

GP PAY – ECR API INTERFACE

1. INTRODUCTION

Welcome to the comprehensive documentation for GP PAY App's Intent-based API. GP PAY App is a payment application that empowers users with a diverse array of transactional capabilities. API, we use allows you to effortlessly incorporate GP PAY App's transaction functionality into your own Android application. By integrating this API, you can provide your users with a remarkably streamlined and convenient payment experience, enabling them to effortlessly initiate transactions through GP PAY directly from your app.

2. ANDROID INTENT

Android Intent communication plays a pivotal role in the realm of Android application development, serving as a powerful messaging system that facilitates seamless communication between various components within an application or even across different applications. At its core, an Intent serves as a versatile message object, capable of carrying diverse data types, ranging from text to images, while also triggering crucial actions like launching activities or initiating services.

In the context of GP PAY App, we are using integrated API which is widely used by Global Payments partners. These Intents contains vital transaction details, such as the transaction amount, transactionID, and other essential fields necessary for initializing transactions. By utilizing these Intents, other applications can effortlessly start an Activity within GP PAY App that is designed to handle the payment transaction, ensuring a smooth and secure payment process for users.

3. OPERATIONS

Code	Operation
CP	Payment
CC	Cancel last transaction
CR	Refund
CT	Totals
CS	Subtotals
CA	Pre-authorization
CF	Pre-authorization completion
CI	Incremental pre-authorization
CU	User cancellation (any transaction in the last 90 days)
CL	Handshake
CM	POS Download – function not available (only unattended app)
TG	Get Config – function not available (only unattended app)

3.1. CP – Payment

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
    "sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CP")
    json.put("Amount", "11.00")
    json.put("Control", "32") // It will ask for TIP on POS
    json.put("TransactionID", "0123456789") // use Random()
    json.put("InvNumber", "12345678") // optional field
    json.put("CurrencyCode", "978") // check iso standard for values
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {
    Log.d(TAG, "Exception: ${e.message}")
}
    
```

Request example:

```

Request: {
    "Operation": "CP",
    "Amount": "11.00",
    
```

```
"Control": 33,
"TransactionID": "066836220",
"InvNumber": "12345678",
"CurrencyCode": "978",
"Language": "en"
}
```

Response example:

```
Response: {
  "Result": "1",
  "TerminalID": "10000085",
  "RespMessage": "Approved",
  "CardBrand": "VI Business TEST",
  "BIN": "479608",
  "MaskedPAN": "*****6335",
  "PINIndicator": "N",
  "AID": "A0000000031010",
  "CurrencyCode": "978",
  "Signature": "N",
  "CustomerReceipt": " P2C, s. r. o.\n Prievozska 4\|C\n 831 04 Bratislava\n-----
-----\n145 18.12.2023 11:03:01 0040\n RECEIPT FOR CUSTOMER\n-----
-----\n PAYMENT\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: [L] *****6335\nVISA\nAID: A0000000031010\n\nPurchase
amount EUR 11.00\nTip amount EUR 1.00\nAmount EUR 12.00\n\n\nVisa
Contactless\nApproval code: 766318\nSequence number: 001145008\n\nRC:
001\n\n SUCCESSFUL TRANSACTION\n Approved\n-----\n
Thank you\n Keep the receipt for later\n checking\nVersion: P2C A01.01 (001)_gp\n",
  "HostRC": "001",
  "TransactionID": "43238842",
  "MerchantReceipt": " P2C, s. r. o.\n Prievozska 4\|C\n 831 04 Bratislava\n-----
-----\n145 18.12.2023 11:03:01 0040\n RECEIPT FOR MERCHANT\n-----
-----\n PAYMENT\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: [L] *****6335\nVISA\nAID: A0000000031010\n\nPurchase
amount EUR 11.00\nTip amount EUR 1.00\nAmount EUR 12.00\n\n\nVisa
Contactless\nApproval code: 766318\nSequence number: 001145008\n\nRC:
001\n\n SUCCESSFUL TRANSACTION\n Approved\n-----\n
Thank you\n Keep the receipt for later\n checking\nVersion: P2C A01.01 (001)_gp\n"
}
```

3.1.1. Request fields for CP:

Field	Description	Mandatory
Operation	CP for card payment	Yes
Amount	Amount of payment in format "00.00"	Yes
Control	Check section about control	No

InvNumber	Check section about invoice number	No
TransactionID	Number representing each transaction	Yes
CurrencyCode	Currency of transaction	No
Language	Language of receipt and payment screen	No

3.1.2. Response fields for CP:

Field	Description
Result	Result of payment, check section result for further information
TerminalID	Identifier of terminal
RespMessage	Response in language based on terminal configuration/request msg
CardBrand	Brand name of card inserted/tapped/swiped
BIN	First digits of card used for identifying card type
MaskedPAN	Masked primary account number
PINIndicator	If for payment Pin was inserted or not.
AID	Application identifier of card.
CurrencyCode	Currency in ISO format.
Signature	If for payment signature was necessary or not.
CustomerReceipt	Customer receipt in string format.
HostRC	Response from host in numeric format
TransactionID	ID of transaction based on request
MerchantReceipt	Merchant receipt in string format
SequenceNumber	Specific number used for identifying transaction with authorization
AuthCode	Authorization code provided from authorization
TransactionTime	Transaction timestamp in the format YYYYMMDDHHMMSS
HostTransID	Specific transaction ID from the authorization.
AmountAuthorized	Full amount authorized by host

3.2. CC – Cancellation of last transaction

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
"sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CC")
    json.put("HostTransID", "08116677 231218") previous HostTranID
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {
    Log.d(TAG, "Exception: ${e.message}")
}

