Pay At Table API

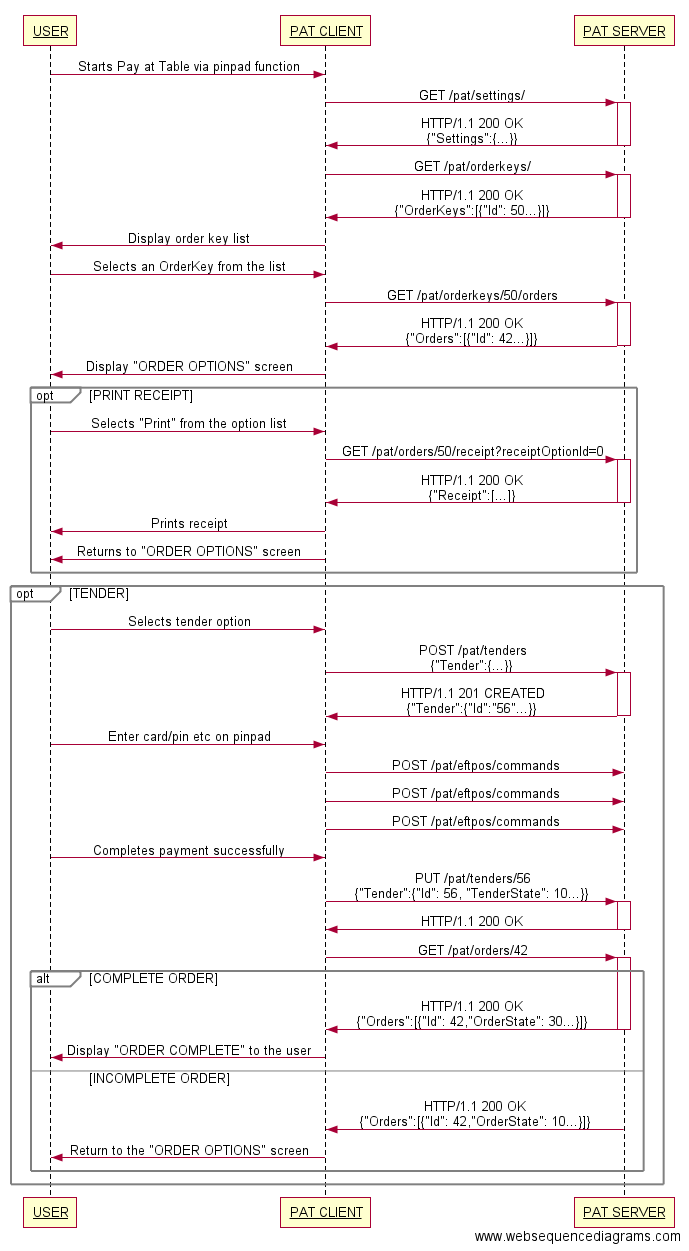
# Overview

The purpose of the Pay at Table API is to provide a common interface for the Pay at Table client (e.g. a pinpad) to retrieve available tables and orders so that payment functions (e.g. tender, customer receipt etc) can be performed on them. These methods can be accessed either via a REST Api server or through the POS system as explained in the later sections.

A typical transaction flow:

* The user initiates a Pay at Table transaction using a function code on the pinpad.
* The Pay at Table client requests the settings from the server
* The Pay at Table client requests a list of tables from the server
* Tables are presented to the user, either as a list using the *DisplayName* property of a *Table* or by allowing the user to manually key a *DisplayNumber*.
* Once the user selects a table, the Pay at Table client requests orders available on that table.
* If no orders are available, the Pay at Table client presents a display to the user and allows them to select another table.
* If orders are available the Pay at Table client presents available options for that order (e.g. print receipt, tender). If multiple orders are available, the Pay at Table client displays all available orders and asks the user to select which order to process.
* If the user selects the *“Print Receipt”* option, the Pay at Table client will request the customer receipt from the server, print it and display the order options again. If multiple print options are available from the settings, the user is asked to select which mode to print before the request is sent to the server.
* If the user selects the *“Tender”* option, the Pay at Table client starts a payment on the pinpad. If multiple tender options are available from the settings, the Pay at Table client displays these options and asks the user to select the tender type before proceeding with the payment.
* The transaction request, display events and transaction event are sent to the server as EFTPOS commands
* Once the payment is complete, the Pay at Table client updates the tender with a completed state. It is assumed as this point the POS server would also update the order state.

