHANGES
v1	1.0.0	First release
v2	1.8.0	New endpoints: • Touch TMS Update • Subtotal • Get last transaction • Get all successful transactions • PreAuth cancel • Reversal by ID • Moto payment • Transaction status
v3	3.0.0	New endpoints: • PreAuth transactions • Daughter companies • Scanner • PreAuth increment • PreAuth external complete • PreAuth external cancel • Referral refund New field tokens New format of date for serialization
v4	3.4.0	New endpoints: • Intent • Result • Card verify New format of API calls
v5	4.1.0	New endpoints: • Get taps • Get denylist last update • Customer tap service start • Customer tap service stop • Cashback • QR payment (from v 4.4.0) • Decrement preauthorization (from v 4.4.0) New fields in transaction Result response: • par • monetToken
v6	4.5.0	New version of TransactionResult with new fields: • dccOffer • dccResult • surcharge • externalTransactionId New fields in the Transport payment request • transportAmount • transportCurrencyCode

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 five 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.

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.

Action:
```
switchio.pay.ECR
```

URI:
See selected operation URI example.

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

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

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

Input data structure
```

```
@Serializable
data class PaymentRequest(
       @SerialName("TransactionID")
       val transactionId: String,
       @SerialName("Amount")
       val amount: Int,
       @SerialName("CurrencyCode")
       val currencyCode: Int,
       @SerialName("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
transactionType	string	Transaction types; enums in table TransactionType parameter
responseCode	TerminalErrorCodes	Code of the response, see TerminalErrorCodes
responseMessage	string	Message of the response
sequenceNumber	string	Number of the sequence
authCode	string	Code of the authorization
reversal	boolean	Information if the transaction is a reversal or not
amount	int	Amount of the transaction in cents. 500 is 5.00 euro
invoiceNumber	string	Variable symbol, max length is 20 char
currency	int	Code of the currency
dateTimeTerminal	date	Date and time on terminal, when was the result made
dateTimeServer	date	Date and time on server when, was the result made
aid	string	Application identifier
pan	string	Primary account number
panSequenceNumber	string	Sequence number of the primary account number
expiration	string	Expiration date of the card
conto	string	Card issuer
appLabel	string	Label of application
pinCommunicationsKey	PinCommunicationsKey	Type communication key PIN, enums PinCommunicationsKey
emvResponseData	EmvResponseData	Object of the response EmvResponseData
emvAdditionalResponseData	EmvAdditionalResponseData	Object of the response EmvAdditionalResponseData
aosa	int	Available offline sending amount
ErrorFlag	char	Error flag
emvData	EmvTlv	EMV data
cvmTypeList	List<CvmType>	Type of the TRX. Enum is in CvmType

batchTotals	batchTotals	Object of the BatchTotals parameter

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": "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": ["fkdfkjbekjrhljb" ]  //default - empty list
}
```


Result structure v2

ENTITY	TYPE	DESCRIPTION
transactionId	string	ID of the transaction.
transactionType	TransactionType	Transaction types; enums in table TransactionType parameter
aid	string	OPTIONAL Application identifier
amount	int	OPTIONAL Amount of the transaction in cents. 500 is 5.00 euro
appLabel	string	OPTIONAL Label of application

authCode	string	Code of the authorization
callReversal	boolean	Information on whether a reversal was generated for this transaction
reversal	boolean	Information if the transaction is a reversal or not
conto	string	OPTIONAL Card issuer
currency	int	OPTIONAL Code of the currency
dateTimeTerminal	string	Date and time on terminal, when was the result made
dateTimeServer	string	OPTIONAL Date and time on server when, was the result made
emvResponseData	EmvResponseData	OPTIONAL Object of the response EmvResponseData
expiration	string	OPTIONAL Expiration date of the card
invoiceNumber	string	OPTIONAL Variable symbol, max length is 20 char
responseCode	TerminalErrorCodes	Code of the response, see TerminalErrorCodes
responseMessage	string	Message of the response
terminalIdAcquirer	string	OPTIONAL ID of the acquirer's terminal
terminalIdIssuer	string	OPTIONAL ID of the issuer's terminal
merchantIdAcquirer	string	OPTIONAL ID of the merchant's acquirer
merchantIdIssuer	string	OPTIONAL ID of the merchant's issuer
pan	string	OPTIONAL Primary account number
token	string	OPTIONAL Encrypted combination of data
panSequenceNumber	string	OPTIONAL Sequence number of the primary account number
aosa	int	OPTIONAL Available offline sending amount
cardHolderMessage	string	OPTIONAL Message from the server for the cardholder
merchantMessage	string	OPTIONAL

		Message from the server for the merchant
availableBalance	string	OPTIONAL Balance of the available funds
sequenceNumber	string	Number of the sequence
cvmTypeList	List<CvmType>	Type of the TRX. Enum is in CvmType
cardInputType	CardInputType	OPTIONAL Type of the input. Enum is in the CardInputType
isReversed	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": ["fkdfkjbekjrhljb" ]  //default - empty list
}
```