```

Example request:

```

Request: {
  "Operation": "CC",
  "Control": "1",
  "TransactionID": "7322499718",
  "CurrencyCode": "978",
  "HostTransID": "08116677 231218",
  "Language": "en"
}

```

Example response:

```

Response: {
  "Result": "0",
  "AuthCode": "116677",
  "TerminalID": "10000085",
  "HostTransID": "08116677 231218",
  "RespMessage": "APPROVED",
  "CardBrand": "VI Business TEST",
  "BIN": "479608",
  "MaskedPAN": "*****6335",
  "PINIndicator": "Y",
  "AID": "A0000000031010",
  "CurrencyCode": "978",
}

```

```

"Signature": "N",
"SequenceNumber": "001145009",
"TransactionTime": "20231218123824",
"AmountAuthorized": "131.00",
"CustomerReceipt": "    P2C, s. r. o.\n    Prievozska 4\\|C\n    831 04 Bratislava\n-----
-----\n001 18.12.2023 12:38:20 0000\n RECEIPT FOR CUSTOMER\n-----
-----\n    PAYMENT\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: [L] *****6335\nVISA\nAID: A0000000031010\n\nAmount EUR
131.00\n\nPIN OK\n\nVisa Contactless\nApproval code: 116677\nSequence number:
001145009\n\nRC:          000\n\n    APPROVED\n-----\n
Thank you\n Keep the receipt for later\n    checking\nVersion: P2C A01.01 (001)_gp\n",
"HostRC": "000",
"TransactionID": "7920167789",
"MerchantReceipt": "    P2C, s. r. o.\n    Prievozska 4\\|C\n    831 04 Bratislava\n-----
-----\n001 18.12.2023 12:38:20 0000\n RECEIPT FOR MERCHANT\n-----
-----\n    PAYMENT\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: [L] *****6335\nVISA\nAID: A0000000031010\n\nAmount EUR
131.00\n\nPIN OK\n\nVisa Contactless\nApproval code: 116677\nSequence number:
001145009\n\nRC:          000\n\n    APPROVED\n-----\n
Thank you\n Keep the receipt for later\n    checking\nVersion: P2C A01.01 (001)_gp\n"
}

```

3.2.1. Request fields for CC:

Field	Description	Mandatory
Operation	CC for cancellation	Yes
HostTransID	Host transaction identifier of previous transaction	Yes
Control	Check section about control	No
InvNumber	Check section about invoice number	No
TransactionID	Number representing each transaction	Yes
CurrencyCode	Currency of transaction	No
Language	Language of receipt and payment screen	No

3.2.2. Response fields for CC:

Field	Description
Result	Result of payment, check section result for further information
AuthCode	Authorization code provided from authorization
TerminalID	Terminal identifier
HostTransID	Specific transaction ID from the authorization.
RespMessage	Response in language based on terminal configuration/request msg
CardBrand	Brand name of card inserted/tapped/swiped
BIN	First digits of card used for identifying card type
MaskedPAN	Masked primary account number

<i>PINIndicator</i>	If for payment Pin was inserted or not.
<i>AID</i>	Application identifier of card.
<i>CurrencyCode</i>	Currency in ISO format.
<i>Signature</i>	If for payment signature was necessary or not.
<i>SequenceNumber</i>	Specific number used for identifying transaction with authorization
<i>TransactionTime</i>	Transaction timestamp in the format YYYYMMDDHHMMSS
<i>AmountAuthorized</i>	Full amount authorized by host
<i>CustomerReceipt</i>	Customer receipt in string format
<i>HostRC</i>	Response from host in numeric format
<i>TransactionID</i>	ID of transaction based on request
<i>MerchantReceipt</i>	Merchant receipt in string format

3.3. CR – Refund

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
    "sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CR")
    json.put("Amount", "150.00") amount to be refunded
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {
    Log.d(TAG, "Exception: ${e.message}")
}
    
```

Example request:

```

Request: {
    "Operation": "CR",
    "Amount": "131.00",
    "Control": 1,
    "TransactionID": "949561961",
    "CurrencyCode": "978",
    "Language": "en"
}
    
```

Example response:

```
Response: {
  "Result": "0",
  "AuthCode": "069272",
  "TerminalID": "10000085",
  "HostTransID": "08069272 231218",
  "RespMessage": "APPROVED",
  "CardBrand": "VI Business TEST",
  "BIN": "479608",
  "MaskedPAN": "*****6335",
  "PINindicator": "N",
  "AID": "A0000000031010",
  "CurrencyCode": "978",
  "Signature": "N",
  "SequenceNumber": "999145002",
  "TransactionTime": "20231218124656",
  "AmountAuthorized": "131.00",
  "CustomerReceipt": "    P2C, s. r. o.\n    Prievozska 4\|/C\n    831 04 Bratislava\n-----
-----\n145 18.12.2023 12:46:53 0001\n RECEIPT FOR CUSTOMER\n-----
-----\n    Refund\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: [L] *****6335\nVISA\nAID: A0000000031010\n\nAmount EUR
131.00\n\n\nVisa Contactless\nApproval code: 069272\nSequence number:
999145002\n\nRC:          000\n\n    APPROVED\n-----\n
Thank you\n Keep the receipt for later\n    checking\nVersion: P2C A01.01 (001)_gp\n",
  "HostRC": "000",
  "TransactionID": "949561961",
  "MerchantReceipt": "    P2C, s. r. o.\n    Prievozska 4\|/C\n    831 04 Bratislava\n-----
-----\n145 18.12.2023 12:46:53 0001\n RECEIPT FOR MERCHANT\n-----
-----\n    Refund\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: [L] *****6335\nVISA\nAID: A0000000031010\n\nAmount EUR
131.00\n\nMerchant sign: ..... \n\nVisa Contactless\nApproval code: 069272\nSequence
number: 999145002\n\nRC:          000\n\n    APPROVED\n-----\n
Thank you\n Keep the receipt for later\n    checking\nVersion: P2C A01.01 (001)_gp\n"
}
```

3.3.1. Request fields for CR:

Field	Description	Mandatory
Operation	CR for refund	Yes
Amount	Amount to refund on customer account	Yes
Control	Check section about control	No
TransactionID	Number representing each transaction	Yes
CurrencyCode	Currency of transaction	No
Language	Language of receipt and payment screen	No

3.3.2. Response fields for CR:

Field	Description
<i>Result</i>	Result of payment, check section result for further information
<i>AuthCode</i>	Authorization code provided from authorization
<i>TerminalID</i>	Terminal identifier
<i>HostTransID</i>	Specific transaction ID from the authorization.
<i>RespMessage</i>	Response in language based on terminal configuration/request msg
<i>CardBrand</i>	Brand name of card inserted/tapped/swiped
<i>BIN</i>	First digits of card used for identifying card type
<i>MaskedPAN</i>	Masked primary account number
<i>PINIndicator</i>	If for payment Pin was inserted or not.
<i>AID</i>	Application identifier of card.
<i>CurrencyCode</i>	Currency in ISO format.
<i>Signature</i>	If for payment signature was necessary or not.
<i>SequenceNumber</i>	Specific number used for identifying transaction with authorization
<i>TransactionTime</i>	Transaction timestamp in the format YYYYMMDDHHMMSS
<i>AmountAuthorized</i>	Full amount authorized by host
<i>CustomerReceipt</i>	Customer receipt in string format
<i>HostRC</i>	Response from host in numeric format
<i>TransactionID</i>	ID of transaction based on request
<i>MerchantReceipt</i>	Merchant receipt in string format

3.4. CT – Totals

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
"sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CT")
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {
    Log.d(TAG, "Exception: ${e.message}")
}

```

Example request:

```

Request: {
  "Operation": "CT",
  "Control": 1,
  "TransactionID": "5885453884",
  "Language": "en"
}

```

Example response:

```

Response:{
  "Result": "0",
  "TransactionID": "5885453884",
  "hostCounters": [
    {
      "RecordId": "PWZ_HOST",
      "DebitCount": "1",
      "DebitAmount": "13100",
      "CreditCount": "1",
      "CreditAmount": "13100"
    }
  ],
  "terminalCounters": [

```

```
{
  "RecordId": "VISA",
  "DebitCount": "1",
  "DebitAmount": "13100",
  "CreditCount": "1",
  "CreditAmount": "13100"
},
{
  "RecordId": "PWZ_TERMINAL",
  "DebitCount": "1",
  "DebitAmount": "13100",
  "CreditCount": "1",
  "CreditAmount": "13100"
}
],
"MerchantReceipt": " P2C, s. r. o.\n Prievozska 4\|V\C\n 831 04 Bratislava\n-----
\n145 18.12.2023 12:47:40 0002\n RECEIPT FOR MERCHANT\n-----
\n Close day\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\nBatch number: 145\nFrom:\nTo : 18.12.2023 12:47:40\n\n Totals
OK\nTerminal totals are the same\n as host totals.\n\n\n PAYMENT CARDS\n\n-----
OVERALL TOTALS-----\n COUNT AMOUNT CURR\n-----\n 2
0.00 EUR\n-----Short List-----\n-----VISA-----\nEUR COUNT
AMOUNT\nDebit 1 131.00\nCredit 1 131.00\n-----\nTotal:
2 0.00\n\nRC: 007\n\n APPROVED\n-----\n
Thank you\n Keep the receipt for later\n checking\nVersion: P2C A01.01 (001)_gp\n"
}
```

3.4.1. Request fields for CT:

Field	Description	Mandatory
Operation	CT for totals	Yes
Control	Check section about control	No
TransactionID	Number representing each transaction	Yes
Language	Language of receipt and payment screen	No

3.4.2. Response fields for CT:

Field	Description
Result	Result of payment, check section result for further information
TransactionID	ID of transaction based on request
hostCounters	Counter of transactions count and amount on authorization side
terminalCounters	Counter of transactions count and amount on terminal side
MerchantReceipt	Merchant receipt in string format

3.5. CS – Subtotals

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.posterminal",
"sk.co.kompakts.posterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CS")
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {
    Log.d(TAG, "Exception: ${e.message}")
}

```

Example request:

```

Request: {
  "Operation": "CS",
  "Control": 1,
  "TransactionID": "6488528083",
  "CurrencyCode": "978",
  "Language": "en"
}

```

Example response:

```

Response: {
  "Result": "0",
  "TransactionID": "6488528083",
  "hostCounters": [
    {
      "RecordId": "PWZ_HOST",
      "DebitCount": "1",
      "DebitAmount": "13100",
      "CreditCount": "0",
      "CreditAmount": "0"
    }
  ]
}

```

```

],
"terminalCounters": [
  {
    "RecordId": "VISA",
    "DebitCount": "1",
    "DebitAmount": "13100",
    "CreditCount": "0",
    "CreditAmount": "0"
  },
  {
    "RecordId": "PWZ_TERMINAL",
    "DebitCount": "1",
    "DebitAmount": "13100",
    "CreditCount": "0",
    "CreditAmount": "0"
  }
],
"MerchantReceipt": "  P2C, s. r. o.\n  Prievozska 4\|/C\n  831 04 Bratislava\n-----
\n146 18.12.2023 12:49:14 0004\n  RECEIPT FOR MERCHANT\n-----
\n  Subtotals\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\nBatch number: 146\nFrom: 18.12.2023 12:47:42\nTo : 18.12.2023 12:49:14\n\n
Totals OK\nTerminal totals are the same\n  as host totals.\n\n\n  PAYMENT
CARDS\n\n-----OVERALL TOTALS-----\n COUNT      AMOUNT CURR\n-----
----\n  1      131.00 EUR\n-----Short List-----\n\n-----VISA-----\nEUR
COUNT AMOUNT\nDebit      1  131.00\nCredit      0  0.00\n-----
\nTotal:      1  131.00\n\nRC:          007\n\n  APPROVED\n-----
--\n  Thank you\n  Keep the receipt for later\n  checking\nVersion: P2C A01.01
(001)_gp\n"
}

```

3.5.1. Request fields for CS:

Field	Description	Mandatory
Operation	CS for Subtotals	Yes
Control	Check section about control	No
TransactionID	Number representing each transaction	Yes
Language	Language of receipt and payment screen	No

3.5.2. Response fields for CS:

Field	Description
Result	Result of payment, check section result for further information
TransactionID	ID of transaction based on request
hostCounters	Counter of transactions count and amount on authorization side

<i>terminalCounters</i>	Counter of transactions count and amount on terminal side
<i>MerchantReceipt</i>	Merchant receipt in string format

3.6. CA – Preauthorization

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
    "sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CA")
    json.put("Amount", "11.00") preauthorization amount
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {
    Log.d(TAG, "Exception: ${e.message}")
}
    
```

Example request:

```

Request: {
    "Operation": "CA",
    "Amount": "31.00",
    "Control": 1,
    "TransactionID": "93777446",
    "CurrencyCode": "978",
    "HostTransID": "08956323 231218",
    "Language": "en"
}
    
```

Example response:

```

Response: {
    "Result": "0",
    "AuthCode": "147339",
    "TerminalID": "10000085",
    "HostTransID": "08147339 231218",
    "RespMessage": "APPROVED",
}
    
```



```

"CardBrand": "VI Business TEST",
"BIN": "479608",
"MaskedPAN": "*****6335",
"PINindicator": "N",
"AID": "A0000000031010",
"CurrencyCode": "978",
"Signature": "N",
"SequenceNumber": "001146003",
"TransactionTime": "20231218125043",
"AmountAuthorized": "31.00",
"CustomerReceipt": "    P2C, s. r. o.\n  Prievozska 4\|C\n  831 04 Bratislava\n-----
-----\n146 18.12.2023 12:50:40 0005\n  RECEIPT FOR CUSTOMER\n-----
-----\n  Preauthorisation\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: [L] *****6335\nVISA\nAID: A0000000031010\n\nAmount EUR
31.00\n\n\nVisa Contactless\nApproval code: 147339\nSequence number:
001146003\n\nRC:          000\n\n  APPROVED\n-----\n
Thank you\n Keep the receipt for later\n  checking\nVersion: P2C A01.01 (001)_gp\n",
"HostRC": "000",
"TransactionID": "93777446",
"MerchantReceipt": "    P2C, s. r. o.\n  Prievozska 4\|C\n  831 04 Bratislava\n-----
-----\n146 18.12.2023 12:50:40 0005\n  RECEIPT FOR MERCHANT\n-----
-----\n  Preauthorisation\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: [L] *****6335\nVISA\nAID: A0000000031010\n\nAmount EUR
31.00\n\n\nVisa Contactless\nApproval code: 147339\nSequence number:
001146003\n\nRC:          000\n\n  APPROVED\n-----\n
Thank you\n Keep the receipt for later\n  checking\nVersion: P2C A01.01 (001)_gp\n"
}
    
```

3.6.1. Request fields for CA:

Field	Description	Mandatory
Operation	CA for preauthorization	Yes
HostTransID	Host transaction identifier of previous transaction	Yes
Control	Check section about control	No
InvNumber	Check section about invoice number	No
TransactionID	Number representing each transaction	Yes
CurrencyCode	Currency of transaction	No
Language	Language of receipt and payment screen	No

3.6.2. Response fields for CA:

Field	Description
<i>Result</i>	Result of payment, check section result for further information
<i>AuthCode</i>	Authorization code provided from authorization
<i>TerminalID</i>	Terminal identifier
<i>HostTransID</i>	Specific transaction ID from the authorization.
<i>RespMessage</i>	Response in language based on terminal configuration/request msg
<i>CardBrand</i>	Brand name of card inserted/tapped/swiped
<i>BIN</i>	First digits of card used for identifying card type
<i>MaskedPAN</i>	Masked primary account number
<i>PINIndicator</i>	If for payment Pin was inserted or not.
<i>AID</i>	Application identifier of card.
<i>CurrencyCode</i>	Currency in ISO format.
<i>Signature</i>	If for payment signature was necessary or not.
<i>SequenceNumber</i>	Specific number used for identifying transaction with authorization
<i>TransactionTime</i>	Transaction timestamp in the format YYYYMMDDHHMMSS
<i>AmountAuthorized</i>	Full amount authorized by host
<i>CustomerReceipt</i>	Customer receipt in string format
<i>HostRC</i>	Response from host in numeric format
<i>TransactionID</i>	ID of transaction based on request
<i>MerchantReceipt</i>	Merchant receipt in string format

3.7. CI – Incremental Preauthorization

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
"sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CI")
    json.put("HostTransID", "08147339 231218")
    json.put("TerminalID", "10000085")
    json.put("Last4PANnumber", "1234")
    json.put("SequenceNumber", "000000000")
    json.put("OriginalDate", "231223")
    json.put("Amount", "11.00")
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {

```

Example request:

```

Request: {
    "Operation": "CI",
    "Amount": "31.00",
    "Control": 1,
    "TransactionID": "4711699691",
    "CurrencyCode": "978",
    "HostTransID": "08147339 231218",
    "TerminalID": "10000085",
    "Last4PANnumber": "6335",
    "SequenceNumber": "001146003",
    "OriginalDate": "231218",
    "Language": "en"
}

```

Example response:

```

Response: {
    "Result": "0",
    "AuthCode": "147339",
    "TerminalID": "10000085",
    "HostTransID": "08147339 231218",
    "RespMessage": "APPROVED",
}

```

```
"MaskedPAN": "*****6335",
"CurrencyCode": "978",
"Signature": "N",
"SequenceNumber": "001146003",
"TransactionTime": "20231218125043",
"AmountAuthorized": "31.00",
"CustomerReceipt": "    P2C, s. r. o.\n    Prievozska 4\|C\n    831 04 Bratislava\n-----
-----\n146 18.12.2023 12:53:43 0006\n RECEIPT FOR CUSTOMER\n-----
-----\n Incremental preauthor.\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\nCard: *****6335\nVISA\n\nAmount EUR      31.00\n\nTERMINAL:
10000085\n18.12.2023 00:00:00\nTotal amount EUR    \|u000262.00\n Approval code:
147339\n Sequence number: 001146003\n\nRC:          000\n\n    APPROVED\n-----
-----\n    Thank you\n Keep the receipt for later\n    checking\nVersion:
P2C A01.01 (001)_gp\n",
"HostRC": "000",
"TransactionID": "4711699691",
"MerchantReceipt": "    P2C, s. r. o.\n    Prievozska 4\|C\n    831 04 Bratislava\n-----
-----\n146 18.12.2023 12:53:43 0006\n RECEIPT FOR MERCHANT\n-----
-----\n Incremental preauthor.\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\nCard: *****6335\nVISA\n\nAmount EUR      31.00\n\nTERMINAL:
10000085\n18.12.2023 00:00:00\nTotal amount EUR    \|u000262.00\n Approval code:
147339\n Sequence number: 001146003\n\nRC:          000\n\n    APPROVED\n-----
-----\n    Thank you\n Keep the receipt for later\n    checking\nVersion:
P2C A01.01 (001)_gp\n"
}
```

3.7.1. Request fields for CI:

Field	Description	Mandatory
Operation	CI for increamental preauthorization	Yes
HostTransID	Host transaction identifier of previous transaction	Yes
Control	Check section about control	No
InvNumber	Check section about invoice number	No
TransactionID	Number representing each transaction	Yes
CurrencyCode	Currency of transaction	No
Language	Language of receipt and payment screen	No
TerminalID	Terminal identifier of preauthorization	Yes
Last4PANnumber	Last four digits of PAN used in the preauthorization	Yes
SequenceNumber	Sequence number sent in preauthorization	Yes
OriginalDate	Date of preauthorization in format YYMMDD	Yes

3.7.2. Response fields for CI:

Field	Description
Result	Result of payment, check section result for further information
AuthCode	Authorization code provided from authorization
TerminalID	Terminal identifier
HostTransID	Specific transaction ID from the authorization.
RespMessage	Response in language based on terminal configuration/request msg
MaskedPAN	Masked primary account number
CurrencyCode	Currency in ISO format.
Signature	If for payment signature was necessary or not.
SequenceNumber	Specific number used for identifying transaction with authorization
TransactionTime	Transaction timestamp in the format YYYYMMDDHHMMSS
AmountAuthorized	Full amount authorized by host
CustomerReceipt	Customer receipt in string format
HostRC	Response from host in numeric format
TransactionID	ID of transaction based on request
MerchantReceipt	Merchant receipt in string format

3.8. CF – Preauthorization Completion

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
"sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CF")
    json.put("HostTransID", "08147339 231218")
    json.put("TerminalID", "10000085")
    json.put("Last4PANnumber", "1234")
    json.put("SequenceNumber", "00000000")
    json.put("OriginalDate", "231223")
    json.put("Amount", "11.00")
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {

```

Example request:

```
Request: {
  "Operation": "CF",
  "Amount": "62.00",
  "Control": 1,
  "TransactionID": "7177639150",
  "InvNumber": "12345678",
  "CurrencyCode": "978",
  "HostTransID": "08147339 231218",
  "TerminalID": "10000085",
  "Last4PANnumber": "6335",
  "SequenceNumber": "001146003",
  "OriginalDate": "231218",
  "Language": "en"
}
```

Example response:

```
Response: {
  "Result": "0",
  "AuthCode": "147339",
  "TerminalID": "10000085",
  "HostTransID": "08147339 231218",
  "RespMessage": "APPROVED",
  "MaskedPAN": "*****6335",
  "CurrencyCode": "978",
  "Signature": "N",
  "SequenceNumber": "001146005",
  "TransactionTime": "20231218125043",
  "AmountAuthorized": "62.00",
  "CustomerReceipt": "    P2C, s. r. o.\n    Prievozska 4\|V\C\n    831 04 Bratislava\n-----
-----\n146 18.12.2023 12:55:36 0007\n RECEIPT FOR CUSTOMER\n-----
-----\n PreAuth Completion\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: *****6335\nVISA\n\nAmount EUR
62.00\n\n\nApproval code: 147339\nSequence number: 001146005\nOriginal seq. num:
001146003\n\nRC:          000\n\n    APPROVED\n-----\n
Thank you\n Keep the receipt for later\n    checking\nVersion: P2C A01.01 (001)_gp\n",
  "HostRC": "000",
  "TransactionID": "7177639150",
  "MerchantReceipt": "    P2C, s. r. o.\n    Prievozska 4\|V\C\n    831 04 Bratislava\n-----
-----\n146 18.12.2023 12:55:36 0007\n RECEIPT FOR MERCHANT\n-----
-----\n PreAuth Completion\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nCard: *****6335\nVISA\n\nAmount EUR
62.00\n\n\nApproval code: 147339\nSequence number: 001146005\nOriginal seq. num:
001146003\n\nRC:          000\n\n    APPROVED\n-----\n
Thank you\n Keep the receipt for later\n    checking\nVersion: P2C A01.01 (001)_gp\n"
}
```

3.8.1. Request fields for CF:

Field	Description	Mandatory
Operation	CF for preauthorization completion	Yes
HostTransID	Host transaction identifier of previous transaction	Yes
Control	Check section about control	No
TransactionID	Number representing each transaction	Yes
CurrencyCode	Currency of transaction	No
Language	Language of receipt and payment screen	No
TerminalID	Terminal identifier of preauthorization	Yes
Last4PANnumber	Last four digits of PAN used in the preauthorization	Yes
SequenceNumber	Sequence number sent in preauthorization	Yes
OriginalDate	Date of preauthorization in format YYMMDD	Yes

3.8.2. Response fields for CF:

Field	Description
Result	Result of payment, check section result for further information
AuthCode	Authorization code provided from authorization
TerminalID	Terminal identifier
HostTransID	Specific transaction ID from the authorization.
RespMessage	Response in language based on terminal configuration/request msg
MaskedPAN	Masked primary account number
CurrencyCode	Currency in ISO format.
Signature	If for payment signature was necessary or not.
SequenceNumber	Specific number used for identifying transaction with authorization
TransactionTime	Transaction timestamp in the format YYYYMMDDHHMMSS
AmountAuthorized	Full amount authorized by host
CustomerReceipt	Customer receipt in string format
HostRC	Response from host in numeric format
TransactionID	ID of transaction based on request
MerchantReceipt	Merchant receipt in string format

3.9. CU – Universal Cancellation

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
"sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CU")
    json.put("HostTransID", "08147339 231218")
    json.put("TerminalID", "10000085")
    json.put("Last4PANnumber", "1234")
    json.put("SequenceNumber", "000000000")
    json.put("OriginalDate", "231223")
    json.put("Amount", "11.00")
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {

```

Example request:

```

Request: {
    "Operation": "CU",
    "Amount": "62.00",
    "Control": 1,
    "TransactionID": "801018773",
    "CurrencyCode": "978",
    "TerminalID": "10000085",
    "Last4PANnumber": "6335",
    "SequenceNumber": "001146002",
    "OriginalDate": "231218",
    "Language": "en"
}

```

Example response:

```

Response: {
    "Result": "1",
    "TerminalID": "10000085",
    "RespMessage": "APPROVED",
    "AuthCode": "147339",
    "TerminalID": "10000085",

```



```

"HostTransID": "08147339 231218",
"RespMessage": "APPROVED",
"MaskedPAN": "*****6335",
"CurrencyCode": "978",
"Signature": "N",
"SequenceNumber": "001146003",
"TransactionTime": "20231218125043",
"AmountAuthorized": "31.00",
"CustomerReceipt": "    P2C, s. r. o.\n    Prievozska 4\|C\n    831 04 Bratislava\n-----
\n146 18.12.2023 12:58:03 0008\n    RECEIPT FOR CUSTOMER\n-----
\n    Cancellation\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nAmount EUR        62.00\n\nOriginal transaction:\n-----\n
18.12.2023 00:00:00\n    PAYMENT\nTERMINAL:    10000085\nCard:
*****6335\nVISA\nApproval code: 956323\nSequence number: 001146002\n-----
\n\nRC:            000\n\n    APPROVED\n-----\n
Thank you\n Keep the receipt for later\n    checking\nVersion: P2C A01.01 (001)_gp\n",
"TransactionID": "801018773",
"MerchantReceipt": "    P2C, s. r. o.\n    Prievozska 4\|C\n    831 04 Bratislava\n-----
\n146 18.12.2023 12:58:03 0008\n    RECEIPT FOR MERCHANT\n-----
\n    Cancellation\n\nMERCHANT ID: 739900009911\nTERMINAL:
10000085\n\nAmount EUR        62.00\n\nOriginal transaction:\n-----\n
18.12.2023 00:00:00\n    PAYMENT\nTERMINAL:    10000085\nCard:
*****6335\nVISA\nApproval code: 956323\nSequence number: 001146002\n-----
\n\nRC:            000\n\n    APPROVED\n-----\n
Thank you\n Keep the receipt for later\n    checking\nVersion: P2C A01.01 (001)_gp\n"
}
    
```

3.9.1. Request fields for CU:

Field	Description	Mandatory
Operation	CU for user cancellation	Yes
HostTransID	Host transaction identifier of previous transaction	Yes
Control	Check section about control	No
TransactionID	Number representing each transaction	Yes
CurrencyCode	Currency of transaction	No
Language	Language of receipt and payment screen	No
TerminalID	Terminal identifier of preauthorization	Yes
Last4PANnumber	Last four digits of PAN used in the preauthorization	Yes
SequenceNumber	Sequence number sent in preauthorization	Yes
OriginalDate	Date of preauthorization in format YYMMDD	Yes

3.9.2. Response fields for CU:

Field	Description
Result	Result of payment, check section result for further information
AuthCode	Authorization code provided from authorization
TerminalID	Terminal identifier
HostTransID	Specific transaction ID from the authorization.
RespMessage	Response in language based on terminal configuration/request msg
CardBrand	Brand name of card inserted/tapped/swiped
BIN	First digits of card used for identifying card type
MaskedPAN	Masked primary account number
PINIndicator	If for payment Pin was inserted or not.
AID	Application identifier of card.
CurrencyCode	Currency in ISO format.
Signature	If for payment signature was necessary or not.
SequenceNumber	Specific number used for identifying transaction with authorization
TransactionTime	Transaction timestamp in the format YYYYMMDDHHMMSS
AmountAuthorized	Full amount authorized by host
CustomerReceipt	Customer receipt in string format
HostRC	Response from host in numeric format
TransactionID	ID of transaction based on request
MerchantReceipt	Merchant receipt in string format

3.10. CL – Handshake

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
    "sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CL")
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {
    Log.d(TAG, "Exception: ${e.message}")
}
    
```

Example request:

```
Request: {
  "Operation": "CL",
  "TransactionID": "653332556"
}
```

Example response:

```
Response:
{
  "Result": "0",
  "RespMessage": "Admin OK",
  "TransactionID": "653332556",
  "MerchantReceipt": " Pay2Con s. r. o. \n Vajnorská 100 \B \n 831 04
Bratislava \n ----- \n 001 11.01.2024 09:31:48 0003 \n POTVRDENKA PRE
OBCHODNÍKA \n ----- \n Test linky \n \n MERCHANT ID:
7311211752 \n TERMINÁL: 10000085 \n \n RC: 007 \n \n PRIJATÉ
ADMIN \n ----- \n Ďakujeme Vám. \n Potvrdenku si odložte \n pre
prípadnú kontrolu. \n Verzia: P2C A01.01 (001)_gp \n "
```

3.10.1. Request fields for CL:

Field	Description	Mandatory
Operation	CL for Handshake	Yes
TransactionID	ID of transaction based on request	Yes
Control	Check section about control	No

3.10.2. Response fields for CL:

Field	Description
Result	Result of payment, check section result for further information
RespMessage	Response in language based on terminal configuration/request msg
TransactionID	ID of transaction based on request
MerchantReceipt	Merchant receipt in string format

3.8. CM – POS Download

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.posterminal",
"sk.co.kompakts.posterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "CM")
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {
    Log.d(TAG, "Exception: ${e.message}")
}

```

Example request:

Request:

```

{
  "Operation": "CM",
  "TransactionID": "7287666363"
}

```

Example response:

Response:

```

{
  "Result": "0",
  "TransactionID": "7287666363",
  "RespMessage": "Download OK !",
  "LastTmsCheck": "2024-01-11 09:32:55"
}

```

3.11.1. Request fields for CM:

Field	Description	Mandatory
Operation	CM for POS download	Yes
Control	Check section about control	No
TransactionID	Number representing each transaction	Yes

3.11.2. Response fields for CM:

Field	Description
Result	Result of payment, check section result for further information
RespMessage	Response in language based on terminal configuration/request msg
TransactionID	ID of transaction based on request
LastTmsCheck	Timestamp of the last connection to TMS

3.12. TG – Get Terminal Config -> This feature is not available in GP versions less than 1024.

```

val intent = Intent()
intent.component = ComponentName("sk.co.kompakts.postterminal",
"sk.co.kompakts.postterminal.MainActivity")
val json= JSONObject()
var req = ""
try {
    json.put("Operation", "TG")
    json.put("TransactionID", "0123456789") // use Random()
    json.put("Control", "1") // Control of printing
    json.put("Language", "en") // possible languages sk, cz, de, ro, en
    req = json.toString()
} catch (e: JSONException) {
    e.printStackTrace()
}

if (req.isEmpty()) return

intent.putExtra("POS_EMULATOR_EXTRA", req) // for identifying intent
try {
    startActivityForResult(intent, <requestCode>)
} catch (e: Exception) {
    Log.d(TAG, "Exception: ${e.message}")
}
    
```

Example request:

Request:

```
{
  "Operation": "TG",
  "TransactionID": "7287116363"
}
```

Example response:

Response:

```
{
  "Result": "0",
  "TransactionID": "3513565019",
  "RespMessage": "Admin OK !",
  "appVersion": "P2C A01.01 (001)_gp_b1001",
  "configState": "0",
  "serialNumber": "1850290076",
  "partNumber": "A920Pro-0AW-RD5-08EU",
  "mid": "7311211752",
  "tid": "10000085",
  "activeCurrency": "978",
  "supportedLang": "en;de;sk;cz;ro",
  "activeLang": "sk",
  "hostEnv": "test"
}
```

3.12.1. Request fields for TG:

Field	Description	Mandatory
Operation	TG for Get Config	Yes
Control	Check section about control	No
TransactionID	Number representing each transaction	Yes

3.12.2. Response fields for TG:

Field	Description
Result	Result of payment, check section result for further information
TransactionID	ID of transaction based on request
RespMessage	Response in language based on terminal configuration/request msg

<i>appVersion</i>	Version of GP PAY application installed
<i>configState</i>	State 1 or state 0 depending on if pos download should be performed
<i>serialNumber</i>	Serial number of HW
<i>partNumber</i>	Part number of HW
<i>mid</i>	Merchant ID
<i>tid</i>	Terminal ID
<i>activeCurrency</i>	Currency active on terminal
<i>supportedLang</i>	All supported language separated by “;”
<i>activeLang</i>	Language set on terminal
<i>hostEnv</i>	Environment of authorization

4. Fields description:

4.1. Request fields

Operation:

- ➔ Operation field is mandatory field in every request. It contains the necessary code of transaction which is parsed in payment application. If application cannot parse message, it will send corresponding response:

```
{
  "Result": "9",
  "RespMessage": "Mandatory Field Missing or has Wrong Value"
}
```

Amount:

- ➔ Amount is mandatory field in operations: CP, CR, CA, CF, CI, CU. If request does not contain amount it will send out response:

```
{
  "Result": "9",
  "RespMessage": "Mandatory Field Missing or has Wrong Value"
}
```

Control:

- ➔ Control field is optional field, which can be present in every request. When setting up a transaction, you have the option to customize how the Point of Sale (POS) system reacts by using the "Control" field. This field is like a switchboard that lets you turn certain features on or off depending on your needs.

Here's a simple guide to what each switch (or bit) does:

- Bit 0x01: Turn this on if you want the POS system to double-check the signature.
- Bit 0x02: Flip this switch if you'd like to have a receipt printed out.
- Bit 0x08: Use this if you prefer not to send the receipt data back as part of the transaction response.
- Bit 0x20: This is for when you want to prompt for a tip at the start of a sale or purchase transaction.

If you leave the "Control" field blank, the POS system will work in its default mode, which means it won't print receipts but will send the receipt data to the Electronic Cash Register application.

InvNumber:

- ➔ Invoice Number is special number with maximum length of 10 characters. This value is sent direct to authorization center. Later it can be used for identifying transaction for the merchant. It is optional field, and it can be send during debit transactions.

TransactionID:

- ➔ TransactionID is mandatory field for every request. It is required to be sent and it is used for mapping transaction between ECR application and payment application. It should have length of 10 digits. Most common solution for this is to use Random () function.

CurrencyCode:

- ➔ Currency code of the transaction. The field can contain three digits value corresponding to the ISO 4217 currency table. Possible values: 203, 978, 840, 946, 346, 826... If you will send out currency not supported by POS, terminal will use default currency. It is optional field.

Language:

- ➔ Language on which transaction should be performed. It is recommended to after change on language call request TG. For getting possible languages check *supportedLang*. Optional field.

TerminalID

- ➔ Mandatory field for enhanced matching transactions such as CU, CI, CF. It is used for mapping TID of previous transaction with new one. This value should be sent from the previously performed transaction. If not correct TID is send out, authorization will provide declined response.

Last4PANnumber:

- ➔ Mandatory field for enhanced matching transactions such as CU, CI, CF. It is used to check if new transaction match PAN of previous transaction which is set to be changed by new request. If not correct last 4 of pan is send out, authorization will provide declined response.

SequenceNumber:

- ➔ Mandatory field for enhanced matching transactions such as CU, CI, CF. It is used to pair transactions. The sequence number is provided in every response of transaction type of operations. If not correct sequence number is provided, authorization will provide declined response.

OriginalDate:

- ➔ Mandatory field for enhanced matching transaction such as CU, CI, CF. It is used to pair transactions. The date needs to be the date of previous transaction. Format of field is YYMMDD. If not correct date is provided, authorization will provide declined response.

HostTransID:

- ➔ Specific transaction id provided from the host. It is used only for the local match of transactions. Optional field.

4.2. Response fields

appVersion:

- ➔ Version of payment application installed on terminal. Only for the information purpose.

configState:

- ➔ Contains information if POS download should be performed or not. In case value is 0, it means POS download does not need to be performed. Otherwise, it is recommended to call POS download operation.

serialNumber:

- ➔ Serial number of HW on which application is installed. Only for the information purpose.

partNumber:

- ➔ Part number of HW on which application is installed. Only for the information purpose.

mid:

- ➔ Merchant ID of terminal ID connected to POS. Only for the information purpose.

tid:

- ➔ Terminal ID connected to POS. Only for the information purpose.

activeCurrency:

- ➔ Currency set on terminal. Currency format corresponding to the ISO 4217 currency table. Possible values: 203, 978, 840, 946, 346, 826...

supportedLang:

- ➔ All values which can be used for the setting language. All languages are separated by character “;”. Only for the information purpose.

activeLang:

- ➔ Language set on terminal. Only for the information purpose.

hostEnv:

- ➔ Informs about environment on which terminal is installed.
Possible options are: Prod, Test, Simulator.

LastTmsCheck:

- ➔ Informs about last time TMS check was performed. It is in timestamp format: YYYY-MM-DD HH:MM:SS.

Result:

- ➔ Indicates the transaction result with the following possible values:
0: Approved/Success
1: Declined
2-8: Declined reserved.
9: Declined by card or terminal or other local error (e.g., communication error, timeout, canceled, etc.)

AuthCode:

- ➔ Authorization code provided upon transaction approval.

TerminalID:

- ➔ Identifier for the terminal where the transaction was processed.

HostTransID:

- ➔ Identifier of the transaction for host purposes.

RespMessage:

- ➔ Provides additional text information on the transaction result. This information can be sent by the host or locally by the payment application.

CardBrand:

- ➔ Identifies the brand of the card used (e.g., VISA, MasterCard).

BIN:

- ➔ Bank Identification Number, which are the first 6 digits of the card number.

MaskedPAN:

- ➔ Masked Personal Account Number to protect cardholder information.

PINindicator:

- ➔ Indicates whether a PIN was entered (Y for yes, N for no).

AID:

- ➔ Application Identifier for smart card only.

CurrencyCode:

- ➔ The currency code for the transaction.

Signature:

- ➔ Specifies whether signature checking is required (Y for yes, N for no).

SequenceNumber:

- ➔ Transaction Sequence Number for a host, used for pairing in case of enhanced matching transaction.

TransactionTime:

- ➔ Transaction date and time in the format yyyyMMddHHmmss.

AmountAuthorized:

- ➔ Full amount authorized by authorization centre.

CustomerReceipt:

- ➔ Formatted text for the customer receipt.

HostRC:

- ➔ Host response code indicating the result from the card issuer.

TransactionID:

- ➔ Specific value for each request
*in case of the same value in last 10 requests GP PAY will return result of request with that TransactionID

MerchantReceipt:

- ➔ Merchant receipt (formatted text) or array in case of multi-merchant transactions.

hostCounters:

- ➔ Host counters that include the number of transactions and amounts for a particular host

terminalCounters:

- ➔ Terminal counters that include the number of transactions and amounts for a particular terminal.