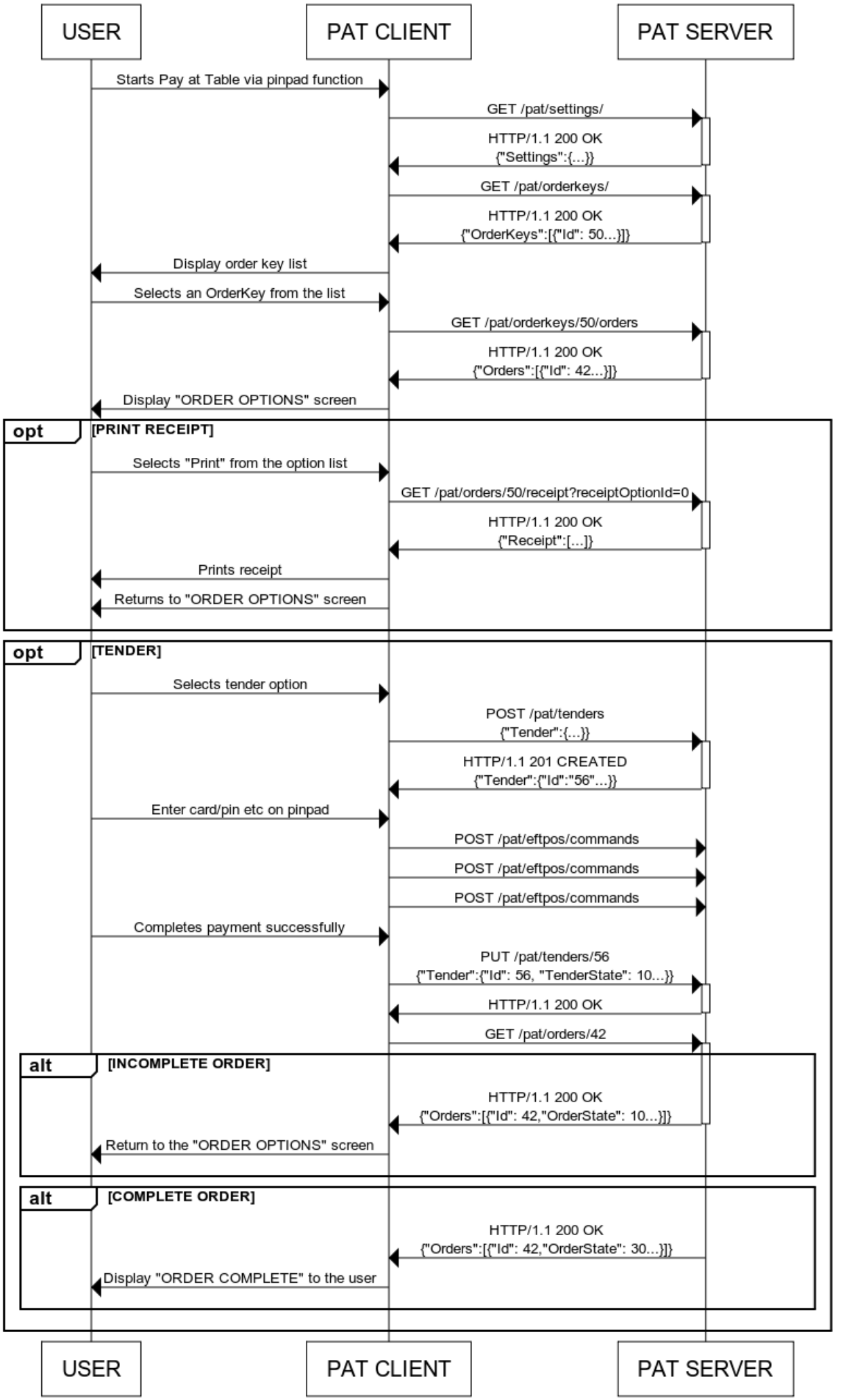
PAY AT TABLE API

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.

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 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 the user selects the *"Tender"* option, the Pay at Table client starts a payment on the pinpad.
- 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.