Result structure v3

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

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

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": "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": ["fkdfkjbekjrhljb"]  //default - empty list
}
```


Result structure v4

PARAMETER	TYPE	DESCRIPTION
aid	string	OPTIONAL Application identifier.
amount	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).
aosa	int	OPTIONAL Available offline sending amount.
appLabel	string	OPTIONAL Label of application.
authCode	string	OPTIONAL Code of the authorization.
availableBalance	string	OPTIONAL Balance of the available funds.
brand	string	OPTIONAL
cardInputType	string	OPTIONAL Type of the input. Enum is in the CardInputType
cardHolderMessaage	string	OPTIONAL Card issuer.
currency	int	OPTIONAL ISO 4217 currency code.
cvmTypeList	string array	OPTIONAL Type of the transaction. CvmType
dateTimeServer	string	OPTIONAL Date and time on the server when the result was made.
dateTimeTerminal	string	OPTIONAL Date and time on the terminal when the

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

transactionType	string	Transaction type. TransactionTypes
tokens	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": "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": ["fkdfkjbekjrhljb"]  //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
aid	string	OPTIONAL Application identifier.
amount	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).
aosa	int	OPTIONAL Available offline sending amount.
appLabel	string	OPTIONAL Label of application.
authCode	string	OPTIONAL Code of the authorization.
availableBalance	string	OPTIONAL Balance of the available funds.
brand	string	OPTIONAL
cardInputType	string	OPTIONAL Type of the input. Enum is in the CardInputType
cardToken	string	OPTIONAL
cardHolderMessaage	string	OPTIONAL Card issuer.
currency	int	OPTIONAL ISO 4217 currency code.
cvmTypeList	string array	OPTIONAL Type of the transaction. CvmType
dateTimeServer	string	OPTIONAL Date and time on the server when the result was made.
dateTimeTerminal	string	OPTIONAL Date and time on the terminal when the result was made.
emvResponseData	object	OPTIONAL Object of the response. EmvResponseData
smartCardScheme	string	OPTIONAL Scheme of the smart card.
authorisationResponseCode	string	OPTIONAL The response code of authorization.
expiration	string	OPTIONAL Expiration date of the card.

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

Result structure v6

PARAMETER	TYPE	DESCRIPTION
aid	string	OPTIONAL Application identifier.
amount	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).
aosa	int	OPTIONAL Available offline sending amount.
appLabel	string	OPTIONAL Label of application.
authCode	string	OPTIONAL Code of the authorization.
availableBalance	string	OPTIONAL Balance of the available funds.
brand	string	OPTIONAL
cardInputType	string	OPTIONAL Type of the input. Enum is in the CardInputType
cardToken	string	OPTIONAL
cardHolderMessaage	string	OPTIONAL Card issuer.
currency	int	OPTIONAL ISO 4217 currency code.
cvmTypeList	string array	OPTIONAL Type of the transaction. CvmType
dateTimeServer	string	OPTIONAL Date and time on the server when the result was made.
dateTimeTerminal	string	OPTIONAL Date and time on the terminal when the result was made.
dccOffer	object	OPTIONAL dccOffer object
type	string	Card type. Possible values: • VISA • MASTERCARD
currency	string	Name of the DCC currency

amountExponent	int	Amount exponent. Indicates how many decimal places the amount is calculated.
convertedAmount	long	Converted amount
rateExponent	int	DCC rate exponent. Indicates how many decimal places the currency is calculated.
rate	long	Conversion rate
language	string	Language code
ecbRateExponent	int	European Central Bank rate exponent. Indicates how many decimal places the currency is calculated.
ecbRate	long	European Central Bank rate.
markup	int	Markup amount
currencyCode	int	Code of the DCC currency
dccResult	string	OPTIONAL DCC result. Possible values: • PERFORMED • NOT_PERFORMED
emvResponseData	object	OPTIONAL Object of the response. EmvResponseData
smartCardScheme	string	OPTIONAL Scheme of the smart card.
authorisationResponseCode	string	OPTIONAL The response code of authorization.
expiration	string	OPTIONAL Expiration date of the card.
externalTransactionId	string	OPTIONAL Transaction ID in an external system.
invoiceNumber	string	OPTIONAL Variable symbol, max length is 20 char.
isReversed	boolean	OPTIONAL Information whether the transaction was reversed.
merchantIdAcquirer	string	OPTIONAL ID of the merchant's acquirer.
merchantIdIssuer	string	OPTIONAL ID of the merchant's issuer.
merchantMessage	string	OPTIONAL Message for the merchant
monetToken	string	OPTIONAL
pan	string	OPTIONAL Primary account number.

par	string	OPTIONAL
panSequenceNumber	string	OPTIONAL Sequence number of the primary account number.
refungId	string	OPTIONAL ID of the refund transaction
responseCode	string	OPTIONAL Code of the response. TerminalErrorCodes
responseMessage	string	OPTIONAL Message of the response.
reversal	boolean	OPTIONAL Information if the transaction is a reversal or not.
sequenceNumber	string	OPTIONAL Number of the sequence.
surcharge	int	OPTIONAL Surcharged amount
terminalIdAcquirer	string	OPTIONAL ID of the acquirer's terminal.
terminalIdIssuer	string	OPTIONAL ID of the issuer's terminal.
tipAmount	integer	OPTIONAL Amount of the tip
transactionId	string	ID of the transaction.
transactionType	string	Transaction type. TransactionTypes
tokens	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": "TNEXGO01",
  "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": "M1TNEXGO01",
  "terminalIdIssuer": "M1TNEXGO01",
  "tokens": ["fkdfkjbekjrhljb"],
  "transactionId": "48a3ea7d-b82f-4b42-abbb-5043365e4faa",
  "transactionType": "SALE_ONLINE"
}
```

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/closing_batch"
ECR v5: URI = "app://switchiopay/api/main/v5/closing_batch"
ECRv6: URI = "app://switchiopay/api/main/v6/closure"

Bundle result key = "transaction_result"

Example request for ECR v1:
```
paya://closing_batch
```

Example request for ECR v4:
```
app://switchiopay/api/main/v4/closing_batch
```

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

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
daughterCompanyId	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"

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
login	string	Login of the terminal.
password	string	Password of the terminal.
host	string	IP address/hostname of the TMS server.
port	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"

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
UpToDate	No action needed.
UpdateRequired	Terminal configuration should be updated as soon as possible.
UpdateImmediately	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.
UpdateAfterClosing	Terminal configuration should be updated after a successful closing.
UpdateAtSpecificTime	Terminal configuration should be updated at the provided time. (hour: <integer>, minute: <integer>)
Undefined	Unknown data received from the server.
UndefinedInvalidData	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"

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"

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"

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
transactionId	string	ID of the transaction.
transactionType	string	Type of the transaction. In this case, it is HANDSHAKE.
responseCode	string	Code of the response. TerminalErrorCodes
responseMessage	string	Message of the response.