The Pay at Table client request the selected order again. If the order is complete a message is displayed on the pinpad, otherwise the user is presented with the order options again.



# Pay At Table Data Request Format

The Pay At Table data requests can be passed via an external server using two methods: a) REST Api, or the b) POS through DoCsdReserved3/OnCsdReserved3 event (ActiveX interface) or the generic POS command request/response (TCP/IP Interface).

## REST Api

### HTTP Response Codes

|  |  |
| --- | --- |
| HTTP Response Codes | Description |
| 200 OK | The request was successful |
| 201 Created | The request was successful and a resource has been created |
| 204 No Content | The request was successful, there is no content in the response |
| 400 Bad Request | The client request is invalid |
| 401 Unauthorised | The client needs to authenticate before it can continue |
| 403 Forbidden | The client doesn’t have access to the resource |
| 404 Not Found | The requested resource wasn’t found |
| 500 Server Error | The server encountered an internal error processing the request. |

### Methods

|  |  |  |
| --- | --- | --- |
|  | HTTP Method | Description |
| [Get Settings](#_Get_Settings) | GET /api/settings | Get settings for the Pay at Table client |
| [Get Tables](#_Get_Order_Keys) | GET /api/tables | Get a lookup list of tables used to find an order |
| [Get Orders By Table](#_Get_Orders_by) | GET /api/tables/{table-id}/orders | Get a list of orders associated with a table |
| [Get Order](#_Get_Order) | GET /api/orders/{order-id} | Get an order based on an order id. |
| [Get Customer Receipt From Order](#_Get_Customer_Receipt) | GET /api/orders/{order-id}/receipt?receiptOptionId=[string] | Get a customer receipt for a given order. Can accept an optional receipt option id. |
| [Create Tender](#_Create_Tender) | POST /api/tenders | Create a tender |
| [Update Tender](#_Update_Tender) | PUT /api/tenders/{tender-id} | Update a tender |
| [Create EFTPOS Command](#_Create_EFTPOS_Command) | POST /api/eftpos/commands | Create an EFTPOS command |

#### Get Settings

##### Description

Get the settings for the pay at table client

##### Request

GET /api/settings

Do not supply a request body for this method.

##### Response

If successful the body contains a PATResponse object with the Settings property populated a Settings object.

Supported response codes: 200, 400, 401, 403 and 500.

HTTP/1.1 200 OK

Content-type: application/json

{

"Settings": {

"TenderOptions": [{

"Id": "0",

"TenderType": 0,

"Merchant": "00",

"DisplayName": "EFTPOS",

"EnableSplitTender": false

}],

"ReceiptOptions": [{

"Id": "0",

"ReceiptType": 0,

"DisplayName": "Customer"

}],

"CsdReservedString2": "EFTPOS",

"TxnType": "P",

"IsTippingEnabled": false

}

}

#### Get Tables

##### Description

Get a lookup list of tables used to find an order.

The Pay at Table client will either present a list of selectable items to the user using the “DisplayName” property, or request the user enter a number which will be used to find a table based on the “DisplayNumber” property.

The Id property is a unique identifier for the table used in subsequent requests, and is not presented to the user.

##### Request

GET /api/tables

Do not supply a request body for this method.

##### Response

If successful the body contains a PATResponse object with the Tables property populated by an array of Table.

Supported response codes: 200, 400, 401, 403 and 500.

HTTP/1.1 200 OK

Content-type: application/json

{

"Tables": [{

"Id": "50",

"DisplayName": "TABLE 1",

"DisplayNumber": 1

},

{

"Id": "51",

"DisplayName": "TABLE 2",

"DisplayNumber": 2

},

{

"Id": "52",

"DisplayName": "TABLE 3",

"DisplayNumber": 3

}

}

#### Get Orders by Table

##### Description

Get a list of orders associated with a table.