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 /pat/settings	Get settings for the Pay at Table client
Get Tables	GET /pat/tables	Get a lookup list of tables used to find an order
Get Orders By Table	GET /pat/tables/{table-id}/orders	Get a list of orders associated with a table
Get Order	GET /pat/orders/{order-id}	Get an order based on an order id.
Get Customer Receipt From Order	GET /pat/orders/{order-id}/receipt	Get a customer receipt for a given order
Create Tender	POST /pat/tenders	Create a tender
Update Tender	PUT /pat/tenders/{tender-id}	Update a tender
Create EFTPOS Command	POST /pat/eftpos/commands	Create an EFTPOS command

Get Settings

Description

Get the settings for the pay at table client

Request

```
GET /pat/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"
            }]
      }
}
```

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 /pat/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 /pat/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 /pat/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 /pat/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 /pat/tenders
{
      "Tender": {
            "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",
            "TenderOptionId": "0",
            "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 /pat/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 object with the Tender property populated by a 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 /pat/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 object with the EFTPOSCommand property populated by an 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": ""
      }
}
```

Model

PATRequest

Description

A wrapper for a request to the pay at table API. The contents will depend on the method being called.

```
{
	"EFTPOSCommand": ...,
	"Tender": ...
}
```

Properties

Name	Type	Description
EFTPOSCommand	EFTPOSCommand	
Tender	Tender	

PATResponse

Description

A wrapper for a response from the pay at table API. The contents will depend on the method which generated the response.

```
{
	"Tables": [],
	"Orders": [],
	"Order": ...,
	"Receipt": ...,
	"EFTPOSCommand": ...,
	"Tender": ...,
	"Settings": ...
}
```

Properties

Name	Type	Description
EFTPOSCommand	EFTPOSCommand	
Tender	Tender	
Orders	Order[]	An array of Order
Tables	Table[]	An array of Table
Order	Order	
Receipt	Receipt	
Settings	Settings	

TenderOption

Description

The tender option describes a payment option available to the Pay at Table client. This will typically be “EFTPOS”, however other options (such as gift card) could be supported.

```
{
	"Id": "0",
	"TenderType": 0,
	"Merchant": "00",
	"DisplayName": "EFTPOS",
	"EnableSplitTender": false
}
```

Properties

Name	Type	Description
Id	String	A unique identifier for this tender option. This is passed back to the server when a tender is created.
TenderType	Integer	Defines how this tender option is handled by the Pay at Table client. Possible values: • (0) EFTPOS
Merchant	String	The merchant code to use in the request if this tender option is to be sent to a pinpad. Default to “00”.
DisplayName	String	Max 14 characters. A name which can be presented to the user to identify this tender option
EnableSplitTender	Boolean	True if the user should be able to tender for an amount less than the total of the order

ReceiptOption

Description

Describes a receipt option available to the Pay at Table client. This will typically be “Customer”, however other options could be supported.

```
{
	"Id": "0",
	"ReceiptType": 0,
	"DisplayName": "Customer"
}
```

Properties

Name	Type	Description
Id	String	A unique identifier for this receipt option. This is passed back to the server when a receipt is requested
ReceiptType	Integer	Defines the receipt type. This will be passed back to the server if the user selects this receipt type. Possible values: • (0) Customer
DisplayName	String	Max 14 characters. A name which can be presented to the user to identify this receipt option

Settings

Description

Defines settings for the pay at table client.

```
{
	"Settings": {
		"TenderOptions": [{
			"Id": "0",
			"TenderType": 0,
			"Merchant": "00",
			"DisplayName": "EFTPOS",
			"EnableSplitTender": false
		}],
		"ReceiptOptions": [{
			"Id": "0",
			"ReceiptType": 0,
			"DisplayName": "Customer"
		}]
	}
}
```

Properties

Name	Type	Description
TenderOptions	TenderOption[]	Lists the tender options available to the Pay at Table client. If left null or empty the option to tender will not be available on Pay at Table client when the user selects an order. If only one option is available, the Pay at Table client will automatically select that option when the user chooses to tender.
ReceiptOptions	ReceiptOption[]	Lists the tender options available to the Pay at Table client. If left null or empty the option to print will not be available on Pay at Table client when the user selects an order. If only one option is available, the Pay at Table client will automatically select that option when the user chooses an order.

Table

Description

Defines an item in a lookup table used to find an order.

For example, an array of Table could represent the tables in a restaurant. The user could then be presented with either a list of table names contained in the *DisplayName* property, or the *DisplayNumber* property could be used to select a specific *Table*.

After the user has selected an *Table*, the Pay at Table client will call the */pat/tables/{table-id}/orders* method to retrieve the orders available for this *Table*.

```
{
	"Id": "50",
	"DisplayName": "TABLE 1",
	"DisplayNumber": 1
}
```

Properties

Name	Type	Description
Id	String	Unique identifier.
DisplayName	String	Max 14 characters. A name which represents this table that could be displayed to a user.
DisplayNumber	Integer	A number which represents this table that could be displayed to a user.

Order

Description

An *Order* defines a sale. Orders available for tender will have an *OrderState* set to 10 (active).

```
{
	"Id": "101",
	"DisplayName": "Elsa",
	"OrderState": 0,
	"AmountOwing": 100.00,
	"TableId": "50"
}
```

Properties

Name	Type	Description
Id	String	Unique identifier. Read only.
DisplayName	String	Max 14 characters. A name which represents this table that could be displayed to a user.
OrderState	Integer	The state of the order. This is used by the Pay at Table client to determine if an order is available for tender. Possible values: • (0) Pending – The order exists, but isn't yet available for tender. • (10) Active – The order exists and is available for tender. • (20) Tendering – A tender is currently in progress. The result is not known. The order is not available for tender. • (30) Complete – The order is complete and is not available for tender.
AmountOwing	Decimal	The outstanding amount on this order. This is used by the Pay at Table client to determine the maximum tender amount.
TableId	String	The id of the *Table* attached to this order. Can be null.

Tender

Description

A *Tender* defines a payment.

```
{
	"Id": "1042",
	"TenderOptionId": "0",
	"TenderState": 2,
	"AmountPurchase": 80.00,
	"OrigionalAmountPurchase": 100.00
}
```

Properties

Name	Type	Description
Id	String	Unique identifier. Read only.
TenderOptionId	String	The id of the tender option the operator selected to create this tender
TenderState	Integer	The state of a tender is defined by the *TenderState* property. The initial state is set to *Pending* (0). When the payment is complete the *Tender* object will be updated and the *TenderState* changed to *CompletedSuccessful* (1) or *CompletedUnsuccessful* (2).
AmountPurchase	Decimal	The amount of this tender.
Origional AmountPurchase	Decimal	If the tender amount is changed (e.g. A $100 purchase on a giftcard is completed for the remaining amount on the card - $80.50) this value will reflect the original tender amount before it was changed.

Receipt

Description

```
{
	"Receipt":
	{
		"Lines":
		[
			"---------LINE 1---------",
			"---------LINE 2---------",
			"---------LINE 3---------"
		]
	}
}
```

Properties

Name	Type	Description
Lines	String[]	An array of lines to appear on the receipt. Each receipt line has a maximum of 24 characters

EFTPOSCommand

Description

```
{
. . .
}
```

Properties

Name	Type	Description
Id	String	Unique identifier. Read only.
TenderId	String	Id of the tender that this EFTPOS command is associated with
OrigionalEFTPOS CommandId	String	The id of the original EFTPOS request if this is an event.
EFTPOSCommand Type	Integer	DoTransaction = 100, DoLogon = 101, TransactionEvent = 200, LogonEvent = 201, DoKeyPress = 300, DisplayEvent = 400, PrintEvent = 401
EFTPOSCommand State	Integer	AwaitingDeviceAck = 0, AwaitingDeviceResponse = 10, CompletedSuccessful = 20, CompletedUnsuccessful = 30
AccountType	String	
AmtCash	Decimal	
AmtPurchase	Decimal	
AmtTip	Decimal	
AmtTotal	Decimal	
Application	String	
AuthCode	String	
Caid	String	
Catid	String	
CardName	String	
CardType	String	
CsdReservedString1	String	
CsdReservedString2	String	
CsdReservedString3	String	
CsdReservedString4	String	
CsdReservedString5	String	
CsdReservedBool1	Boolean	
CutReceipt	Boolean	
CurrencyCode	String	
DataField	String	
Date	String	
DateExpiry	String	
DateSettlement	String	
DialogPosition	String	
DialogTitle	String	
DialogType	String	
DialogX	Integer	
DialogY	Integer	
EnableTip	Boolean	
EnableTopmost	Boolean	
Merchant	String	
MessageType	String	
PanSource	String	
Pan	String	

PosProductId	String	
PurchaseAnalysisData	String	
ReceiptAutoPrint	Boolean	
ResponseCode	String	
ResponseText	String	
Rrn	String	
Success	Boolean	
STAN	String	
Time	String	
TxnRef	String	
TxnType	String	
Track1	String	
Track2	String	