Result example:

```
{
"transactionId":"89555389-4f67-4cb0-aee8-544ea11dbd1c",
"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"

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": ["8798F7W98723KDGF08723HKDJHF897"],
}
```

Result example v4:

```
{
    "resultCode": 0,
    "maskedCardNumber": "970348******3910",
    "tokens": ["8798F7W98723KDGF08723HKDJHF897"],
    "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"

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": "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": ["fkdfkjbekjrhljb"]  //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"

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": ["fkdfkjbekjrhljb"]  //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
transactionId	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"

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
aid	string	OPTIONAL Application identifier.
appLabel	string	OPTIONAL Label of application.
authCode	string	OPTIONAL Code of the authorization.
cardHolderMessaage	string	OPTIONAL Message for cardholder.

cardInputType	string	OPTIONAL Type of the input. Enum is in the CardInputType
brand	string	OPTIONAL Card brand.
cvmTypeList	string array	OPTIONAL Type of the transaction. CvmType
dateTimeServer	string	OPTIONAL Date and time on the server when the result was made.
dateTimeTerminal	string	OPTIONAL Date and time on the terminal when the result was made.
expiration	string	OPTIONAL Expiration date of the card.
merchantIdAcquirer	string	OPTIONAL ID of the merchant's acquirer.
merchantIdIssuer	string	OPTIONAL ID of the merchant's issuer.
merchantMessaage	string	OPTIONAL Message for merchant.
pan	string	OPTIONAL Primary account number.
panSequenceNumber	string	OPTIONAL Sequence number of the primary account number.
responseCode	string	OPTIONAL Code of the response. TerminalErrorCodes
responseMessage	string	OPTIONAL Message of the response.
sequenceNumber	string	OPTIONAL Number of the sequence.
terminalIdAcquirer	string	OPTIONAL ID of the acquirer's terminal.
terminalIdIssuer	string	OPTIONAL ID of the issuer's terminal.
transactionId	string	ID of the transaction.
transactionType	string	Transaction type. TransactionTypes
cardToken	string	OPTIONAL Encrypted combination of data.
par	string	OPTIONAL Card PAR number
transactionReference	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":"1111222233334444",
		"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"

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
daughterCompanyId	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"

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": "TDEMOD01",
    "terminalIdIssuer": "terminalIdIssuer",
    "merchantIdAcquirer": "M1TDEMOD01",
    "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"
}
]
```

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
transactionId	string	ID of the transaction.
amount	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).
tipAmount	int	OPTIONAL Amount of the tip.
currencyCode	int	Code of the currency. ISO 4217 currency code
invoiceNumber	string	OPTIONAL Variable symbol, max length is 20 char
daughterCompanyId	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"

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
transactionId	string	ID of the transaction.
amount	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).
currencyCode	int	Code of the currency. ISO 4217 currency code
invoiceNumber	string	OPTIONAL Variable symbol, max length is 20 char
daughterCompanyId	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"

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. PREAUTHORIZATION

Performs a pre-authorization with a specified amount, currency and invoice number. To process a pre-authorization, it is required to tap or insert a payment card.

5.3.1. Request

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction.
amount	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).
currencyCode	int	Code of the currency. ISO 4217 currency code
invoiceNumber	string	OPTIONAL Variable symbol, max length is 20 char
daughterCompanyId	string	OPTIONAL ID of the daughter company. Required only when virtualization is enabled.

ECR v1: URI = "paya://preauth"
ECR v2: URI = "paya://preauth_v2"
ECR v3: URI = "paya://preauth_v3"
ECR v4: URI = "app://switchiopay/api/main/v4/preauth"
ECR v5: URI = "app://switchiopay/api/main/v5/preauth"
ECR v6: URI = "app://switchiopay/api/main/v6/preauth"

Parameter key = "data"
Bundle result key = "transaction_result"

Example request for ECR v1:
```
paya://preauth?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/preauth?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/preauth
- parameter key: data
- parameter value: request json

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
transactionId	string	ID of the transaction.
originalTransactionId	string	ID of the original transaction that is being incremented.
incrementAmount	int	Increment amount is the amount that the transaction is being incremented by.
invoiceNumber	string	OPTIONAL Variable symbol, max length is 20 char
daughterCompanyId	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"

Parameter key = "data"
Bundle result key = "transaction_result"

Example request for ECR v1:
```
paya://preauth_increment?data={"transactionId":
"89555389-4f67-4cb0-aee8-544ea11dbd1c", "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-aee8-544ea11dbd1c", "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
transactionId	string	ID of the transaction.
originalTransactionId	string	ID of the original transaction that is being incremented.
decrementAmount	int	Decrement amount is the amount that the transaction is being decremented by.
invoiceNumber	string	OPTIONAL Variable symbol, max length is 20 char
daughterCompanyId	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"

Bundle result key = "transaction_result"