The Pay at Table client will send this request after a user has selected one of the tables returned from a previous call to Get Tables.

##### Request

GET /api/tables/{table-id}/orders

Do not supply a request body for this method

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| table-id | String | Required. The id of an table orders are being requested from. |

##### Response

If successful the body contains a PATResponse object with the Orders property populated by an array of Order.

Supported response codes: 200, 400, 401, 403 and 500.

HTTP/1.1 200 OK

Content-type: application/json

{

"Orders": [{

"Id": "101",

"DisplayName": "Elsa",

"OrderState": 0,

"AmountOwing": 100.00,

"TableId": "50"

}]

}

#### Get Order

##### Description

Get an order based on an order id.

The Pay at Table client will send this request after a user has selected one of the orders returned from a previous call to Get Orders by Table.

##### Request

GET /api/orders/{order-id}

Do not supply a request body for this method.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| order-id | String | Required. The id of the order being requested. |

##### Response

If successful the body contains a PATResponse object with the Order property populated by an Order.

Supported response codes: 200, 400, 401, 403, 404 and 500.

HTTP/1.1 200 OK

Content-type: application/json

{

"Order": {

"Id": "101",

"DisplayName": "Elsa",

"OrderState": 0,

"AmountOwing": 100.00,

"TableId": "50"

}

}

#### Get Customer Receipt from Order

##### Description

Get a customer receipt based on an order id.

##### Request

GET /api/orders/{order-id}/receipt?receiptOptionId=[string]

Do not supply a request body for this method

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| order-id | String | Required. The id of the order the receipt is being requested from. |
| receiptOptionId | String | Optional. The id of the ReceiptOption used to generate this receipt request. |

##### Response

If successful the body contains a PATResponse object with the Receipt property populated by a Receipt.

Supported response codes: 200, 400, 401, 403, 404 and 500.

HTTP/1.1 200 OK

{

"Receipt": {

"Lines": ["Line 1","Line 2","Line 3"]

}

}

#### Create Tender

##### Description

Creates a tender. A tender is an object which contains information about a payment.

##### Request

POST /api/tenders

{

"Tender": {

"Id" : null,

"OrderId": "101",

"TenderOptionId": "0",

"TenderState": 0,

"AmountPurchase": 100.00,

"OriginalAmountPurchase ": 100.00

}

}

The request body contains a PATRequest with the Tender property populated by a Tender.

The *OrderId* property must reference a valid order.

The *TenderOptionId* property references the tender option selected by the user.

##### Response

If successful the body will contain a PATResponse object with the Tender property populated by a Tender. The Tender in the response will have the Id property populated by a unique Id.

Supported response codes: 201, 400, 401, 403, 404 and 500.

HTTP/1.1 201 OK

Content-type: application/json

{

"Tender": {

"Id": "1042",

"OrderId": "101",

"TenderState": 0,

"AmountPurchase": 100.00,

"OriginalAmountPurchase": 100.00

}

}

#### Update Tender

##### Description

Updates a tender.

It is possible that the *AmountPurchase* in an updated tender will not be the same as the *AmountPurchase* in the original tender. E.g. A $100 purchase on a giftcard is completed for the remaining amount on the card ($80.50).

The *Id* property must point to a valid tender and match the {tender-id} in the request url.

The *OrderId* property must point to a valid order.

##### Request

PUT /api/tenders/{tender-id}

Content-type: application/json

{

"Tender": {

"Id": "1042",

"TenderOptionId": "0",

"OrderId": "101",

"TenderState": 2,

"AmountPurchase": 80.50,

"OriginalAmountPurchase": 100.00

}

}

The request body contains a PATRequest with the Tender property populated by a Tender.

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| tender-id | String | Required. The id of the tender being updated. |

##### Response

If successful, this method returns a [PATResponse](#_PATResponse) object with the Tender property populated by a [Tender](#_Tender). In most cases the Tender in the response will mirror the request.

