



BAMAPI

(*Software Brick And Mortar API*)

Version: draft 0.1
Date: 2014-feb-12
Author: Jeroen de Vries

Index

Version history..... 3
ORDERS..... 5
 HTTP GET..... 5
 Expected JSON 6
ORDERDETAILS 8
 HTTP GET..... 8
 Expected JSON 9
 HTTP POST..... 11
 Expected JSON 11

Version history

2004-feb-14: Draft 0.1

The Softwear POS is able to fulfill eCommerce orders in a brick-and-mortar shop running the Softwear POS. In order to do so, the POS needs realtime information from the eCommerce site. This document describes the requirements for eCommerce APIs to allow Softwear POS to exchange the appropriate information.

The Softwear POS expects a restfull api capable of exchanging JSON documents.

The naming of endpoint is arbitrary, endpoints can be configured in the POS. 2 enpoints are required named ORDERS and ORDERDETAILS for convenience.

ORDERS

HTTP GET

An HTTP GET on the ORDERS endpoint is supposed to return a list of unfulfilled orders. The endpoint needs to implement the following interface:

against `www.url2yourAPI.com/orders?[shopid=][user=][password=]`

`shopid`

If the `shopid` parameter is implemented, it is supposed to filter the list so that it will only return data for a specific store. This can be useful if the eCommerce site allows pickups from a store or if the eCommerce site is used from within the store. The `shopid` will be provided by Softwear.

`user`

The optional `user` parameter can be used to protect the API from unauthorized use and/or to authenticate in a multitenant environment.

`password`

The optional `password` parameter can be used to protect the API from unauthorized use.

Expected JSON

The POS expects this API to return a JSON document containing an array of objects. Every object needs the following properties:

order_id	A unique ID to be used in subsequent ORDER api calls
amount	The total order amount multiplied by 100
date	The order date formatted as YYYYMMDD
customer_name	The name of the customer
shop_id	The shopid of the store where the order should be fulfilled (optional)
nr_pieces	Total number of pieces in order
Paid	True if order is already paid for, false if not

For example:

```
[
  {
    "order_id": "1234567890",
    "amount": 12995,
    "date": 20140201,
    "customer_name": "Mr. Ed",
    "shop_id": 50,
    "nr_pieces": 2,
    "paid": true
  },
  {
    "order_id": "1234567891",
    "amount": 9995,
    "date": 20140201,
    "customer_name": "Captain Kirk",
    "shop_id": 50,
    "nr_pieces": 1,
    "paid": true
  },
  {
    "order_id": "1234567892",
    "amount": 8995,
    "date": 20140201,
    "customer_name": "Dr. Doolittle",
    "nr_pieces": 1,
    "paid": true
  }
]
```


ORDERDETAILS

HTTP GET

An HTTP GET on the ORDERDETAILS endpoint is supposed to return details of one particular order.

The endpoint needs to implement the following interface:

www.url2yourAPI.com/orderdetails?order_id=[user=][password=]

`order_id`

This mandatory parameter specifies the unique order which details are queried

`user`

The optional user parameter can be used to protect the API from unauthorized use and/or to authenticate in a multitenant environment.

`password`

The optional password parameter can be used to protect the API from unauthorized use.

Expected JSON

The POS expects this API to return a JSON document containing a single object. The object needs the following properties:

order_id	A same ID used in the api call
amount	The total order amount multiplied by 100
date	The order date formatted as YYYYMMDD
customer_name	The name of the customer
customer_street	The streetname of the customer
customer_houseno	The housenumber of the customer
customer_zipcode	The zipcode of the customer
customer_city	The city of the customer
customer_email	The email address of the customer
customer_birthdate	The birthdate of the customer in YYYYMMDD format (optional)
shop_id	The shopid of the store where the order should be fulfilled (optional)
nr_pieces	Total number of pieces in order
paid	True if order is already paid for, false if not
remarks	Whatever you want
lines	An array of line objects
line.line_id	A unique line_id (optional)
line.article_key	An article key known by Softwear to uniquely identify the SKU. Preferable an EAN or UPC barcode
line.nr_pieces_ordered	The number of pieces the customer ordered
line.nr_pieces_shipped	The number of pieces shipped to the customer so far (optional)
line.nr_pieces_backordered	The number of pieces still to ship to the customer (optional)
price	The price of the article times 100
Vat	The VAT percentage of the article times 100 (optional, defaults to standard VAT setting)

For example:

```
{
  "order_id": "1234567890",
  "amount": 12995,
  "date": "20140201",
  "customer_name": "Mr. Ed",
  "customer_street": "Stable",
  "customer_houseno": 1,
  "customer_zip": "3333 XX",
  "customer_city": "Wilbur's town",
  "customer_email": "ed@stable.com",
  "customer_birthdate": "19491014",
  "shop_id": 50,
  "nr_pieces": 2,
  "paid": true,
  "remarks": "Free hay giftcard",
  "lines": [
    {
```

```
"line_id":123456,
"article_key":1234567890120,
"nr_pieces_ordered":1,
"nr_pieces_shipped":0,
"nr_pieces_backordered":1,
"price":12995,
"vat":0
},
{
  "line_id":123457,
  "article_key":1234567890231,
  "nr_pieces_ordered":1,
  "nr_pieces_shipped":0,
  "nr_pieces_backordered":1,
  "price":0,
  "vat":0
}
]
```

HTTP POST

An HTTP POST on the ORDERDETAILS endpoint is supposed to allow the POS to update the status and/or content of one particular order.

The endpoint needs to implement the following interface:

`www.url2yourAPI.com/orderdetails?order_id=[user=][password=]`

`order_id`

This mandatory parameter specifies the unique order which details are updated

`user`

The optional user parameter can be used to protect the API from unauthorized use and/or to authenticate in a multitenant environment.

`password`

The optional password parameter can be used to protect the API from unauthorized use.

The body of the POST request is supposed to process a document identical as the one described in the GET request on the previous page. The POS will only update:

- The paid property if an unpaid order is now paid by the customer.
- The `nr_pieces_shipped` and `nr_pieces_backordered` properties for lines that are shipped to the customer.
- Additional lines might be added to the lines array of a customer in the shop decides to purchase additional items. These lines may or may not be shipped to the customer depending on the `nr_pieces_shipped` and `nr_pieces_backordered` properties.
- Depending on whether or not additional items were added, the amount and `nr_pieces` properties can be changed accordingly.

Expected JSON

The POS expects this API to return a JSON document containing a single object. The object needs the following properties:

<code>order_id</code>	A same ID used in the api call
<code>status</code>	OK if the update succeeded, FAILED if the update did not succeed
<code>error</code>	An error message in case the update was unsuccessful (optional)

For example:

```
{
  "order_id": "1234567890",
  "status": "OK"
}
```