Example request for ECR v5:

```
app://switchiopay/api/main/v5/preauth_decrement?data={"transactionId":
"89555389-4f67-4cb0-aee8-544ea11dbd1c", "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
transactionId	string	ID of the transaction.
originalTransactionId	string	ID of the original transaction that is being incremented.
finalAmount	int	Final amount is the sum of the incremental transaction and the original transaction.
invoiceNumber	string	OPTIONAL Variable symbol, max length is 20 char
daughterCompanyId	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"

Parameter key = "data"
Bundle result key = "transaction_result"

Example request for ECR v1:
```
paya://preauth_complete?data={"transactionId":
"89555389-4f67-4cb0-aee8-544ea11dbd1c", "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-aee8-544ea11dbd1c", "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
transactionId	string	ID of the transaction.
originalTransactionId	string	ID of the original transaction that is being incremented.
daughterCompanyId	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"

Parameter key = "data"
Bundle result key = "transaction_result"

Example request for ECR v1:
```
paya://preauth_cancel?data={"transactionId":
"89555389-4f67-4cb0-aee8-544ea11dbd1c", "originalTransactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

Example request for ECR v4:
```
app://switchiopay/api/main/v4/preauth_cancel?data={"transactionId":
"89555389-4f67-4cb0-aee8-544ea11dbd1c", "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
transactionId		string	ID of the transaction.
originalTransactionData		object	
	terminalId	string	ID of the terminal.
	authCode	string	Code of the authorization.
	seqNumber	string	Number of the sequence.
finalAmount		int	Final amount is the sum of the incremental transaction and the original transaction.
invoiceNumber		string	OPTIONAL Variable symbol, max length is 20 char

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"

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 CANCEL

Cancel a preauth transaction created on a different POS.

5.9.1. Request

PARAMETER		TYPE	DESCRIPTION
transactionId		string	ID of the transaction.
originalTransactionData			
	terminalId	string	ID of the terminal
	authCode	string	Code of the authorization.
	seqNumber	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"

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

5.9.2. Result

See result structure and result examples.

5.10. GET TRANSACTION

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

5.10.1. 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"

Parameter key = "data"
Bundle result key = "transaction_result"

Example request for ECR v1:
```
paya://get_transaction?data={"transactionId":
"89555389-4f67-4cb0-aee8-544ea11dbd1c"}
```

Example request for ECR v4:
```
app://switchiopay/api/light/v4/get_transaction?data={"transactionId":
"89555389-4f67-4cb0-aee8-544ea11dbd1c"}
```

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

5.10.2. Result

See result structure and result examples.

5.11. 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).

5.11.1. Request

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the reverse transaction.
originalTransactionId	string	ID of the transaction that is being reversed.
daughterCompanyId	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"

Parameter key = "data"
Bundle result key = "transaction_result"

Example request for ECR v1:
```
paya://reversal?data={"transactionId":
"89555389-4f67-4cb0-aee8-544ea11dbd1c", "originalTransactionId":
"88bb2366-716d-11ed-a1eb-0242ac120002"}
```

Example request for ECR v4:
```
app://switchiopay/api/main/v4/reversal?data={"transactionId":
"89555389-4f67-4cb0-aee8-544ea11dbd1c", "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

5.11.2. Result

See result structure and result examples.

5.12. MOTO

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

5.12.1. Request

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction.
amount	int	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).
currencyCode	int	ISO 4217 currency code.
invoiceNumber	string	OPTIONAL Number of the invoice.
daughterCompanyId	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"

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

5.12.2. Result

See result structure and result examples.

5.13. TRANSACTION STATUS

Returns status of selected transaction represented by transactionId.

5.13.1. 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"

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

5.13.2. Result

See result structure and result examples.

5.14. 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.

5.14.1. Request

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction.
amount	int	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).
currencyCode	int	ISO 4217 currency code.
refundId	string	ID of the refund.
invoiceNumber	string	OPTIONAL Number of the invoice.
daughterCompanyId	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"

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": "234563958345763453889", "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": "234563958345763453889", "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

5.14.2. Result

See result structure and result examples.

5.15. 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.

5.15.1. 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”

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

5.15.2. Result

See result structure and result examples.

5.16. QR PAYMENT

Initiates a QR payment transaction. Transaction is completed by QR code scanning - see SCANNER.

5.16.1. Request

PARAMETER	TYPE	DESCRIPTION
transactionId	string	ID of the transaction.
amount	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).
currencyCode	int	Code of the currency. ISO 4217 currency code
invoiceNumber	string	OPTIONAL Variable symbol, max length is 20 char
daughterCompanyId	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"}
```

5.16.2. 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"

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
transactionId	string	ID of the transaction.
transportAmount	int	OPTIONAL 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).
transportCurrency Code	int	OPTIONAL Currency code visualized on the terminal/validator display when the card is tapped.
InvoiceNumber	string	Variable symbol, max length is 20 char
transportData	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"

Parameter key = "data"
Bundle result key = "transaction_result"

Example request for ECR v1:
```
paya://payment_transport?data={"transactionId":"3ffed32e-6243-430e-905b-f0c55
52feeff", "invoiceNumber":"1234", "transportData":"data"}
```