HTTP/1.1 200 OK

Content-type: application/json

{

"Tender": {

"Id": "1042",

"TenderOptionId": "0",

"OrderId": "101",

"TenderState": 2,

"AmountPurchase": 80.50,

"OriginalAmountPurchase": 100.00

}

}

#### Create EFTPOS Command

##### Description

Create an EFTPOS command

##### Request

The request body contains a PATRequest with the EFTPOSCommand property populated by an EFTPOSCommand

POST /api/eftpos/commands

{

"EFTPOSCommand": {

"TenderId": "0",

"OrigionalEFTPOSCommandId": "0",

"EFTPOSCommandType": 0,

"EFTPOSCommandState": 20

"AccountType": "",

"AmtCash": 0.0,

"AmtPurchase": 100.0,

"AmtTip": 0.0,

"AmtTotal": 0.0,

"Application": "",

"AuthCode": "",

"Caid": "",

"Catid": "",

"CardName": "",

"CardType": "",

"CsdReservedString1": "",

"CsdReservedString2": "",

"CsdReservedString3": "",

"CsdReservedString4": "",

"CsdReservedString5": "",

"CsdReservedBool1": false,

"CutReceipt": false,

"CurrencyCode": "",

"DataField": "",

"Date": "",

"DateExpiry": "",

"DateSettlement": "",

"DialogPosition": "",

"DialogTitle": "",

"DialogType": "",

"DialogX": 0,

"DialogY": 0,

"EnableTip": false,

"EnableTopmost": false,

"Merchant": "",

"MessageType": "",

"PanSource": " ",

"Pan": "",

"PosProductId": "",

"PurchaseAnalysisData": "",

"ReceiptAutoPrint": false,

"ResponseCode": "",

"ResponseText": "",

"Rrn": "",

"Success": false,

"STAN": "",

"Time": "",

"TxnRef": "",

"TxnType": "",

"Track1": "",

"Track2": ""

}

}

##### Response

If successful, this method returns a [PATResponse](#_PATResponse) object with the EFTPOSCommand property populated by an [EFTPOSCommand](#_EFTPOSCommand). In most cases the EFTPOSCommand in the response will mirror the request.

HTTP/1.1 200 OK

Content-type: application/json

{

"EFTPOSCommand": {

"TenderId": "0",

"OrigionalEFTPOSCommandId": "0",

"EFTPOSCommandType": 0,

"EFTPOSCommandState": 20

"AccountType": "",

"AmtCash": 0.0,

"AmtPurchase": 100.0,

"AmtTip": 0.0,

"AmtTotal": 0.0,

"Application": "",

"AuthCode": "",

"Caid": "",

"Catid": "",

"CardName": "",

"CardType": "",

"CsdReservedString1": "",

"CsdReservedString2": "",

"CsdReservedString3": "",

"CsdReservedString4": "",

"CsdReservedString5": "",

"CsdReservedBool1": false,

"CutReceipt": false,

"CurrencyCode": "",

"DataField": "",

"Date": "",

"DateExpiry": "",

"DateSettlement": "",

"DialogPosition": "",

"DialogTitle": "",

"DialogType": "",

"DialogX": 0,

"DialogY": 0,

"EnableTip": false,

"EnableTopmost": false,

"Merchant": "",

"MessageType": "",

"PanSource": " ",

"Pan": "",

"PosProductId": "",

"PurchaseAnalysisData": "",

"ReceiptAutoPrint": false,

"ResponseCode": "",

"ResponseText": "",

"Rrn": "",

"Success": false,

"STAN": "",

"Time": "",

"TxnRef": "",

"TxnType": "",

"Track1": "",

"Track2": ""

}

}

### 







































## POS

### ActiveX Interface

#### POS to EFT-Client command

* Set TxnType to ‘@’
* Set CsdReservedString1 to the Pay @ Table response structure below
* Call DoCsdReserved3()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Field | Length | Format | Description |
| 1 | Header length | 6 | Numeric. Right aligned, zero padded. | The length of the Pay @ Table header to follow. |
| 2 | Response header | \* | Alphanumeric | JSON formatted Pay @ Table response header. To find the length of the content, check Content-length in the header. |
| 3 | Response content | \* | Alphanumeric | JSON formatted Pay @ Table response content |

