

# AADE 1155 JSON ERP protocol

## Version [↗](#)

12

## Overview [↗](#)

The EDPS JSON ECR communication protocol allows the control of an EFT/POS by an ECR application via JSON commands.

This is the adjusted for 1155 version, covering both connection types (provider or ecr with the erp).

The communication is with the TCP/IP protocol, with the pos acting as the server. The port is 8888.

## Special Notes [↗](#)

### Pairing [↗](#)

The POS will accept transactions with an ECRToken only if the pairing ecr-pos has been completed. This translates to the following request:

1. Pairing Request with the ecrId
  2. SessionKey Request with the encrypted (under the masterkey) session key and the kcv as described below.
- For testing purposes the ECR should be under the 979703476 AFM

### Voids [↗](#)

The POS will accept void transactions without signature or token.

### Ecr Tokens response to FHMAS [↗](#)

We added the *tokenResponse* field in transaction response. This is the transaction result as described in A1098. This was done for convenience.

### Split payments on preloaded transactions [↗](#)

This depends on the POS setup - the functionality is available with setting.

The user has an option "MEPIKO" on the menu of each preloaded transaction. The user will provide an amount (which is less or equal to the original amount) and execute the transaction.

The preloaded will appear in the list with reduced amount. All split transactions will have the same ecrRefNo.

There are changes in the Get Transaction Response message. Instead of the key "txn":{}, the key "txns":[] will be used. txns is a list of txn.

### Serial connection support [↗](#)

SCMD = 0x02 (start command)

ECMD = 0x03 (end command)

The messages then are SCMD {json request/response} ECMD

example ping request: 0x02{"msgType":"PingRequest"}0x03

## Preloaded transactions [↗](#)

After preloaded transactions have been executed successfully, a Sync request must be sent from the ERP to collect the results.

In case you need to confirm receipt of the results you may use the "waitAck" key in the Sync request. The POS will then expect an AckSync request before marking the transactions in the sync response as synchronized.

In case the "repeat"="T", for each Sync response an AckSync will be needed.

Other requests of interest for the preloaded transactions are: GetPreloadedStatusRequest and ClearPreloadedRequest

## Table Of Contents [↗](#)

- [Version](#)
- [Overview](#)
- [Special Notes](#)
  - [Pairing](#)
  - [Voids](#)
  - [Ecr Tokens response to FHMAS](#)
  - [Spit payments on preloaded transactions](#)
  - [Serial connection support](#)
  - [Preloaded transactions](#)
- [Table Of Contents](#)
- [1 Details](#)
  - [1.1 General](#)
    - [1.1.1 Command Overview](#)
    - [1.1.2 Validations](#)
  - [1.2 Message Breakdown](#)
    - [1.2.1 PairingRequest](#)
      - [Example](#)
      - [Request Breakdown](#)
    - [1.2.2 PairingResponse](#)
      - [Example](#)
      - [Response Breakdown](#)
    - [1.2.3 SessionKeyRequest](#)
      - [Example](#)
      - [Request Breakdown](#)
    - [1.2.4 SessionKeyResponse](#)
      - [Example](#)
      - [Response Breakdown](#)
    - [1.2.5 Ping](#)
      - [Example](#)
      - [Request Breakdown](#)
    - [1.2.6 PingResponse](#)
      - [Example](#)
      - [Response Breakdown](#)
    - [1.2.7 TransactionRequest](#)
      - [Example](#)
      - [Request Breakdown](#)
    - [1.2.8 TransactionResponse](#)
      - [Example](#)
      - [Response Breakdown](#)

- 1.2.9 AtomicLoyaltyRequest
- 1.2.10 AtomicLoyaltyResponse
- 1.2.11 CurrentBatchTotals Request
- 1.2.12 CurrentBatchTotals Response
  - Response Breakdown
- 1.2.13 BatchClose Request
  - Request Breakdown
- 1.2.14 BatchClose Response
  - Response Breakdown
  - Batch Entry Breakdown
- 1.2.15 Sync Request
- 1.2.16 Sync Response
- 1.2.17 Sync Acknowledgment Request
- 1.2.18 Sync Acknowledgment Response
  - Example
  - Response Breakdown
- 1.2.19 Get Transaction Request
- 1.2.20 Get Transaction Response
- 1.2.21 Get Preloaded Status Request
- 1.2.22 Get Preloaded Status Response
  - Example
  - Response Breakdown
- 1.2.23 Clear Preloaded Request
- 1.2.24 Clear Preloaded Response
  - Example
  - Response Breakdown
- 1.2.25 Get Pre-auth Status Request
- 1.2.26 Get Pre-auth Status Response
  - Example
  - Response Breakdown
- 1.2.27 Terminal Status Request
- 1.2.28 Terminal Status Response
  - Response Breakdown
- 1.2.29 Terminal Info Request
- 1.2.30 Terminal Info Response
  - Response Breakdown
- 1.2.31 Cancel Current Payment Request
- 1.2.32 Cancel Current Payment Response
  - Example
  - Response Breakdown
- 1.2.33 Reverse Last Transaction Request
- 1.2.34 Reverse Last Transaction Response
  - Example
  - Response Breakdown
- 1.2.35 Read Card Request
- 1.2.36 Read Card Response
  - Response Breakdown
- 1.3 Examples
  - 1.3.1 Sale TransactionRequest
    - Received request
    - Sent response
  - 1.3.2 Refund TransactionRequest

- [Received request](#)
- [Sent response](#)
- [1.3.3 Void TransactionRequest](#)
  - [Received request](#)
  - [Sent response](#)
- [1.3.4 Preauth TransactionRequest \(no signature or token\)](#)
  - [Received request](#)
  - [Sent response](#)
- [1.3.5 Postauth \(Auth Completion\) TransactionRequest](#)
  - [Received request](#)
  - [Sent response](#)
- [1.3.6 Loyalty Inquiry TransactionRequest](#)
  - [Received request](#)
  - [Sent response](#)
- [1.3.7 Batch Totals Response](#)
  - [Received request](#)
  - [Send response](#)
  - [Batch totals breakdown](#)
- [1.3.8 Batch Close Response](#)
  - [Received request](#)
  - [Send response](#)
  - [Batch entries breakdown](#)
- [Appendix](#)
  - [A1 Transaction Types](#)
  - [A2 Response Codes](#)
  - 
  - [A3 Loyalty Schemes](#)

# 1 Details [🔗](#)

## 1.1 General [🔗](#)

### 1.1.1 Command Overview [🔗](#)

Command Name	Description
<b>Connectivity Commands</b>	
PairingRequest	Get the ECR ID and request the masterkey from Esend. (Masterkey As described in 1098.)  Or get the ERP ID and the providerId
SessionKeyRequest	Get the encrypted - under the masterkey - sessionkey
PingRequest	Ping the EDPS Host
<b>Transaction Commands</b>	
TransactionRequest	Initiate a transaction. See appendix A1: Transaction Types
GetTransactionRequest	Get a specific transaction response
SyncRequest	Synchronization between the ERP and POS

AckSyncRequest	When the synchronization requires acknowledgement.
GetPreloadedStatusRequest	Informs on the status of a preloaded transaction
ClearPreloadedRequest	Deletes preloaded transactions
GetPreauthStatusRequest	Informs on the status of a preauth transaction
AtomicLoyaltyRequest	<i>Standalone Redemption etc. - RFU</i>
<b>Batch Commands</b>	
CurrentBatchTotals	Get current batch totals.
BatchClose	Close batch.
<b>Unattended Commands</b>	
TerminalStatusRequest	Get the terminal status
TerminalInfoRequest	Get the terminal info
CancelCurrentPaymentRequest	Cancel the current payment request
ReverseLastTransactionRequest	Reverse the last transaction
ReadCardRequest	Read customer's card

## 1.1.2 Validations [🔗](#)

Description	Data	Validation
Provider's signature input: datetime	input	now-datetime<24h
Provider's signature input: payable amount.  The payable amount on the input data (signed data)  should be equal to the amount in the message.	input, amount	payable_amount=amount
Provider's signature input: TID	input	tid = local tid
EcrRefNo	ecrRefNo	Validation parseInt(ecrRefNo)

## 1.2 Message Breakdown [🔗](#)

### 1.2.1 PairingRequest [🔗](#)

If the providerId is used, the POS will check if the provider's public key exists, and if not it will retrieve it from the AADE API.

Example [↗](#)

```
1 {
2   "msgType": "PairingRequest",
3   "ecrId": "ABC00112233",
4   "isProvider": "020"
5 }
```

Request Breakdown [↗](#)

	JSON Object	
Description	Key	Value
Message Type	msgType	PairingRequest
ECR ID	ecrId	Alphanumeric, length 11, if "isProvider" key exists this is the ERP ID
Provider	isProvider	Optional, 3 char providerId

1.2.2 PairingResponse [↗](#)

Example [↗](#)

```
1 { "tid": "99980631" }
```

Response Breakdown [↗](#)

	JSON Object	
Description	Key	Value
Configuration Version	tid	Numeric, 8 chars length

1.2.3 SessionKeyRequest [↗](#)

Example [↗](#)

```
1 {
2   "msgType": "SessionKeyRequest",
3   "encKey": "1ED9F7AE0B2509281BBC2DE38EF2A12B:CC5FFF"
4 }
```

Request Breakdown [↗](#)

	JSON Object	
Description	Key	Value
Message Type	msgType	PairingRequest
Encrypted Session Key	encKey	length 39, provided as described in 1098

### 1.2.4 SessionKeyResponse [🔗](#)

Example [🔗](#)

```
1 { "rc": "00" }
```

Response Breakdown [🔗](#)

JSON Object		
Description	Key	Value
Response code	rc	See Appendix A2 (response codes)

### 1.2.5 Ping [🔗](#)

Example [🔗](#)

```
1 { "msgType": "PingRequest" }
```

Request Breakdown [🔗](#)

JSON Object		
Description	Key	Value
Message Type	msgType	PingRequest

### 1.2.6 PingResponse [🔗](#)

Example [🔗](#)

```
1 { "rc": "00" }
```

Response Breakdown [🔗](#)

JSON Object		
Description	Key	Value
Response code	rc	See Appendix A2 (response codes)

### 1.2.7 TransactionRequest [🔗](#)

Example [🔗](#)

```
1 {
2   "msgType": "TransactionRequest",
3   "ecrRefNo": "1562",
4   "txnType": "1",
5   "amount": 12400,
6   "tipAmount": 0,
7   "installments": "01",
8   "cashier": "abc",
9   "lpr": "T",
10  "token": "A/S321/F0190:978:2/D20231025115530/RSSS77880021/H133/T192/M0/QF7556B09",
}
```

```

11 "providerId": "001",
12 "input": "7BED034B3DF5FE405ADD17C35C65B568598CEF2F;;20240205053825;12400;10000;2400;12400;12345678",
13 "signature":
    "MEYCIQDIuVbyCxdhMV3MhUINtdvtHK5mfIS4jx/ws26pEvzh1QIhAMvFvldE8RmpcFhd4gmS4s2FRzu4w6VIT07KcoVaA//2"
14 }
15

```

## Request Breakdown [🔗](#)

JSON Object			
Description	Key	Value	
Message Type	msgType	TransactionRequest	Mandatory
A reference number generated by the cash register.	ecrRefNo	Numeric, max length 10	Mandatory (unique). Validation parseInt(ecrRefNo)
Receipt Number	receipt	Numeric, max length 6	Conditional (used in Cancel, Void, Adjust, Postauth)
The transaction type. See Appendix: TransactionTypes	txnType	Alphanumeric, 1 char	Mandatory
The amount in cents	amount	Numeric, max length 12	Conditional (not used in balance inquiry). The amount is equal to the payable amount in case of provider signature. The amount is equal to the ecr token amount in case of ECR.
The tip amount in cents	tipAmount	Numeric, max length 12	Optional
Number of installments	installments	Numeric, 2 digit, eg. "02", "16"	Conditional (used in sale, refund, preauth)
Postdating (RFU)	postdating	Numeric, 2 digit, eg. "02", "22"	Reserved for future use
(RFU)	instPart	Numeric, 1 char	Reserved for future use
The cashier that operates the device	cashier	Alphanumeric, max length 8	Optional
The approval code returned from the bank.	authId	Alphanumeric, max length 6	Conditional (used in types Cancel, Void, Postauth)
Use printer	lpr	1 char, "T"/"F"	Optional (default value set by EDPS in the terminal configuration)
The ECR token provided by the electronic cash register	token	Alphanumeric	Conditional (under specific erp config) - Subject to changes
The MyAADE service provider ID, provides a unique signature for the transaction	providerId	Numeric, 3 digits	Conditional (if the erp is connected to a provider)



Provider's input to sign	input	Alphanumeric	Conditional (if the erp is connected to a provider)
Provider's signature	signature	Base64, 96 chars	Conditional (if the erp is connected to a provider)

### 1.2.8 TransactionResponse [🔗](#)

#### Example [🔗](#)

```

1  {
2    "eid": "E000000101",
3    "eciRefNo": "1234",
4    "txn": {
5      "receipt": "823",
6      "rc": "00",
7      "batchNo": "13",
8      "ts": "20240423142019",
9      "originalAmount": "12400",
10     "finalAmount": "10000",
11     "loyaltyAmount": "2400",
12     "tipAmount": "0",
13     "providerId": "001",
14     "input": "7BED034B3DF5FE405ADD17C35C65B568598CEF2F;;20240205053825;12400;10000;2400;12400;12345678",
15     "signature": "MEYCIQDIuVbyCxdhMV3MhUINtdvtHK5mfIS4jx/ws2GpEvzh1QIhAMvFv1dE8RmpcFhd4gmS4s2FRzu4w6VIT07KcoVaA//2"
16   },
17   "rrno": "109814570860",
18   "authid": "570860",
19   "bid": "017",
20   "transactionId": "017;109814570860;570860", // 1155 acquirer signature: acquirerId;rrno;authId
21   "installments": "01",
22   "ota": "00000000",
23   "cardNo": "411111*****1111",
24   "cardType": "VISA",
25   "cardHolder": "-",
26   "trm": "T@1",
27   "loy": {
28     "loyScheme": "G",
29     "loyRC": "00",
30     "awardedPoints": "1",
31     "consumedPoints": "0",
32     "balance": "1200",
33     "merchantPoints": "000000000000",
34     "masterMerchantPoints": "000000000001",
35     "adjustedAmount": "10000"
36   },
37   "emv": {
38     "aid": "A0000000031010",
39     "app": "VISA",
40     "tc": "ARQC: 3862006558F4218D"
41   }
42 }

```

## Response Breakdown [🔗](#)

### Response Data Object Description

	JSON Object	
Description	Key	Value
A reference number generated by the cash register	ecrRefNo	Numeric string, max length 10
Terminal Identifier	eid	Alphanumeric, 10 chars length
<b>TXN data</b>	<b>txn</b>	
Receipt Number	receipt	Alphanumeric, variable length
Response Code See Appendix A2 (response codes)	rc	Alphanumeric, 2 chars length
Batch Number	batchNo	Variable length ASCII numeric
Timestamp of the transaction Format 'YYYYMMDDHHmmss' eg 20240423142019	ts	Numeric string, 14 digits length
The original amount in cents	originalAmount	Numeric, max length 12, negative for voids, refunds
The final amount in cents	finalAmount	Numeric, max length 12, negative for voids, refunds
The loyalty amount in cents	loyaltyAmount	Numeric, max length 12
The tip amount in cents	tipAmount	Numeric, max length 12
The provider Id	providerId	As in request - Conditional
The provider input	input	As in request - Conditional
The provider signature	signature	As in request - Conditional
The token	token	As in request - Conditional
The token Response is for 'closing' the token on the FHMAS side.	tokenResponse	The transaction result message as described in A1098 - Conditional
Reference Number	rrno	Numeric, 12 digits length
Bank Authorization Code	authId	Alphanumeric, 6 chars length
Bank ID - ACQUIRER	bid	Alphanumeric, 3 chars length
Transcation Id / Acquirer Signature	transactionId	acquirerId;rrno;authId as discribed in 1155
Installments	installments	Numeric, 2 digits length
On Top Amount (RFU)	ota	Numeric, 8 digits length
Masked card number. eg 496099*****0614	cardNo	Alphanumeric, 16 chars length
Cardholder name from credit card.	cardHolder	Alphanumeric, variable length

Card type	cardType	Alphanumeric, variable length
Terminal Risk Management	trm	Alphanumeric, 3 chars length
Loyalty Data	loy	Loyalty Data Object (see table below)
<b>EMV Data</b>	<b>emv</b>	
AID	aid	Alphanumeric, variable length
Label	app	Alphanumeric, variable length
TC / AC	tc	Alphanumeric, variable length

#### Loyalty Data Object Description

Loyalty Data		
Description	Key	Value
Loyalty Scheme ID See Appendix A3	loyScheme	Alphanumeric, 1 char
Loyalty response codes See Appendix A2 (response codes)	loyRC	Alphanumeric, 2 chars length
Points Awarded	awardedPoints	Numeric string, variable length
Points Consumed	consumedPoints	Numeric string, variable length
Points Balance	balance	Numeric string, variable length
RFU	promoCode	RFU
Loyalty scheme specific value	merchantPoints	Numeric string, variable length
Loyalty scheme specific value	masterMerchantPoints	Numeric string, variable length
Adjusted Amount (Loyalty scheme specific)	adjustedAmount	Numeric string, variable length

- In case of **Rreloaded transactions (txnType=9)** the response is: { "rc":"00" }

#### 1.2.9 AtomicLoyaltyRequest [🔗](#)

Name	DataType	Description
LOY.Scheme	Alphanumeric, 1 char	<b>Loyalty Scheme ID</b>  Unknown: '?'  Alpha Bonus: 'B'  Eurobank Epistrofi: 'F' or 'E'  Go4More: 'G'  Piraeus Yellow: 'Y'
eTxnType	Alphanumeric, 1 char	See Appendix A1 (transaction types)

iECRRefNo	Numeric, max 10 digits	A reference number generated by the cash register.
lAmount	Numeric, max 12 digits	The amount in cents
iReceiptNo	Alphanumeric, 1 char	Receipt Number (used in void, loyalty void, adjust etc)
szCashier	Alphanumeric, 1 char	The cashier that operates the device
EID (for unattended terminals only)	10 bytes ASCII char	Terminal identification
bUseTermPrinter	01 byte ASCII 'T'/'F'	Use printer (T for true, F for false)

### 1.2.10 AtomicLoyaltyResponse [🔗](#)

Same as 1.2.8 Transaction Response

### 1.2.11 CurrentBatchTotals Request [🔗](#)

```
1 { "msgType":"CurrentBatchTotals" }
```

JSON Object		
Description	Key	Value
Message Type	MsgType	CurrentBatchTotals

### 1.2.12 CurrentBatchTotals Response [🔗](#)

```
1 {
2     "currentBatchNo":"001201",
3     "batchTotals":"001000000000600001000000000100001000000000080000000000000000"
4 }
```

### Response Breakdown [🔗](#)

JSON Object		
Description	Key	Value
Current Batch Number	currentBatchNo	Alphanumeric 10 chars length
Batch Totals	batchTotals	<p>Alphanumeric, 60 chars</p> <p>This is a composite field. Composed of the following subfields appended (no delimiter)</p> <ul style="list-style-type: none"> <li>• sales count : 03 byte ASCII numeric - left padded with zero</li> <li>• sales sum (in cents) : 12 byte ASCII numeric - left padded with zero</li> <li>• void sales count : 03 byte ASCII numeric - left padded with zero</li> <li>• void sales sum (in cents) : 12 byte ASCII numeric - left padded with zero</li> <li>• refunds count : 03 byte ASCII numeric - left padded with zero</li> </ul>

		<ul style="list-style-type: none"> <li>• refunds sum (in cents) : 12 byte ASCII numeric - left padded with zero</li> <li>• void refunds count : 03 byte ASCII numeric - left padded with zero</li> <li>• void refunds sum (in cents) : 12 byte ASCII numeric - left padded with zero</li> </ul>
Sales Count	salesCount	sales count : Numeric string (3 digits at most)
Sales Sum	salesTotalAmount	sales sum (in cents) : Numeric string (12 digits at most)
Void Sales Count	voidSalesCount	void sales count : Numeric string (3 digits at most)
Void Sales Sum	voidSalesTotalAmount	void sales sum (in cents) : Numeric string (12 digits at most)
Refunds Count	refundsCount	refunds count : Numeric string (3 digits at most)
Refunds Sum	refundsTotalAmount	refunds sum (in cents) : Numeric string (12 digits at most)
Void Refunds Count	voidRefundsCount	void refunds count : Numeric string (3 digits at most)
Void Refunds Sum	voidRefundsTotalAmount	void refunds sum (in cents) : Numeric string (12 digits at most)

### 1.2.13 BatchClose Request [🔗](#)

```
1 { "msgType":"BatchClose" }
```

#### Request Breakdown [🔗](#)

	JSON Object	
Description	Key	Value
Message Type	msgType	BatchClose
Use Printer	lpr	1 char  "T" to use printer  "F" to not use printer

### 1.2.14 BatchClose Response [🔗](#)

#### Response Breakdown [🔗](#)

	JSON Object	
Description	Key	Value
Current Batch Number	currentBatchNo	Numeric, 6 digits
Response Code	rc	Alphanumeric, 2 chars  See Appendix A2 Response Codes
Batch Approval	authId	Numeric, 6 digits
Batch Entries Count	batchCount	Numeric, 3 digits

		The number of transactions appended in this batch
Batch Entries	batchEntries	<p>Multiple entries, Variable length ASCII char</p> <p>A number of entries containing information about each transaction appended in this batch.</p> <p>See table below (Batch Entry)</p>
Terminal Batch Totals	termBatchTotals	<p>Alphanumeric, 60 chars</p> <p>see 1.4.12 CurrentBatchTotals Response for details</p>
Host Batch Totals	hostBatchTotals	<p>Alphanumeric, 60 chars</p> <p>see 1.4.12 CurrentBatchTotals Response for details</p>
Sales Count	salesCount	sales count : Numeric string (3 digits at most)
Sales Sum	salesTotalAmount	sales sum (in cents) : Numeric string (12 digits at most)
Void Sales Count	voidSalesCount	void sales count : Numeric string (3 digits at most)
Void Sales Sum	voidSalesTotalAmount	void sales sum (in cents) : Numeric string (12 digits at most)
Refunds Count	refundsCount	refunds count : Numeric string (3 digits at most)
Refunds Sum	refundsTotalAmount	refunds sum (in cents) : Numeric string (12 digits at most)
Void Refunds Count	voidRefundsCount	void refunds count : Numeric string (3 digits at most)
Void Refunds Sum	voidRefundsTotalAmount	void refunds sum (in cents) : Numeric string (12 digits at most)

#### Batch Entry Breakdown [↗](#)

Batch Entry (Content separated by SEP #)		
Name	DataType	Description
Receipt	Numeric, variable length	Receipt Number
Transaction Type	Alphanumeric, 1 char	See Appendix A1 (transaction types)
Amount	Numeric, variable length	The amount in cents

Tip	Numeric, variable length	The tip amount in cents
Installments	Numeric, variable length	Installments
Postdating	Numeric, variable length	Postdating (not in use), always 0
InstPart	Numeric, variable length	InstPart (not in use), always 0
OTA	Numeric, variable length	0
STAN	Numeric, variable length	System Trail Audit Number
Approval	Alphanumeric, 6 chars	Bank Authorization Code
BID	Alphanumeric, 3 chars	Bank ID
RRNO	Numeric, 12 chars	Reference Number
Cardno	Alphanumeric, 16 chars	Masked card number. eg 496099*****0614
Card Type	Alphanumeric, variable length	Card Type
TRM	Alphanumeric, 3 chars	Terminal Risk Management
Cashier	Alphanumeric, 4 chars	The cashier that operates the device

- The BatchClose Response if synchronization is needed will be TXN\_SYNC\_REQUIRED= "62"

### 1.2.15 Sync Request [🔗](#)

```
1 { "msgType":"SyncRequest", "waitAck":"T" }
```

#### Request Breakdown

	JSON Object		
Description	Key	Value	
Message Type	msgType	SyncRequest	Mandatory
Wait for acknowledgement	waitAck	T	Optional - if it is used an AckSyncRequest will be expected before clearing the data.

### 1.2.16 Sync Response [🔗](#)

```
1 { "rc":"00",
2   "repeat":"F",
3   "waitAck":"F",
4   "txns":[{
5     "eid":"E000000101",
6     "ecrRefNo":"1234",
7     "txn":{
8       "receipt":"823",
```

```

9      "rc": "00",
10     "batchNo": "13",
11     "ts": "142019",
12     "originalAmount": "12400",
13     "finalAmount": "10000",
14     "loyaltyAmount": "2400",
15     "tipAmount": "0",
16     "providerId": "001",
17     "input": "7BED034B3DF5FE405ADD17C35C65B568598CEF2F;;20240205053825;12400;10000;2400;12400;12345678",
18     "signature": "MEYCIQDIuVbyCxdhMV3MhUINtdvtHK5mfIS4jx/ws2GpEvzh1QIhAMvFvldE8RmpcFhd4gmS4s2FRzu4w6VIT07KcoVaA//2"
19   ,
20     "rrno": "109814570860",
21     "authId": "570860",
22     "bid": "017",
23     "transactionId": "017;109814570860;570860", // 1155 acquirer signature: acquirerId;rrno;authId
24     "installments": "01",
25     "ota": "00000000",
26     "cardNo": "411111*****1111",
27     "cardType": "VISA",
28     "cardHolder": "-",
29     "trm": "T@1",
30     "loy": {
31       "loyScheme": "6",
32       "loyRC": "00",
33       "awardedPoints": "1",
34       "consumedPoints": "0",
35       "balance": "1200",
36       "merchantPoints": "000000000000",
37       "masterMerchantPoints": "000000000001",
38       "adjustedAmount": "10"
39     },
40     "emv": {
41       "aid": "A0000000031010",
42       "app": "VISA",
43       "tc": "ARQC: 3862006558F4218D"
44     }
45   }
46 ] }

```

## Response breakdown

	JSON Object	
Description	Key	Value
Response Code	rc	Alphanumeric, 2 chars See Appendix A2 Response Codes
Repeat request flag	repeat	1 char "T" repeat the sync request "F" sync completed
Awaiting Acknowledgement	waitAck	1 chat "T" the results must be acknowledged



		"F" no ack needed
Transactions list	txns	List of transactions. There is a TransactionResponse(1.2.8) for each txn

### 1.2.17 Sync Acknowledgment Request [🔗](#)

```
1 { "msgType": "AckSyncRequest" }
```

#### Request Breakdown

	JSON Object	
Description	Key	Value
Message Type	msgType	AckSyncRequest

### 1.2.18 Sync Acknowledgment Response [🔗](#)

#### Example [🔗](#)

```
1 { "rc": "00" }
```

#### Response Breakdown [🔗](#)

	JSON Object	
Description	Key	Value
Response code	rc	See Appendix A2 (response codes)

### 1.2.19 Get Transaction Request [🔗](#)

```
1 { "msgType": "GetTxnRequest", "ecrRefNo": "1562" }
```

#### Request Breakdown

	JSON Object	
Description	Key	Value
Message Type	msgType	GetTxnRequest
Txn reference number	ecrRefNo	Numeric, max length 10

### 1.2.20 Get Transaction Response [🔗](#)

Same as 1.2.8 Transaction Response or TXN\_NOT\_FOUND.

\*\* In case of split payment on preloaded, all the transactions with the same ecrRefNo are returned (key= "txns":[]).

### 1.2.21 Get Preloaded Status Request [🔗](#)

```
1 {"msgType":"GetPreloadedStatusRequest", "ecrRefNo": "1562"}
```

#### Request Breakdown

	JSON Object		
Description	Key	Value	
Message Type	msgType	GetPreloadedStatusRequest	Mandatory
Txn reference number	ecrRefNo	Numeric, max length 10	Mandatory

### 1.2.22 Get Preloaded Status Response [🔗](#)

#### Example [🔗](#)

```
1 { "status":"OPEN" }
```

#### Response Breakdown [🔗](#)

JSON Object		
Description	Key	Value
Status	status	OPEN, OPEN SPLIT, DONE, VOID, NOT FOUND.

There are 5 statuses: OPEN, OPEN SPLIT, DONE, VOID, NOT FOUND

### 1.2.23 Clear Preloaded Request [🔗](#)

```
1 {"msgType":"ClearPreloadedRequest", "ecrRefNo": "1562"}
```

#### Request Breakdown

	JSON Object		
Description	Key	Value	
Message Type	msgType	ClearPreloadedRequest	Mandatory
Txn reference number	ecrRefNo	Numeric, max length 10	Optional - if missing or empty all preloaded transactions will be deleted

1.2.24 Clear Preloaded Response [↗](#)

Example [↗](#)

```
1 { "rc": "00" }
```

Response Breakdown [↗](#)

JSON Object		
Description	Key	Value
Response code	rc	See Appendix A2 (response codes)

1.2.25 Get Pre-auth Status Request [↗](#)

```
1 { "msgType": "GetPreAuthStatusRequest", "ecrRefNo": "1562" }
```

Request Breakdown

	JSON Object		
Description	Key	Value	
Message Type	msgType	GetPreloadedStatusRequest	Mandatory
Txn reference number	ecrRefNo	Numeric, max length 10	Mandatory

1.2.26 Get Pre-auth Status Response [↗](#)

Example [↗](#)

```
1 { "status": "ACTIVE", expires: "20250306235959" }
```

Response Breakdown [↗](#)

JSON Object		
Description	Key	Value
Status	status	ACTIVE, NOT FOUND.
Expiration date	expires	datetime or empty

1.2.27 Terminal Status Request [↗](#)

```
1 { "msgType": "TerminalStatusRequest" }
```

Request Breakdown

	JSON Object	
Description	Key	Value

Message Type	msgType	TerminalStatusRequest
--------------	---------	-----------------------

### 1.2.28 Terminal Status Response [🔗](#)

#### Response Breakdown [🔗](#)

JSON Object		
Description	Key	Value
Status	status	READY, BUSY, PROCESSING

### 1.2.29 Terminal Info Request [🔗](#)

```
1 { "msgType":"TerminalInfoRequest" }
```

#### Request Breakdown

JSON Object		
Description	Key	Value
Message Type	msgType	TerminalInfoRequest

### 1.2.30 Terminal Info Response [🔗](#)

#### Response Breakdown [🔗](#)

JSON Object		
Description	Key	Value
Serial Number	serialNumber	Numeric, length 10
Terminal ID	terminalId	Numeric, length 10
Merchant ID (EID)	merchantId	Alphanumeric, length 15
Merchant Name	merchantName	Alphanumeric, length 80
Merchant Address	merchantAddress	Alphanumeric, length 60
Merchant City	merchantCity	Alphanumeric, length 40
Merchant Post Code	merchantPostCode	Numeric, length 5
Application Version	appVersion	Alphanumeric, length 10
Firmware Version	firmVersion	Alphanumeric, length 60

### 1.2.31 Cancel Current Payment Request [🔗](#)

```
1 { "msgType":"CancelCurrentPaymentRequest" }
```

#### Request Breakdown

	JSON Object	
Description	Key	Value
Message Type	msgType	CancelCurrentPaymentRequest

### 1.2.32 Cancel Current Payment Response [🔗](#)

#### Example [🔗](#)

```
1 { "rc":"00" }
```

#### Response Breakdown [🔗](#)

JSON Object		
Description	Key	Value
Response Code	rc	See Appendix A2 (response codes)

### 1.2.33 Reverse Last Transaction Request [🔗](#)

```
1 { "msgType":"ReverseLastTransactionRequest" }
```

#### Request Breakdown

	JSON Object	
Description	Key	Value
Message Type	msgType	ReverseLastTransactionRequest

### 1.2.34 Reverse Last Transaction Response [🔗](#)

#### Example [🔗](#)

```
1 { "rc":"00" }
```

#### Response Breakdown [🔗](#)

JSON Object		
Description	Key	Value
Response Code	rc	See Appendix A2 (response codes)

### 1.2.35 Read Card Request [🔗](#)

```
1 { "msgType":"ReadCardRequest", "format":"MASKED" }
```

#### Request Breakdown

	JSON Object	
--	-------------	--

Description	Key	Value
Message Type	msgType	ReadCardRequest
Card number format option	format	MASKED, HASHED

## 1.2.36 Read Card Response [↗](#)

### Response Breakdown [↗](#)

	JSON Object	
Description	Key	Value
Card Type	cardType	VISA
PAN	pan	Masked PAN, or TOKEN depending on the “format” field value
Expiration date	exp	Date

## 1.3 Examples [↗](#)

### 1.3.1 Sale TransactionRequest [↗](#)

#### Received request [↗](#)

##### With ERP-ECR connection

```
{"msgType": "TransactionRequest", "ecrRefNo": "1562", "txnType": "1", "amount": 100, "tipAmount": 0, "installments": "01", "cashier": "abc", "lpr": "T", "token": "A/S321/F0190:978:2/D20231025115530/RSSS77880021/H133/T192/M0/QF7556B09"}
```

##### With Provider

```
{"msgType": "TransactionRequest", "ecrRefNo": "1562", "txnType": "1", "amount": 100, "tipAmount": 0, "installments": "01", "cashier": "abc", "lpr": "T", "providerId": "001", "input": "7BED034B3DF5FE405ADD17C35C65B568598CEF2F;;20240205053825;12400;10000;2400;12400;12345678", "signature": "MEYCIQDIuVbyCxdhMV3MhUINtdvtHK5mfIS4jx/ws2GpEvzh1QIhAMvFvldE8RmpcFhd4gmS4s2FRzu4wGVIT07KcoVaA//2"}
```

#### Sent response [↗](#)

```
{"eid": "E000000101", "ecrRefNo": "", "txn": {"receipt": "832", "rc": "00", "batchNo": "13", "ts": "101757", "rrno": "110200605965", "authId": "432974", "bid": "026", "ota": "00000000", "cardNo": "510099*****6005", "cardType": "MASTERCARD", "cardHolder": "-", "trm": "P-1"}, "emv": {"aid": "A0000000041010", "app": "MASTERCARD", "tc": "ARQC: D727F6F9DA9B9C37"}}
```

### 1.3.2 Refund TransactionRequest [↗](#)

#### Received request [↗](#)

##### With ERP-ECR connection

```
{"msgType": "TransactionRequest", "ecrRefNo": "1563", "txnType": "2", "amount": 100, "tipAmount": 0, "installments": "01", "cashier": "abc", "lpr": "T", "token": "Z/S321/F0190:978:2/D20231025115530/RSSS77880021/H133/T192/M0/QF7556B09"}
```

##### With Provider

```
{"msgType": "TransactionRequest","ecrRefNo": "1563","txnType": "2","amount": 100,"tipAmount": 0,"installments": "01","cashier": "abc","lpr": "T","providerId": "001","input": "7BED034B3DF5FE405ADD17C35C65B568598CEF2F;;20240205053825;12400;10000;2400;12400;12345678", "signature": "MEYCIQDIuVbyCxdhMV3MhUINtdvtHK5mfIS4jx/ws2GpEvzh1QIhAMvFvldE8RmpcFhd4gmS4s2FRzu4wGVIT07KcoVaA//2"}
```

#### Sent response [↗](#)

```
{"eid": "E000000101","ecrRefNo": "", "txn": {"receipt": "834","rc": "00","batchNo": "13","ts": "101904","rrno": "110200611967","authId": "370480","bid": "026","ota": "00000000","cardNo": "510099*****6005","cardType": "MASTERCARD","cardHolder": "-", "trm": "P-1"}, "emv": {"aid": "A0000000041010","app": "MASTERCARD","tc": "AAC: 146AEB9B7AE4D1C3"}}
```

### 1.3.3 Void TransactionRequest [↗](#)

#### Received request [↗](#)

With ERP-ECR connection

```
{"msgType": "TransactionRequest","ecrRefNo": "1564","txnType": "4","amount": "100","receipt": "835","authId": "311011","cashier": "abc","lpr": "T","token": "V/S321/F0190:978:2/D20231025115530/RSSS77880021/H133/T192/M0/QF7556B09"}
```

With Provider

```
{"msgType": "TransactionRequest","ecrRefNo": "1564","txnType": "4","amount": "100","receipt": "835","authId": "311011","cashier": "abc","lpr": "T","providerId": "001","input": "7BED034B3DF5FE405ADD17C35C65B568598CEF2F;;20240205053825;12400;10000;2400;12400;12345678", "signature": "MEYCIQDIuVbyCxdhMV3MhUINtdvtHK5mfIS4jx/ws2GpEvzh1QIhAMvFvldE8RmpcFhd4gmS4s2FRzu4wGVIT07KcoVaA//2"}
```

#### Sent response [↗](#)

```
{"eid": "E000000101","ecrRefNo": "", "txn": {"receipt": "835","rc": "00","batchNo": "13","ts": "103904","rrno": "110200604969","authId": "327039","bid": "026","ota": "00000000","cardNo": "411111*****1111","cardType": "VISA","cardHolder": "-", "trm": "T@1"}}
```

### 1.3.4 Preauth TransactionRequest (no signature or token) [↗](#)

#### Received request [↗](#)

```
{"msgType": "TransactionRequest","ecrRefNo": "1565","txnType": "6","amount": 100,"tipAmount": 0,"installments": "01","cashier": "abc","lpr": "T"}
```

#### Sent response [↗](#)

```
{"eid": "E000000101","ecrRefNo": "", "txn": {"receipt": "836","rc": "00","batchNo": "13","ts": "104015","rrno": "110200615970","authId": "846649","bid": "000","ota": "00000100","cardNo": "411111*****1111","cardType": "VISA","cardHolder": "-", "trm": "T@1"}}
```

### 1.3.5 Postauth (Auth Completion) TransactionRequest [↗](#)

#### Received request [↗](#)

With ERP-ECR connection

```
{"msgType": "TransactionRequest","ecrRefNo": "1566","txnType": "7","amount": 100,"receipt": "836", "authId": "846649", "cashier": "abc","lpr": "T","token": "P/S321/F0190:978:2/D20231025115530/RSSS77880021/H133/T192/M0/QF7556B09"}
```

With Provider

```
{"msgType": "TransactionRequest","ecrRefNo": "1566","txnType": "7","amount": 100,"receipt": "836", "authId": "846649",  
"cashier": "abc","lpr": "T","providerId": "001","input":  
"7BED034B3DF5FE405ADD17C35C65B568598CEF2F;;20240205053825;12400;10000;2400;12400;12345678", "signature":  
"MEYCIQDIuVbyCxdhMV3MhUINdvtHK5mfIS4jx/ws2GpEvzh1QIhAMvFvldE8RmpcFhd4gmS4s2FRzu4wGVIT07KcoVaA//2"}
```

#### Sent response [↗](#)

```
{"eid": "E000000101","ecrRefNo": "", "txn":  
{"receipt": "837", "rc": "00", "batchNo": "13", "ts": "104115", "rrno": "110200615970", "authId": "846649", "bid": "000", "ota": "00000000", "c  
ardNo": "411111*****1111", "cardType": "VISA", "cardHolder": "-", "trm": "T@5"}}
```

### 1.3.6 Loyalty Inquiry TransactionRequest [↗](#)

#### Received request [↗](#)

```
{"msgType": "TransactionRequest","txnType": "I","lpr": "T"}
```

#### Sent response [↗](#)

```
{"eid": "E000000101","ecrRefNo": "0", "loy":  
{"loyScheme": "?", "loyRC": "00", "awardedPoints": "0", "consumedPoints": "0", "balance": "0", "merchantPoints": "000000000000", "ma  
sterMerchantPoints": "000000000000", "adjustedAmount": "0"}}
```

### 1.3.7 Batch Totals Response [↗](#)

#### Received request [↗](#)

```
{ "msgType": "CurrentBatchTotals" }
```

#### Send response [↗](#)

```
{"currentBatchNo": "000013", "batchTotals": "0210000000020900010000000001000010000000001000000000000000000000"}
```

#### Batch totals breakdown [↗](#)

```
sales count: 021  
sales sum (in cents): 000000002090  
void sales count: 001  
void sales sum (in cents): 000000000100  
refunds count: 001  
refunds sum(in cents): 000000000100  
void refunds count: 000  
void refunds sum(in cents): 000000000000
```

### 1.3.8 Batch Close Response [↗](#)

#### Received request [↗](#)

```
{ "msgType": "BatchClose" }
```

#### Send response [↗](#)

```
{"currentBatchNo": "000014", "rc": "00", "authId": "352433", "batchCount": "004", "batchEntries": "838#1#100#0#1#0#0#0#1174#8058  
40#026#0412105930#510099*****6005#MASTERCARD#P-1#abc  
%%839#1#100#0#1#0#0#0#1175#452741#026#0412105943#510099*****6005#MASTERCARD#P-1#abc  
%%840#1#100#0#1#0#0#0#1176#115881#026#0412105951#510099*****6005#MASTERCARD#P-1#abc  
%%841#2#300#0#1#0#0#0#1177#630907#026#0412110027#510099*****6005#MASTERCARD#P-1#abc  
%%", "termBatchTotals": "00300000000003000000000000000000010000000003000000000000000000", "hostBatchTotals": "003  
000000000300000000000000000000001000000000300000000000000000"}
```



## Batch entries breakdown [↗](#)

receipt: 838  
txn type: 1  
amount: 100  
tip amount: 0  
installments: 1  
postdating: 0  
inst partner: 0  
on top amount: 0  
stan: 1174  
auth ID: 805840  
bid: 026  
timestamp: 0412105930  
pan: 510099\*\*\*\*\*6005  
card type: MASTERCARD  
trm: P-1  
cashier: abc

--

receipt: 839  
txn type: 1  
amount: 100  
tip amount: 0  
installments: 1  
postdating: 0  
inst partner: 0  
on top amount: 0  
stan: 1175  
auth ID: 452741  
bid: 026  
timestamp: 0412105943  
pan: 510099\*\*\*\*\*6005  
card type: MASTERCARD  
trm: P-1  
cashier: abc

--

receipt: 840  
txn type: 1  
amount: 100  
tip amount: 0  
installments: 1  
postdating: 0  
inst partner: 0  
on top amount: 0  
stan: 1176  
auth ID: 115881  
bid: 026  
timestamp: 0412105951  
pan: 510099\*\*\*\*\*6005  
card type: MASTERCARD

trm: P-1  
cashier: abc  
  
--  
  
receipt: 841  
txn type: 2  
amount: 300  
tip amount: 0  
installments: 1  
postdating: 0  
inst partner: 0  
on top amount: 0  
stan: 1177  
auth ID: 630907  
bid: 026  
timestamp: 0412110027  
pan: 510099\*\*\*\*\*6005  
card type: MASTERCARD  
trm: P-1  
cashier: abc

Appendix [↗](#)

A1 Transaction Types [↗](#)

Transaction Type	Value
Sale	1
Refund	2
Void	4
Preauth	6
Postauth	7
Offline Sale	8
Preloaded Sale	9
Adjust Sale	A
Loyalty Inquiry	I

A2 Response Codes [↗](#)

General rule of thumb: Only "00" means that a transaction is approved.

Response Code	Description
00	OK
05	GENERIC_ERROR
57	TXN_NOT_PERMITTED

58	TXN_VALIDATION_FAILED
59	TXN_SIGNED_AMOUNT_ERROR
60	TXN_SIGNATURE_EXISTS
61	TXN_ECRREFNO_INVALID
62	TXN_SYNC_REQUIRED
63	PAIRING_REQUIRED
64	TXN_TIP_EXCEEDS_AMOUNT
76	TXN_NOT_FOUND
C1	CARD_NOT_ENTERED
C2	CARD_DECLINED
C3	CARD_EXPIRED
C4	TXN_ABORTED
Z1	HOST_CONNECT_FAILED
Z2	HOST_CONNECT_TIMEOUT
Z3	HOST_REPLY_TIMEOUT
Z7	ICC_REJECTED_ARPC
Z9	BATCH_EMPTY
ZE	HOST_COMM_FAILED
P1	PRINTER_FAILURE
BZ	TERMINAL_BUSY
XX	GENERIC_ECR_ERROR

### A3 Loyalty Schemes [🔗](#)

Loyalty Sceme Id	Loyalty Scheme
Unknown	?
Alpha Bonus	B
Eurobank Epistrofi	F or E
NBG Go4More	G
Piraeus Yellow	Y