Example request for ECR v4:
```
app://switchiopay/api/main/v4/payment_transport?data={"transactionId":"3ffed3
2e-6243-430e-905b-f0c5552feeff", "transportAmount":10053,
"transportCurrencyCode":203, "invoiceNumber":"1234", "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
processingResultCode	int	Operation result code
message	string	String describing cause of the failure
fareTransactionRequest	fareTransactionRequest	OPTIONAL fareTransactionRequest object
device	device	device object
type	string	EMV reader brand
reader_id	string	Card reader ID
reader_sn	string	Card reader serial number
deny_list_version	string	Current version of the deny list
transaction	transaction	transaction object
token	string	Encrypted combination of data
id	string	Transaction ID
dttm	string	Date and time of the transaction
result	string	Token deny list result
masked_pan	string	Masked PAN of the card
fare_data	string	base64 encoded string data from validator
emv_data	emv_data	emv_data object
tags	array	
encrypted	encrypted	encrypted object
encryption_method	string	Method used for encryption
encrypted_data	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",
       "dttm": "2023-10-12T12:28:53Z",
       "result": "NO_DENYLIST",
       "masked_pan": "541333******4111"
},
"fare_data": "eyJCdXMiOjF9",
 "emv_data": {
       "tags": [
              {
              "key": "9F24",
              "value": "3530303141424344454647484944A4B4C4D4E4F505152535455565758 59"
              },
              {
              "key": "9F06",
              "value": "A0000000041010"
              },
              {
              "key": "5F24",
              "value": "2512"
              }
              ],
"encrypted": {
       "encryption_method": "MNSP",
       "encrypted_data":
"00787F637B727209096D54AC173A2DE33A098C2001BB67DE20605FB79528E5AAFB12DA2FD535C08E40AA8CBF25
682226FA63228FD6A5F37A98BF26F5193405E748433336E09A132B07AC39D9911740A741C936CDF63201499BA30
E7A0F1098E2F9778D59B9949988259FB231CFE6DBD75B528FD0EA9A9E4EB8B76D2620E4F7AA4F6DCEF5C07188F5
30525559F2D62BCA08053A36B49DD22432131AF3275F4B30F6800744D94344BBDCB7ED8021E7FDC8FC655CE7AAD
E7D9F2F34226459DEA6B00FA3BC4D9E1ADAF851053BF94825257C11B79BC749EBD00A9674614C8392CEFE3FD910
8C91E7F87025147F979BAAB23B0B4D1D47A083D20A67BFFDC3D42B89648EED81B0625CAA90873063DD0EFC311F2
B03E1EAE447F796969C02C8C9A62BE5A8F53688F4F76FA3AC531BBB13EF5575F623969041870EF07E6F9164A4AC
440979CCDF1C1FEA76A434D344318A47AADFAB26B27FA358400FB4C7CE75D20359FF672BAF035C7645F87FE3519
FAD008914840E3465737AAD4DAAD1"
              }
       }
}
}
```

6.3. CUSTOMER TAP

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

6.3.1. 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"

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

6.3.2. Result

Result structure:

PARAMETER		TYPE	DESCRIPTION
processingResultCode		int	0 - Success 3 - Card read error 4 - Encryption error 5 - Timeout 8 - Internal error
message		string	Result message.
response		object	Response object with the result.
	result	string	Possible values: • OK • FAIL
	resultMessage	string	OPTIONAL String describing the cause of the failure.
	code	string	Response code from terminal; see TerminalErrorCodes table.

Result example:

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

6.4. UPLOAD TRANSPORT TRANSACTIONS

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

6.4.1. 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"

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

6.4.2. Result

Result structure:

PARAMETER	TYPE	DESCRIPTION
resultCode	int	0 - Success 1 - Upload error 2 - Configuration error 4 - No transaction found
message	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"
}
```

6.5. 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.

6.5.1. 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"

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

6.5.2. 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"
}
```

6.6. CUSTOMER TAP SERVICE STOP

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

6.6.1. 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"

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

6.6.2. 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.

6.7. GET TAPS

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

6.7.1. 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"

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

6.7.2. Result

Result structure:

PARAMETER	TYPE	DESCRIPTION
token	string	Card token
id	string	ID of the tap
dttm	string	Date and time of the tap
result	string	Possible values: • OK • FAIL • EXPIRED • DENYLISTED • NOTIFY_DENYLIST • NO_DENYLIST
detail_result	string	OPTIONAL
masked_pan	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"
```

```
    }
]
```

6.8. GET DENYLIST UPDATE

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

6.8.1. 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"

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

6.8.2. Result

Result structure:
```
<string>
```

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

7. APPLICATION DESCRIPTION

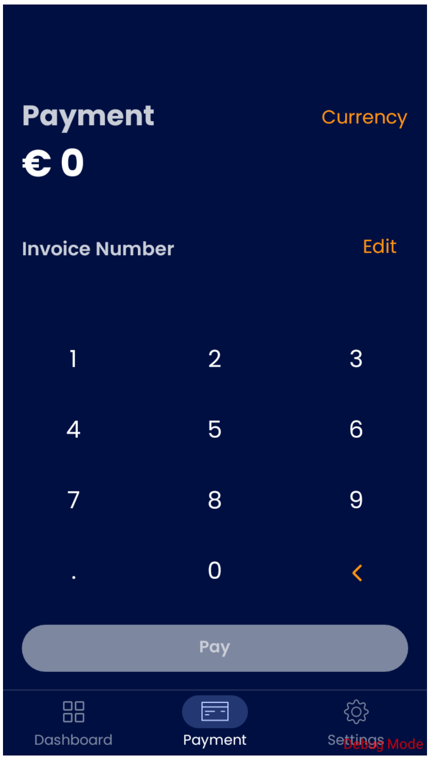
Certified app for Android terminal for EMV card payment. It supports two modes, standalone and external cash register mode.

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.

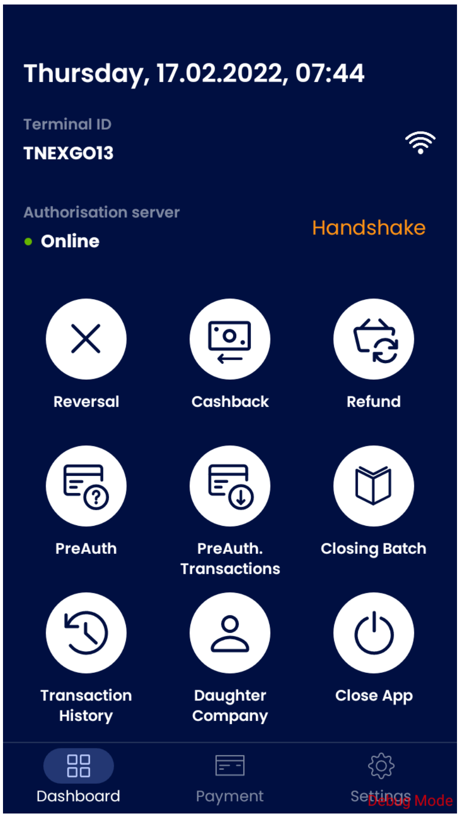
By filling in the necessary data and pressing the “Pay” button, you will be prompted to apply your credit card and the standard payment process will take place.

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

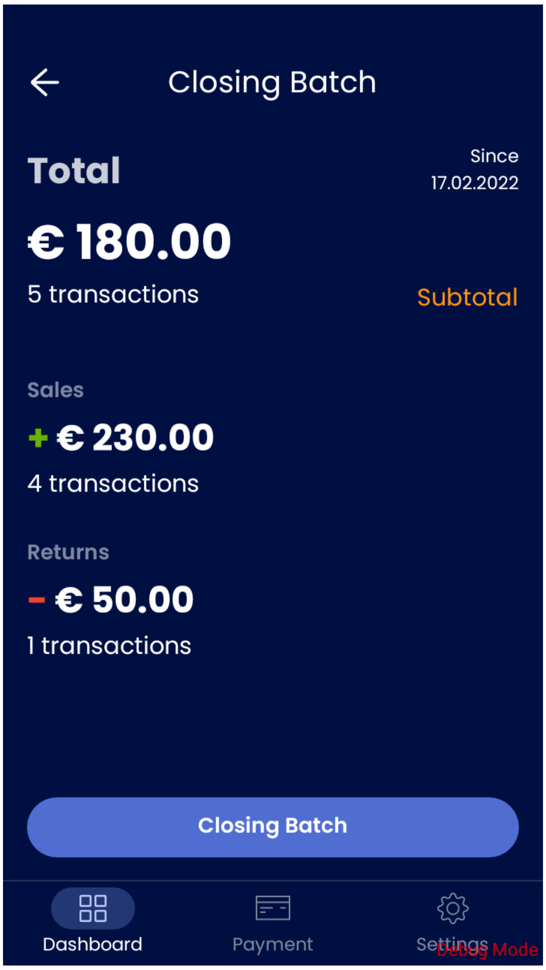
The operation flow of the refund transaction is basically the same as for standard payment.

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.

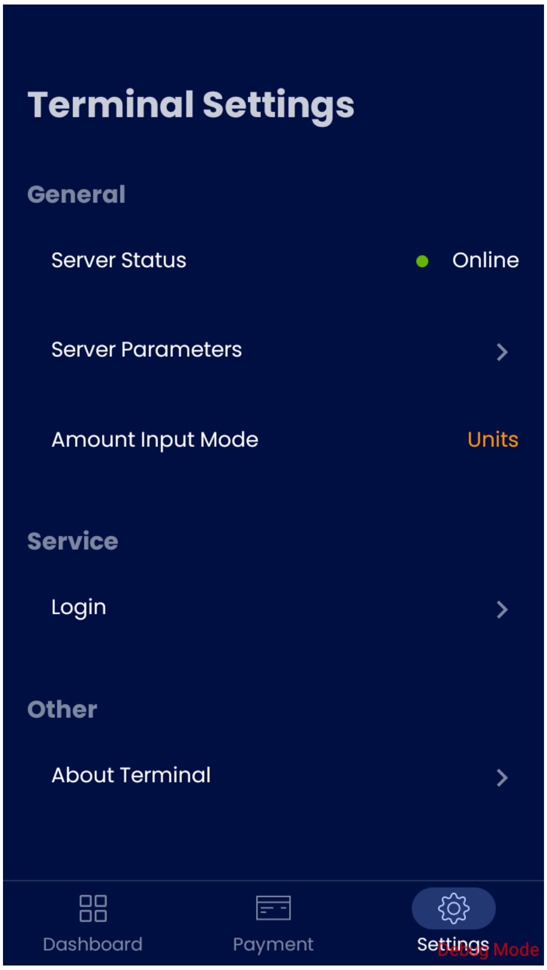
7.4. The closing batch



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



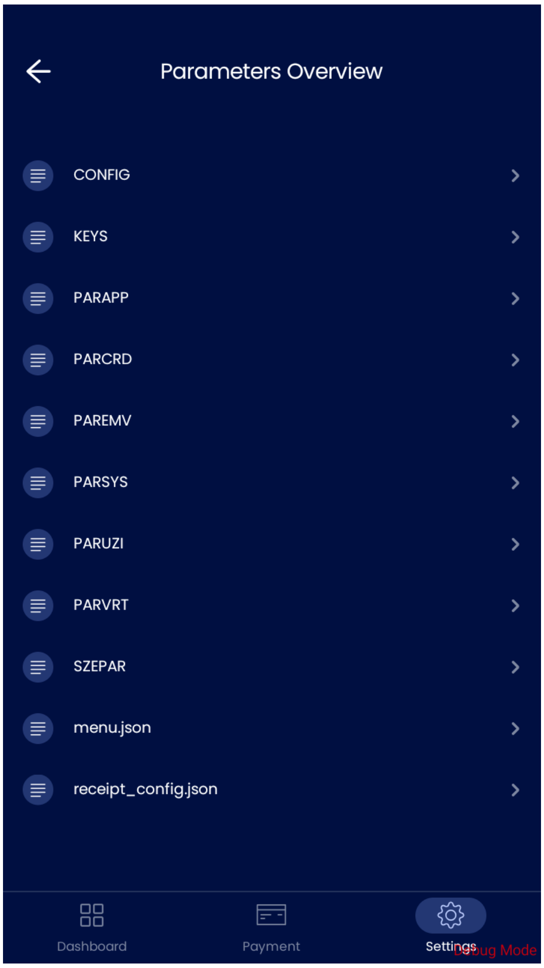
The terminal settings include tools for downloading parameters from the server, editing them, a service menu for testing all terminal functions without the need for an intermediate application protocol, loading keys to the terminal, and more.

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.

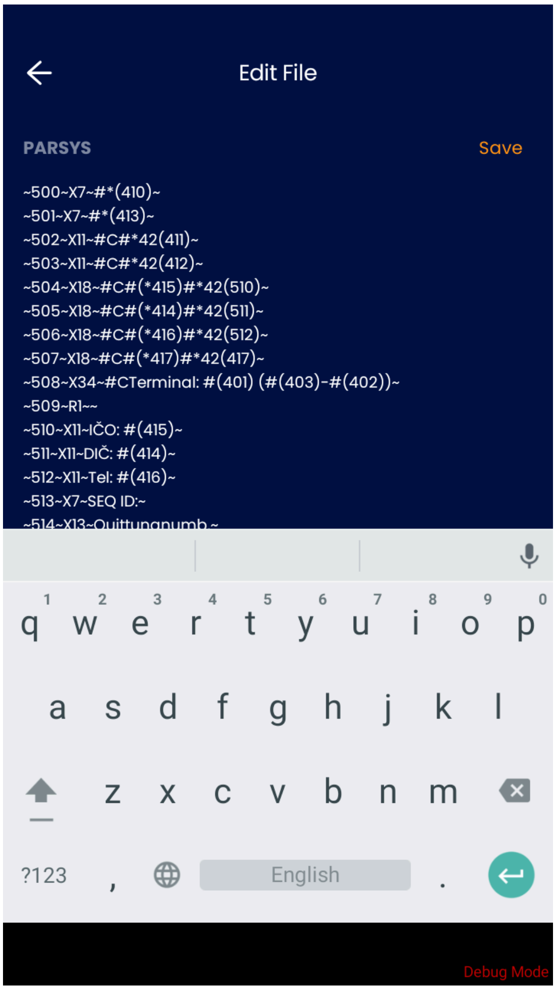
Tapping Server Status runs the handshake process resulting in one of Online, Offline or Unknown status.

7.6. Service Settings

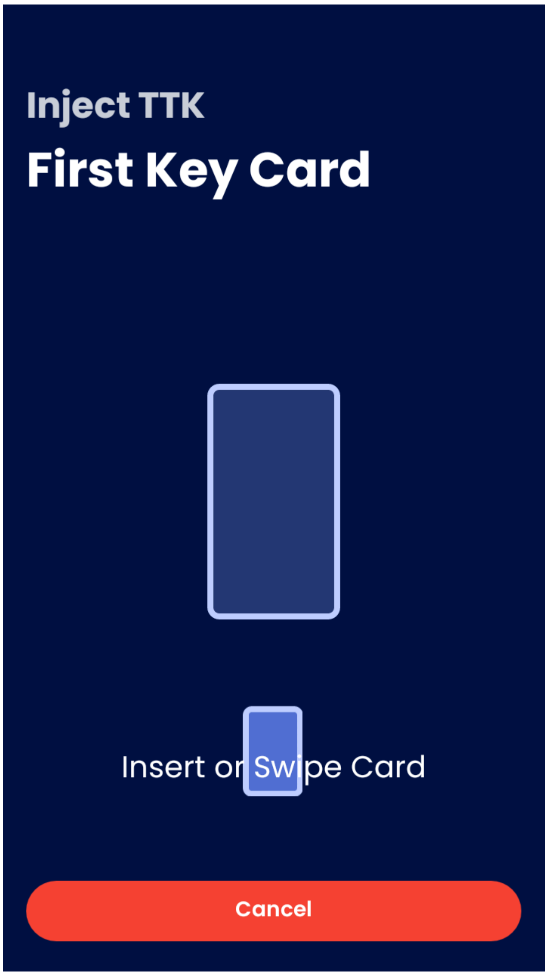


Service settings offer tools to setup or change configuration files by authorized technicians.

Parameters menu allows the technician to edit terminal parameters locally.


Edit File
PARSYS
Save
~500~X7~#*(410)~
~501~X7~#*(413)~
~502~X11~#C#*42(411)~
~503~X11~#C#*42(412)~
~504~X18~#C#(*415)#*42(510)~
~505~X18~#C#(*414)#*42(511)~
~506~X18~#C#(*416)#*42(512)~
~507~X18~#C#(*417)#*42(417)~
~508~X34~#CTerminal: #(401) (#(403)-#(402))~
~509~R1~~
~510~X11~IČO: #(415)~
~511~X11~DIČ: #(414)~
~512~X11~Tel: #(416)~
~513~X7~SEQ ID:~
English
?123
Debug Mode

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	X
2	KERNEL_NOT_ACCESS	Kernel isn't accessible	X
3	TEXT_KERNEL_NOT_INIT	Kernel not initialized	X
4	BLUETOOTH_NOT_SUPPORTED	Bluetooth not supported	X
5	PINBLOCK_TIMEOUT	Timeout for read pinblock	X
6	PINBLOCK_USER_CANCELLED	User cancel a pinblock	X
7	CANCELLED	Default user cancel	X
8	FAILED	Default failure	X
9	NO_PAPER	No paper available	X
10	UNKNOWN_MERCHANT	Unknown merchant	X
11	UNKNOWN	Transaction has not been completed	X
501	SCA_PIN_REQUIRED	Passcode and biometric capable. It is not possible to insert a contact card, it is just contactless.	X
1000	SPDH_DEFINE		-
1000	ApprovedBalancesAvailable	Approved balances available.	✔
1001	ApprovedNoBalancesAvailable	Approved no balances available.	✔
1002	ApprovedCountryClubStatus	Approved country club status.	✔
1003	Approved	Approved (maybe more identification is required).	✔
1004	ApprovedPendingIdentification	Approved pending identification (sign paper draft is required).	✔
1005	ApprovedBlind	Approved blind.	✔
1006	ApprovedVIPStatus	Approved VIP status.	✔
1007	ApprovedAdministrativeTransaction	Approved administrative transaction.	✔

1008	ApprovedNegativeCardFileHitOK	Approved negative card file hit OK.	✔
1009	ApprovedCommercialStatus	Approved commercial status.	✔
1010	ApprovedForPartialAmount	Approved for partial amount.	✔
1050	DeclinedGeneral	General decline.	X
1050	DeclinedProductGroup	Product group not allowed.	X
1050	DeclinedCardBlocked	The card is blocked.	X
1050	DeclinedCardBlockedByRange	The card is blocked by range.	X
1050	DeclinedAccountNotAllowedOnStation	The use of an account card is not allowed on the given gas station.	X
1050	DeclinedProductGroupForStation	Product group not allowed for the given gas station.	X
1050	DeclinedCardDailyTxCountLimitExceed	The card has exceeded the daily maximum number of transactions.	X
1050	DeclinedCardWeeklyTxCountLimitExceed	The card has exceeded the weekly maximum number of transactions.	X
1050	DeclinedCardMonthlyTxCountLimitExceed	The card has exceeded the monthly maximum number of transactions.	X
1050	DeclinedCardDailyFuelLimitExceed	The card has exceeded the daily fuel limit.	X
1050	DeclinedCardWeeklyFuelLimitExceed	The card has exceeded the weekly fuel limit.	X
1050	DeclinedCardMonthlyFuelLimitExceed	The card has exceeded the monthly fuel limit.	X
1050	DeclinedCardDailyMoneyLimitExceed	The card has exceeded the daily money limit.	X
1050	DeclinedCardWeeklyMoneyLimitExceed	The card has exceeded the weekly money limit.	X
1050	DeclinedCardMonthlyMoneyLimitExceed	The card has exceeded its monthly cash limit.	X
1050	DeclinedCardTxPriceLimitExceed	The card has exceeded the maximum cash limit per transaction.	X
1050	DeclinedCardDailyOtherLimitExceed	The card has exceeded the daily money limit on dry goods.	X
1050	DeclinedCardWeeklyOtherLimitExceed	The card has exceeded the weekly money limit on dry goods.	X
1050	DeclinedCardMonthlyOtherLimitExceed	The card has exceeded the monthly money limit on dry goods.	X
1050	DeclinedCardTxFuelAmountLimitExceed	The card has exceeded the maximum amount of fuel per	X

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

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

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

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

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

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

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

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

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

3275	MONEMV_CLS_KERNEL_STATUS_OFFLINE_DECLINED	Transaction is offline declined.	X
3276	MONEMV_CLS_KERNEL_STATUS_ONLINE_AUTHORISATION	Transaction succeeded and needs an online authorisation to be completed.	-
3277	MONEMV_CLS_KERNEL_STATUS_CANCELLED	Transaction has been cancelled.	X
3278	MONEMV_CLS_KERNEL_STATUS_USE_CONTACT_INTERFACE	The transaction has to be conducted over another interface (chip or swipe for example)	X
3279	MONEMV_CLS_KERNEL_STATUS_NOT_ALLOWED	This function cannot be called as no transaction is in process with this kernel or cancellation is not allowed according to the transaction progress.	X
3280	MONEMV_CLS_KERNEL_STATUS_CONTINUE	The transaction flow must continue.	-
3281	MONEMV_CLS_KERNEL_STATUS_SUSPEND	The transaction flow must be stopped.	X
3282	MONEMV_CLS_KERNEL_STATUS_INTERNAL_ERROR	A kernel internal error occurred.	X
3283	MONEMV_CLS_KERNEL_STATUS_LIB_INTERFACE_ERROR	An error occurred into the kernel interface (linked to the custom application).	X
3284	MONEMV_CLS_KERNEL_STATUS_EXPIRED_CERTIFICATE	The certificate used for Offline data authentication is expired.	X
3285	MONEMV_CLS_KERNEL_STATUS_REVOKED_CERTIFICATE	The certificate used for Offline data authentication is revoked.	X
3286	MONEMV_CLS_KERNEL_STATUS_CARD_UNKNOWN	The contactless card accessed isn't the same as at the beginning of the transaction. This error occurs for a new card presentation as Issuer Script Processing …")	X
3287	MONEMV_CLS_KERNEL_STATUS_MOBILE	A mobile phone GPO response has been detected. This error allows to restart a selection application cycle with new user message (for mobile phone) …")	X
3383	MONEMV_CLS_KERNEL_STATUS_UNKNOWN	Unknown	X
4000	FEITIAN_READER	Feitian reader	-
8005	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	✔

		terminal country code, this error occurs.	
21474 83647	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
auhorisationResponseCode	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

CvmType
SALE_ONLINE
SALE_OFFLINE
RETURN_ONLINE
RETURN_OFFLINE
MOTO_SALE
CLOSE_BATCH
SUBTOTAL_BATCH
HANDSHAKE
TOKEN
CASHBACK_ONLINE
PREAUTHORIZATION
PREAUTHORIZATION_INCREMENT
PREAUTHORIZATION_COMPLETION
REFERRAL_RETURN
TRANSACTION_STATUS
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