#### EFT-Client to POS command

* OnCsdReserved3 event will fire
* TxnType will be set to ‘@’
* DataField will be set to the Pay @ Table request structure below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Field | Length | Format | Description |
| 1 | Header length | 6 | Numeric. Right aligned, zero padded. | The length of the Pay @ Table header to follow. |
| 2 | Request header | \* | Alphanumeric | JSON formatted Pay @ Table request header. To find the length of the content, check Content-length in the header. |
| 3 | Request content | \* | Alphanumeric | JSON formatted Pay @ Table request content |

#### Pay @ Table request header

The request header is JSON formatted.

|  |  |  |
| --- | --- | --- |
| Field | Format | Description |
| Version | Numeric | Pay @ Table message version. Default to 1 |
| Content-type | Alphanumeric | application/json |
| Request-type | Alphanumeric | GET,PUT,POST,DELETE |
| Request-method | Alphanumeric | Settings  Tables  TableOrders  Order  OrderReceipt  Tender  EFTPOSCommand |
| Content-length | Numeric | The length of the content to follow |

#### Pay @ Table response header

The header is JSON formatted.

|  |  |  |
| --- | --- | --- |
| Field | Format | Description |
| Version | Numeric | Pay @ Table message version. By default, 1. |
| Content-type | Alphanumeric | application/json |
| Request-type | Alphanumeric | Mirrored from the request |
| Request-method | Alphanumeric | Mirrored from the request |
| Response-code | Numeric | One of the HTTP response codes. 200, 201, 204, 400, 401, 403, 404, 500. |
| Response-text | Alphanumeric | One of the HTTP response codes texts |
| Content-length | Numeric | The length of the content to follow |

#### Sample

Get Tables method

Request [HeaderLength][Header][Content]

A length of 137, followed by the request header. There is no content so that field is not included.

000137

{

"Version": 1,

"ContentType": "application/json",

"RequestType": "GET",

"RequestMethod": "Tables",

"ContentLength": 0

}

Response [HeaderLength][Header][Content]

000188

{

"Version": 1,

"ContentType": "application/json",

"RequestType": "GET",

"RequestMethod": "Tables",

"ResponseCode": 200,

"ResponseText": "OK",

"ContentLength": 0

}

{

"Tables": [{

"Id": "50",

"DisplayName": "TABLE 1",

"DisplayNumber": 1

},

{

"Id": "51",

"DisplayName": "TABLE 2",

"DisplayNumber": 2

}

}

### TCP/IP Interface

TCP/IP Interface

#### POS to EFT-Client command

* Construct a EFTPayAtTableRequest object specifying the Response Header and Content in JSON format as defined in the table below.
* Call WriteRequestAsync supplying the EFTPayAtTableRequest as the parameter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Field | Length | Format | Description |
| 1 | Start flag | 1 | Alphanumeric | Content header. Default to ‘#’ |
| 2 | Command code | 1 | Alphanumeric | Generic POS command type. Default to ‘X’ |
| 3 | Sub code | 1 | Alphanumeric | Pay @ Table command type. Default to ‘@’ |
| 4 | Header length | 6 | Numeric. Right aligned, zero padded. | The length of the Pay @ Table header to follow. |
| 5 | Response Header | \* | Alphanumeric | JSON formatted header. To find the length of the content, check Content-length in the header. |
| 6 | Content | \* | Alphanumeric | JSON formatted Pay @ Table response content |

#### EFT-Client to POS command

* Call ReadResponseAsync.
* Await a returned object type of EFTPayAtTableResponse to access the Request Header and Content.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Field | Length | Format | Description |
| 1 | Start flag | 1 | Alphanumeric | Content header. Default to ‘#’ |
| 2 | Command code | 1 | Alphanumeric | Generic POS command type. Default to ‘X’ |
| 3 | Sub code | 1 | Alphanumeric | Pay @ Table command type. Default to ‘@’ |
| 4 | Header length | 6 | Numeric. Right aligned, zero padded. | The length of the Pay @ Table header to follow. |
| 5 | Request Header | \* | Alphanumeric | JSON formatted Pay @ Table request header. To find the length of the content, check Content-length in the header. |
| 6 | Content | \* | Alphanumeric | JSON formatted Pay @ Table request content |

#### Sample

GET Settings method

Request

[Start flag][message length][Command Code][Sub-code][Response Message][header Length][Header][Content]

A length of 159, followed by the request header. There is no content so that field is not included.

#0192X@APPROVED 000159{

"Version": 1,

"ContentType": "application/json",

"RequestType": "GET",

"RequestMethod": "Settings",

"ContentLength": 0,

"TableID": "",

"OrderID": "",

"ReceiptOptionId": ""

}

Response

[Start flag][message length][Command Code][Sub-code][header Length][Header][Content]

#0619X@000323{

"Version": 1,

"ContentType": "application/json",

"RequestType": "GET",

"RequestMethod": "Settings",

"ContentLength": 283,

"TableId": "",

"OrderId": "",

"ReceiptOptionId": "",

"tender": {

"Id": null,

"OrderId": null,

"TenderState": 0,

"TenderOptionId": null,

"AmountPurchase": 0.0,

"OriginalAmountPurchase": 0.0

},

"ResponseCode": 200,

"ResponseText": "Ok"

}

{

"Tables": null,

"Orders": null,

"Order": null,

"Receipt": null,

"EFTPOSCommand": null,

"Tender": null,

"Settings": {

"TenderOptions": [

{

"Id": "",

"TenderType": 0,

"Merchant": "00",

"DisplayName": "EFTPOS",

"EnableSplitTender": false

}

],

"ReceiptOptions": [

{

"Id": "",

"ReceiptType": 0,

"DisplayName": "Customer"

}

]

}

}

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | Represent EFT-client request commands |
|  |  | Represents a payment |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  | |
|  | |  | | Represents EFT-client request commands |
|  | |  | | Represents a payment |
|  | |  | |  |
|  | |  | |  |
|  | |  | | Represents a sale |
|  | |  | | Proof of sale |
|  | |  | | Defines settings for the pay at table client |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | . If false, the user will not be able to change the amount displayed in the pinpad. |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  | * Order |
|  |  |  |

}],

"CsdReservedString2": "EFTPOS",

"TxnType": "P",

"IsTippingEnabled": false

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
| CsdReservedString2 | String | This property defines which application the EFT-Client is to send the transaction details to. If the property is empty, the default EFTPOS application will be used. Other possible values:   * “EFTPOS” - Use the EFTPOS application (default) * “AGENCY” - Use the Agency application within the terminal. |
| TxnType | String | 1 character text property that determines the type of transaction to perform. If empty, the default “P” is sent out. Possible values:   * “P” – Purchase Cash * “R” – Refund * etc. |
| IsTippingEnabled | Boolean | Indicates to the PC-EFTPOS system to perform a purchase with a possible tip. |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Represent EFT-client request commands

"EFTPOSCommand": {

"TenderId": "0",

"OrigionalEFTPOSCommandId": "0",

"EFTPOSCommandType": 0,

"EFTPOSCommandState": 20

"AccountType": "",

"AmtCash": 0.0,

"AmtPurchase": 100.0,

"AmtTip": 0.0,

"AmtTotal": 0.0,

"Application": "",

"AuthCode": "",

"Caid": "",

"Catid": "",

"CardName": "",

"CardType": "",

"CsdReservedString1": "",

"CsdReservedString2": "",

"CsdReservedString3": "",

"CsdReservedString4": "",

"CsdReservedString5": "",

"CsdReservedBool1": false,

"CutReceipt": false,

"CurrencyCode": "",

"DataField": "",

"Date": "",

"DateExpiry": "",

"DateSettlement": "",

"DialogPosition": "",

"DialogTitle": "",

"DialogType": "",

"DialogX": 0,

"DialogY": 0,

"EnableTip": false,

"EnableTopmost": false,

"Merchant": "",

"MessageType": "",

"PanSource": " ",

"Pan": "",

"PosProductId": "",

"PurchaseAnalysisData": "",

"ReceiptAutoPrint": false,

"ResponseCode": "",

"ResponseText": "",

"Rrn": "",

"Success": false,

"STAN": "",

"Time": "",

"TxnRef": "",

"TxnType": "",

"Track1": "",

"Track2": ""

}

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |