Optimus – Sabre 2.0 Returns Support on OMS

**Piotr Ignaczak, 0.10**

Table of contents

[Introduction 3](#_Toc457896428)

[Document Purpose 3](#_Toc457896429)

[Related Documents 3](#_Toc457896430)

[Project Background 4](#_Toc457896431)

[Sub-Project Objectives 4](#_Toc457896432)

[Document Information 5](#_Toc457896433)

[Document Sign Off 5](#_Toc457896434)

[Business Level Overview 6](#_Toc457896435)

[System interactions 6](#_Toc457896436)

[Solution overview 7](#_Toc457896437)

[Implementation 7](#_Toc457896438)

[Order provisioning with RMA number 7](#_Toc457896439)

[Returns process 8](#_Toc457896440)

[Returns Update Webservice 8](#_Toc457896441)

[Database 12](#_Toc457896442)

[UI Changes 17](#_Toc457896443)

[Information to be presented: 19](#_Toc457896444)

[Main menu modifications 20](#_Toc457896445)

[Update Order Tracking Service 21](#_Toc457896446)

[Reporting 24](#_Toc457896447)

# Introduction

## Document Purpose

The purpose of this document is to define applications functionality & design to support product returns. The basic concept is described in HLD Sabre - Returns v0.1

This document will define the solution from Optimus platform point of view. Details of surroundings systems (CTS, Tetra) are included in the HLD (see Related Documents below) and are not part of this document.

## Related Documents

1. HLD – Optimus Sabre 2.0 eCommerce
2. HLD Sabre - Returns v0.1
3. SDD - Optimus Sabre2.0 Customer Order Flow Tracing - Cart v0.02

## Project Background

The main goals of Sabre 2.0 project are:

* Provide Samsung Shop Implementation (Sabre 2.0)
* Expand Sabre 1.0 functionality of EXPANSYS shop (to Sabre 2.0 requirements)
* Provide support/modular integration with various internal and external systems
* Provide data for various analysis software

## Sub-Project Objectives

Returns functionality is essential part of the complete user journey. Returns functionality is complex and usually heavily coupled with particular ERP solution. Because of the above Sabre2:0 will be providing only essential layer of returns functionality which will be narrowed down to

* Presenting key information required to perform a refund (i.e. RMA number) (OMS)
* Retrieving and presenting return process updates (OMS)
* Refunds capability (OMS)
* Present end-customer information on returns status (eCommerce SelfCare)

# Document Information

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Comments | Author | Date |
| 0.01 | First draft | Head of Business Systems | 7th July 2016 |
| 0.02 | Separate returned products and case handling | Marek Chmielowski – Senior Solution Architect | 15th Jul. 2016 |
| 0.03 | Integration with refunds | Marek Chmielowski – Senior Solution Architect | 19th Jul. 2016 |
| 0.04 | UI details | Marek Chmielowski – Senior Solution Architect | 20th Jul. 2016 |
| 0.05 | CTS statuses processing | Marek Chmielowski – Senior Solution Architect | 20th Jul. 2016 |
| 0.06 | Comments from Piotr included, Failed delivery handling | Marek Chmielowski – Senior Solution Architect | 22th Jul. 2016 |
| 0.07 | Changes after Reviewers Comments | Marek Chmielowski – Senior Solution Architect | 29th Jul. 2016 |
| 0.08 | Changes after Ozcan Comments | Marek Chmielowski – Senior Solution Architect | 1st Aug. 2016 |
| 0.09 | DB Diagran and names changes | Marek Chmielowski – Senior Solution Architect | 2ed Aug. 2016 |
| 0.09 | Simplified returns history display, added RMA search for all orders | Marek Chmielowski – Senior Solution Architect | 3th Aug. 2016 |

Table 1 – Document Version Control

# Document Sign Off

| Project Title | Name | Action | Sign Off Date |
| --- | --- | --- | --- |
| Programme Manager | Ozkan Mustafa | Sign Off |  |
| Head of Delivery and Quality | David McLean | Sign Off |  |
|  |  |  |  |

Table 2 – Document Reviewers and Sign Off

# Business Level Overview

Sabre 2:0 Returns process consist of follows below steps:

1. Goods sent to customer with return label and RMA number (ERP)
2. Customer calls customer services to create a return case (CTS) – can be skipped if customer directly sends item to DC without contacting first.
3. Customer sends the goods back using filled in returns label (Carrier)
4. Once the goods arrive the distribution centre they are assessed and processed (CTS)
5. If return gets accepted it can trigger refund (manual) (OMS)

At the same time CSAgent should be able to view the Returns details in OMS system.

## System interactions

System interactions can be seen in the diagram below:



**Remarks**

Return create order can be skipped when communicating with OMS. In such case return case will be automatically created when OMS will get first callback about return management step.

## Solution overview

Solution from Optimus platform point of view should:

* Present to CSAgent key information required to perform a refund (i.e. RMA number which was delivered to the customer with each order)
* Retrieve from CTS (via push mechanism utilised by ERP system) and store in OMS DB
* Allow refund - trigger manual process
* Present end-customer information on return status (eCommerce - SelfCare)

Solution will be mainly developed in OMS.

# Implementation

## Order provisioning with RMA number

Key return information that should be available in OMS against an order is RMA number.

This can be retrieved by already developed CheckOrderStatus (ESOM) service. RMA number should be available from RMA when order status is changed to Picked and Shipped (so.status -> DISPATCHED)

Data is available as per fragment of XML response below:

…

<cac:AdditionalProperty>

<cbc:Name>**RMANumber**</cbc:Name>

<cbc:Value>**TestInv\_009**</cbc:Value>

</cac:AdditionalProperty>

…

To do:

1. This data should be extracted and RMA number should be stored in order property: RMA\_NUMBER of order property type RETURN. To simplify the solution this should be part of the DISPATCH status handler.
2. RMA number should be presented in OMS on Order Details pop-up windows (new tab Returns)

**Remarks**

1. RMA is provided per order but can be used for several returns for all products included in order. Return of each item is traced by OMS separately.
2. Returns with the same RMA can be distinguished by difference in **ProductCode** and in **return\_id.**
3. RMA stored within order structure is used only as information to CSAgent (e.g. for communication with customer). Returns are processed independently and are using RMA number provided by CTS system.

## Returns process

Once the return case is created there will be a push notification from the UBL sub system to inform OMS about new return case and return case changes. Push will be done via ESOM interface.

### Returns Update Webservice

ESOM will expose webservice (WS SOAP) and perform XML parsing but data storage and handling will be done in OMS:

CTS 🡪 WS 🡪 ESOM (log & maps to objects) 🡪 OMS (OMS Returns Model update)

1. to be called (by ERP/CTS system) to push returns status updates.
2. on each call provided XML (returnsXML) may contain updates for:
   1. single or multiple orders – as separate Items
   2. single or multiple products – as separate Items
3. if several changes are made for the same product (i.e. the same order, product and return\_id) items should be placed in correct order (i.e. chronologically older first).

#### Web service returnsUpdate (ESOM)

Boolean returnsUpdate(returnsXML string)

**Remark**

Returned Boolean value should be used by Tetra push mechanism to handle failure/retry attempts.

##### Example of returnsXML parameter data

This is an example XML of existing webservices.

<Catalogue [namespace definition removed for clarity]>

<cbc:ID>RETURNS</cbc:ID>

<cbc:Name>RETURNS</cbc:Name>

<cbc:IssueDate>2016-06-29</cbc:IssueDate>

<cac:ProviderParty />

<cac:ReceiverParty />

<cac:ContractorCustomerParty>

<cbc:CustomerAssignedAccountID>**expSamsung**</cbc:CustomerAssignedAccountID>

<cbc:SupplierAssignedAccountID>DS</cbc:SupplierAssignedAccountID>

<cbc:AdditionalAccountID>**ex3s1ms0ng**</cbc:AdditionalAccountID>

</cac:ContractorCustomerParty>

<cac:CatalogueLine>

<cbc:ID>MB-MP16DA/EU</cbc:ID>

<cac:ItemComparison>

<cbc:PriceAmount currencyID="GBP">0</cbc:PriceAmount>

<cbc:Quantity>1</cbc:Quantity>

</cac:ItemComparison>

<cac:Item>

<cac:AdditionalItemProperty>

<cbc:Name>**OrderNo**</cbc:Name>

<cbc:Value>I600442</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**CustomerOrderReference**</cbc:Name>

<cbc:Value>66KU5523AFA2</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**ProductCode**</cbc:Name>

<cbc:Value>MB-MP16DA/EU</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**Quantity**</cbc:Name>

<cbc:Value>1</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**Serial**</cbc:Name>

<cbc:Value/>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**Reason**</cbc:Name>

<cbc:Value>14 Day</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**Comments**</cbc:Name>

<cbc:Value>Customer change of mind.</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**RANumber**</cbc:Name>

<cbc:Value>RMA-I600442-529107\_7325</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**Grading**</cbc:Name>

<cbc:Value>5</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>Customer</cbc:Name>

<cbc:Value>14771S</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**Status**</cbc:Name>

<cbc:Value>201</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**StatusDesc**</cbc:Name>

<cbc:Value>Send Response Line Completed</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**return\_id**</cbc:Name>

<cbc:Value>7325</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>account\_code</cbc:Name>

<cbc:Value>14771S</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**status\_code**</cbc:Name>

<cbc:Value>201</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>statusDesc</cbc:Name>

<cbc:Value>Send Response Line Completed</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**Date\_Issued**</cbc:Name>

<cbc:Value>2016-06-29 17:27:26.61</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**Date\_Received**</cbc:Name>

<cbc:Value>2016-06-29 19:27:01.0</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**Date\_Accepted**</cbc:Name>

<cbc:Value>2016-06-29 19:27:01.0</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>Product\_nett\_val</cbc:Name>

<cbc:Value>0</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>NetValue</cbc:Name>

<cbc:Value>0</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>bin\_location</cbc:Name>

<cbc:Value>RET-A</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**pack\_grade**</cbc:Name>

<cbc:Value>5</cbc:Value>

</cac:AdditionalItemProperty>

<cac:AdditionalItemProperty>

<cbc:Name>**customer\_wish**</cbc:Name>

<cbc:Value>**Refund**</cbc:Value>

</cac:AdditionalItemProperty>

</cac:Item>

</cac:CatalogueLine>

… (multiple lines may occur)

</Catalogue>

ESOM implementation should:

1. log info about call to WS in log.orderManagement
2. Parse XML
   1. Convert XML into transport object if no errors
   2. Return false if errors
3. Call synchronously OMS WS (returnsUpdate)
4. Return status returned by OMS

OMS implementation is responsible for business logic implementation and updating DB model:

Key information that we should extract from above XML and store in returns.product table:

* **OrderNo** (in Tetra)
* **CustomerOrderReference** (in OMS)
* **RANumber** (RMA Number )
* **ProductCode** (may be multiple products)
* **return\_id** (may be multiple returns for the same products and RMA)
* **Grading** (Product Grading)
* **pack\_grade** (Packaging Grading)
* **Reason** (Returns Reason )
* **Status** - Returns CTS Status
  + CTS Status will be converted to return.status
* **StatusDesc** (Detailed description)
* **customer\_wish** (possible values: refund / replace / credit / repair)
* **Comments**

All the information has been highlighted in above XML.

Return case will be assigned to OMS order according to **CustomerOrderReference** value**.**

RANumber extracted from CTS data should be equal to RMANumber retrieved from ERP system when order was dispatched (see Order provisioning with RMA number above). However, equality between those numbers is not checked nor reinforced.

**Remark**

1. Return reason dictionary (see HLD) should be expanded to cover “FAILED DELIVERY” case.
2. In XML CTS status is return twice as Status and status\_code -> currently during parsing status\_code will be ignored

##### ReturnUpdate webservice functionality

1. Parse provided xml file (implemented in ESOM).
   1. In the case of error return false and stop processing;
   2. If all necessary information (fields marked with bold in the example above) can be extracted return true;
   3. Check CustomerAssignedAccountID and AdditionalAccountID to be equal **expSamsung** and **ex3s1ms0ng** respectively.
      1. The above values should be site specific and configurable.
      2. Return false if not equal and stop processing.
2. Call OMS return update
3. Process info from XML for each item in sequence (implemented in OMS).
   1. Check CTS status – if maps to RETURNS\_ERROR abandon and return false
   2. For OMS order check if return.case record with matching **return\_id exist**
      1. If not exists – create new returns.case record and set parameters
      2. check if returns.product linked with returns.case and matching ProductCode exist
      3. if not create and link with corresponding return.case record
   3. Store retrieved data for corresponding item in the database using
      1. **CustomerOrderReference** (OMS order number),
      2. **RANumber** (RMA)
      3. **ProductCode** as a indexing key
   4. For each returns.product
      1. set/update correct product status - see return.status table
      2. update needRefund field
4. For all return.case linked with updated returns.products update return.case status accordingly to statuses of return.products linked to this case (implemented in OMS)
   * 1. See return.status table
     2. update needRefund field
5. Await further updates from the CTS system/OMS refunds until refund.product reaches status completed.
6. Awaits for all returns.product linked with returns.case to reach status RETURN\_PRODUCT\_COMPLETED than update returns.case status to RETURN\_CASE\_COMPLETED if no refunds is need or RETURNS\_CASE\_PENDING\_REFUND otherwise (implemented in OMS).
7. If return.case filed needRefunds was set to true (implemented in OMS)
   1. Update and order/product information in UI
   2. trigger refund process
   3. Awaits finalisation of manual refund process (with statues RETURNS\_CASE\_REFUNDED or RETURNS\_CASE\_REFUND\_FAILED

**Remarks**

1. Return case is for future usage. Currently each returned product results in creation of new return case.
2. After return case is accepted CSAgent can perform a refund for all accepted products. See description below

### Database

Following objects will be required in OMS database:

return.type – for future expansion of return types

return.case – linked to order, there can be multiple cases per on sales order

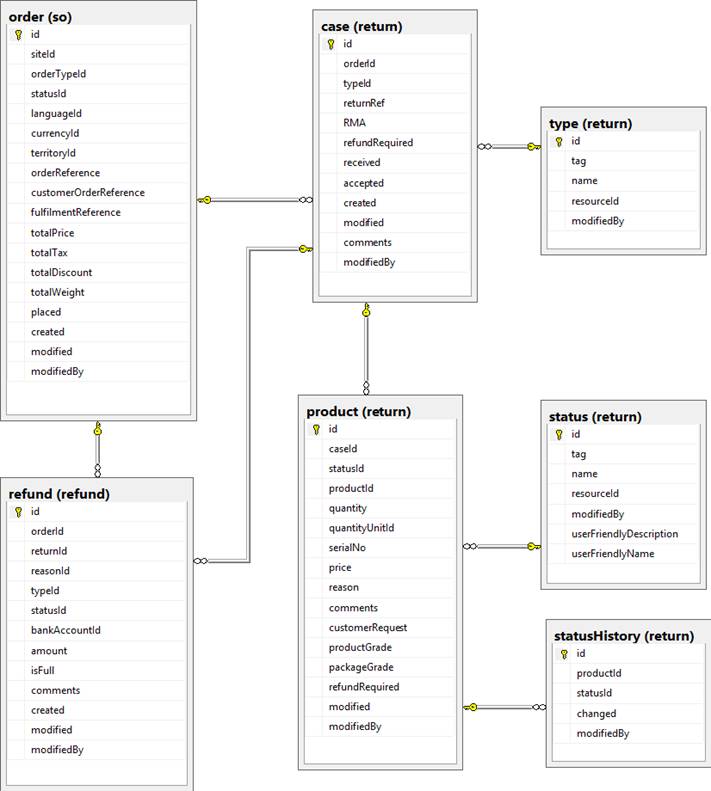
return.product – linked to case (the same product can be return more than once e.g. in the case of repairs)

return.status – dictionary table with possible values for return.case status

return.statusHistory – linked to return.product

return.caseHistory – linked to return.case

Relation diagram for returns, orders and refunds:



In table details bellow colums highlighted bold are extracted from CTS XML data.

#### Table return.status

Return return.product.column statusId should be updated accordingly to following statuses

**Columns**:

Id

tag (see below)

|  |  |  |
| --- | --- | --- |
| **Status Tag** | **CTS Status ID** | **Comments** |
| RETURNS\_PRODUCT\_CREATED | < 120 (?) | Customer raised a case or sent the product to DC |
| RETURNS\_PRODUCT\_ACCEPTED | = 120 | Returned product checked and return accepted  Set return.product field **refundRequired** to **true** when this status is set |
| RETURNS\_PRODUCT\_REPAIR | = 130 | Returned product checked, return product to be repaired and send back |
| RETURNS\_PRODUCT\_REJECTED | CTS status 115 or 999 | Future usage only |
| RETURNS\_PRODUCT\_COMPLETED | CTS status = 201 | Will trigger processing return case e.g. refund |
| RETURNS\_ERROR | All other statuses not listed above | Return false and abandon processing |

**Remarks**

1. Currently all returns will be process as accepted (code 120)
2. Failed delivery is currently process as return
3. Need for refund will be judged by CSAgent on the base of return reason
4. Call to WebService returnsUpdate can be done initially with CTS code >=120. In this case returns.product record should be created automatically and status RETURNS\_PRODUCT\_CREATED logged into DB

Return return.case column statusId should be updated accordingly to following statuses

|  |  |  |
| --- | --- | --- |
| **Status Tag** | **CTS Status ID** | **Comments** |
| RETURNS\_CASE\_CREATED | New case created | Customer raised a case or sent the product to DC. Products should be processed. |
| RETURNS\_CASE\_PENDING\_REFUND | If any product status is RETURNS\_PRODUCT\_ACCEPTED and all others products are in status != RETURNS\_PRODUCT\_CREATED | Processing of all products is finalized and return of at least of one product was accepted i.e. at least one return.product of this case have **refundRequired** field set to true. |
| RETURNS\_CASE\_REFUNDED | When successful return from refund process | This will only be logged in to returns history. Will be automatically changed to RETURN\_COMPLETED |
| RETURNS\_CASE\_REFUND\_FAILED | When refund is impossible or was cancelled | This will only be logged in to returns history. Will be automatically changed to RETURN\_COMPLETED |
| RETURNS\_CASE\_REJECTED | If all products are in status RETURNS\_PRODUCT\_REJECTED | No refund process should be triggered |
| RETURNS\_CASE\_COMPLETED |  |  |

**Remarks**

1. Return case should be created automatically when new returns.product unrelated to any existing return.case is created.

#### Table return.case

All unique return cases pushed from CTS are added into this table. Indexed by **OrderNo** (CTS reference).

Single case may refer to multiple returned products.

**Columns**:

**returnRef (OrderNo** – unique Tetra reference from first product)

**RMA (RANumber** from first/any product**)**

**created (Date\_Issued in XML from first product)**

**received (Date\_Received in XML from first product)**

**accepted (Date\_Accepted in XML from first product)**

refundRequired (default = false)

statusId (link to return.status)

Comments (can be manually added in OMS – not extracted from CTS XML, This comment will be visible to customer on tracking page)

#### Table return.product

Multiple returned products can be linked to single return.case

Columns:

Id (unique)

caseId (link to return.case)

**productId (ProductCode)**

**quantity (from XML Quantity – should be 1)**

**quantityUnirtId (should correspond to Unit)**

**serialNo (from XMl Serial – may be null)**

**productGrade** (Product Grading)

**packageGrade** (Packaging Grading)

**reason** (Returns Reason )

**customerRequest (from XML customer\_wish** (possible values: refund / replace / credit / repair) )

statusId (link to return.status)

refundRequired (default = false)

**comments** (should be extracted from CTS XML data but can be manually edited/added in OMS)

#### Table return.statusHistory

History of return processing for single product

Id (unique)

productId (link to return.product)

changed (timestamp)

statusId (return status from return.status)

#### Table return.caseHistory

History of return/refund processing for single return case

Id (unique)

caseId (link to return.case)

changed (timestamp)

statusId (return status from return.status)

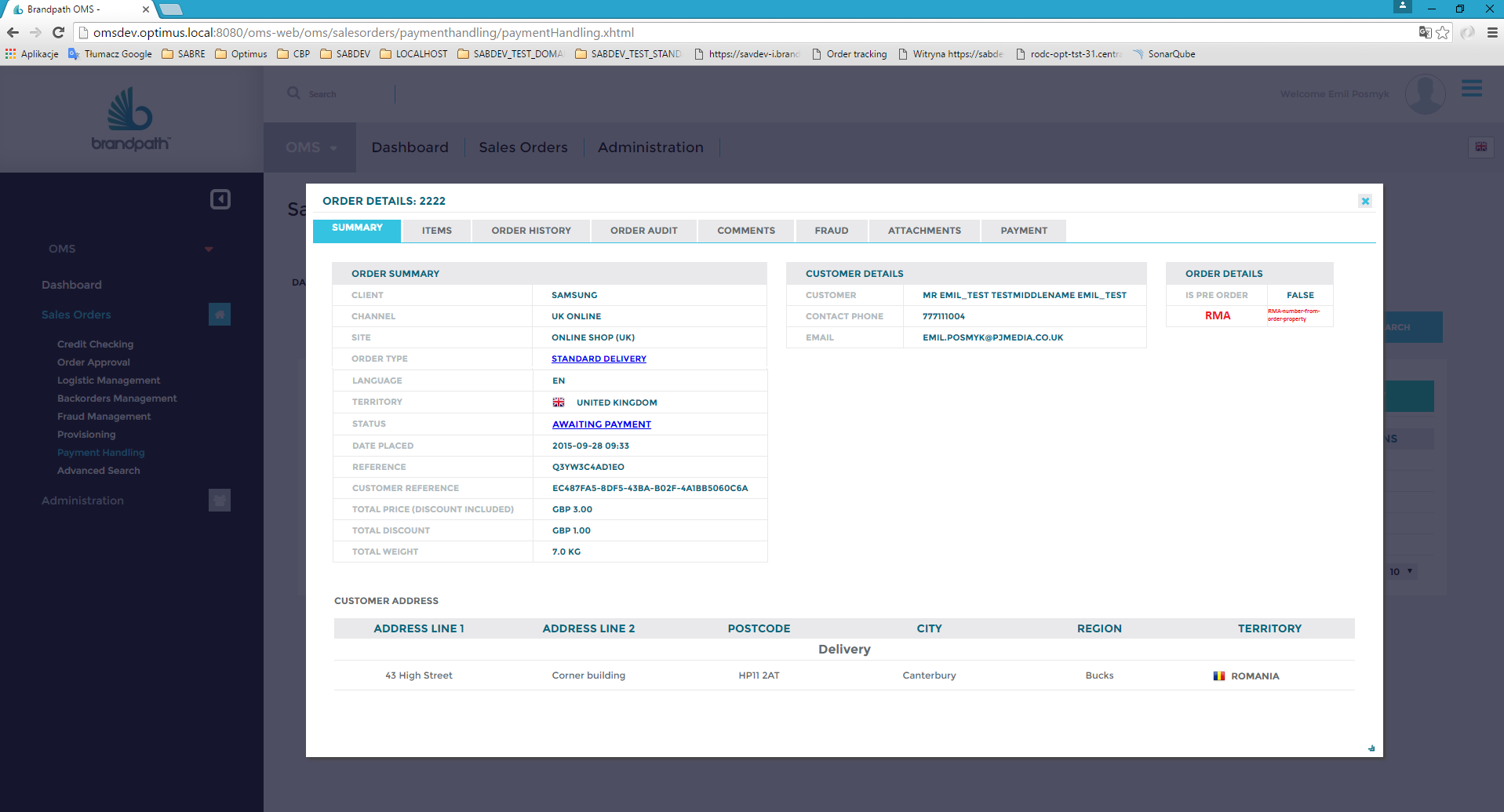
## UI Changes

In order details UI

#### Modify: Order details - tab

Add into order details info about

* Return RMA number – which was send to customer with order delivery (data source order property: RMA\_NUMBER of order property type: RETURN).

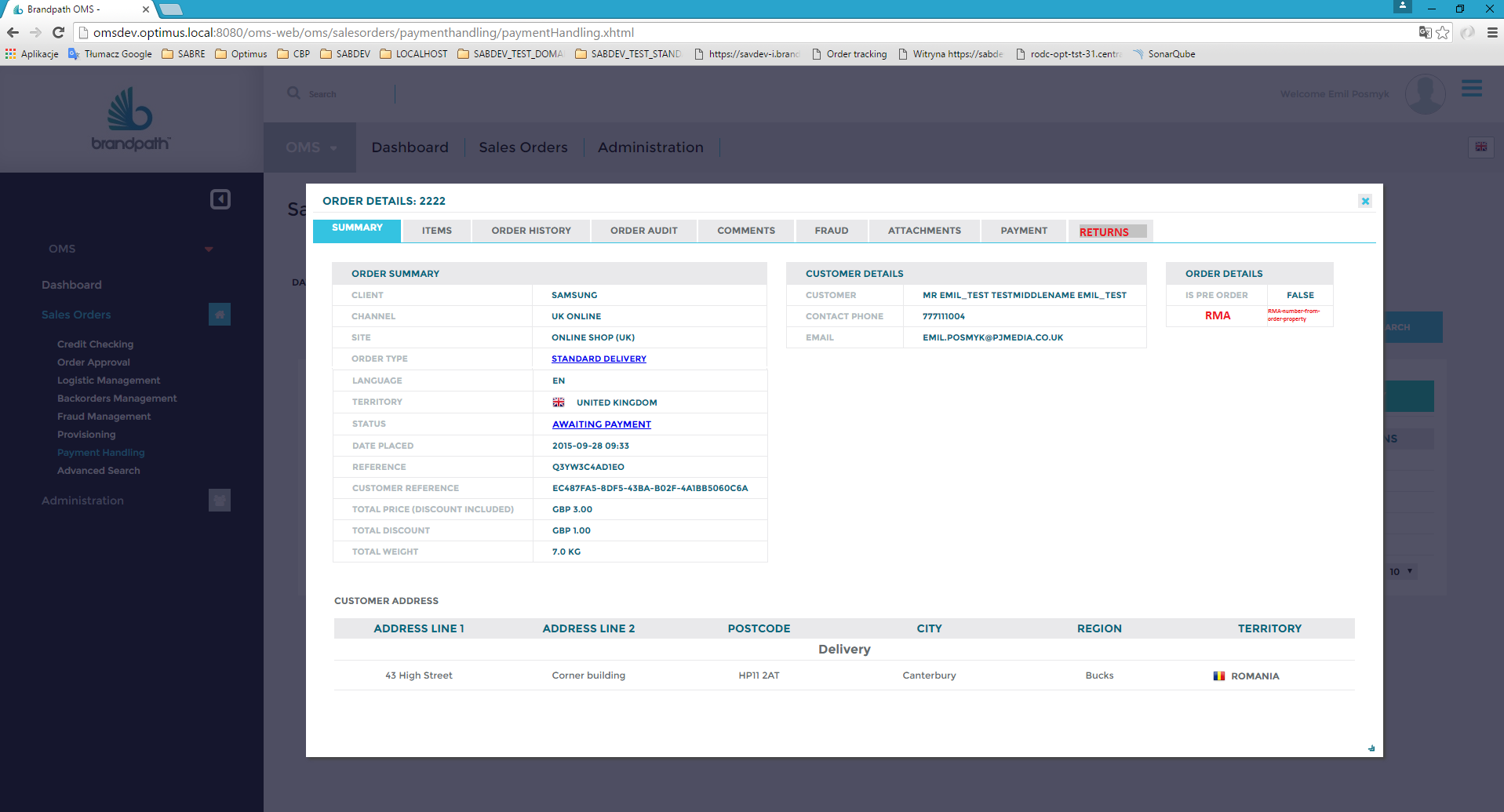


Changes marked in red

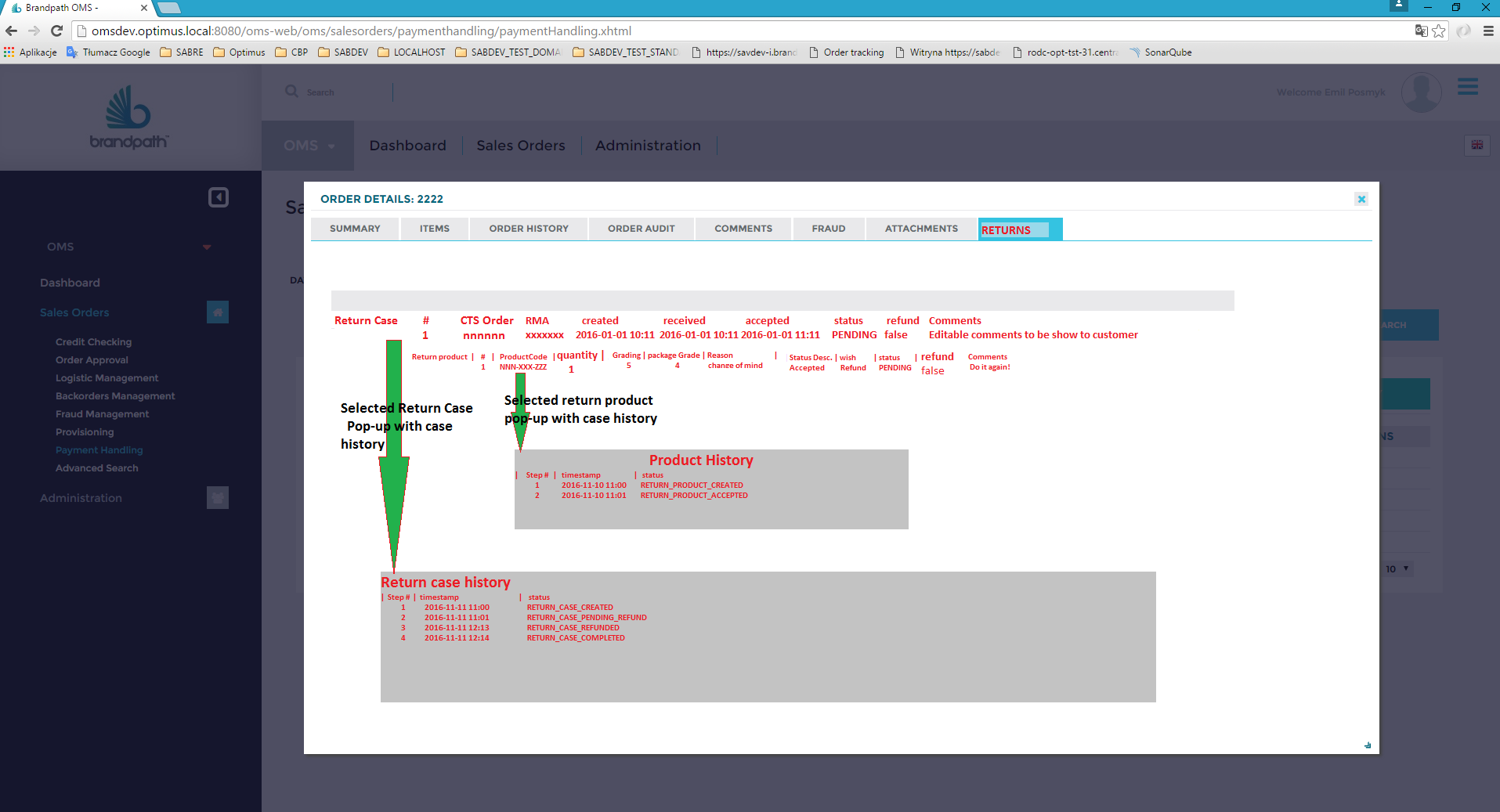
Layout will be optimised i.e. the ritght column ORDER DETAILS will be placed under middle one.

Under RMA data status of returns will be displayed – to indicate that there are returns and user can see returns detail under tab Returns (this tab should be always visible – even where there are no returns)

#### Create new tab RETURNS



Example of tab Returns:



Details

* Return Cases (linked to this order – if any)
  + Returned Products (linked to return case)
    - Return product History (in a popup window)
  + Current status of return case

If there are no returns – info that there is no returns should be presented under Return tab. Tab Returns should be always accessible.

### Information to be presented:

#### For return case:

Id

returnRef (Tetra orderNo)

RMA

created (date)

received (date)

accepted (date)

userFriendlyName (from return.status corresponding to statusId)

refundRequired

Comments

#### For returned product

Id

**productId**

**serialNo**

**quantity**

**productGrade** (Product Grading)

**packageGrade** (Packaging Grading)

**reason** (Returns Reason )

**StatusDesc** (last Detailed description)

**customerRequest** (possible values: refund / replace / credit / repair)

userFriendlyName (from return.status corresponding to statusId)

refundRequired

Comments (can be manually added in OMS)

##### Action buttons

For each return case should be refund action button -> should link to refund page (it will only switch tabs – for now)

For each return product should be refund action button -> should link to refund page (it will only switch tabs – for now)

#### History pop-up windows

##### For selected returned product

In a pop-up windows for each product

List of status change steps

Id (unique)

timestamp

status (return status from return.status)

##### Not to be implemented - For selected returned case

In a pop-up windows for each product

List of status change steps

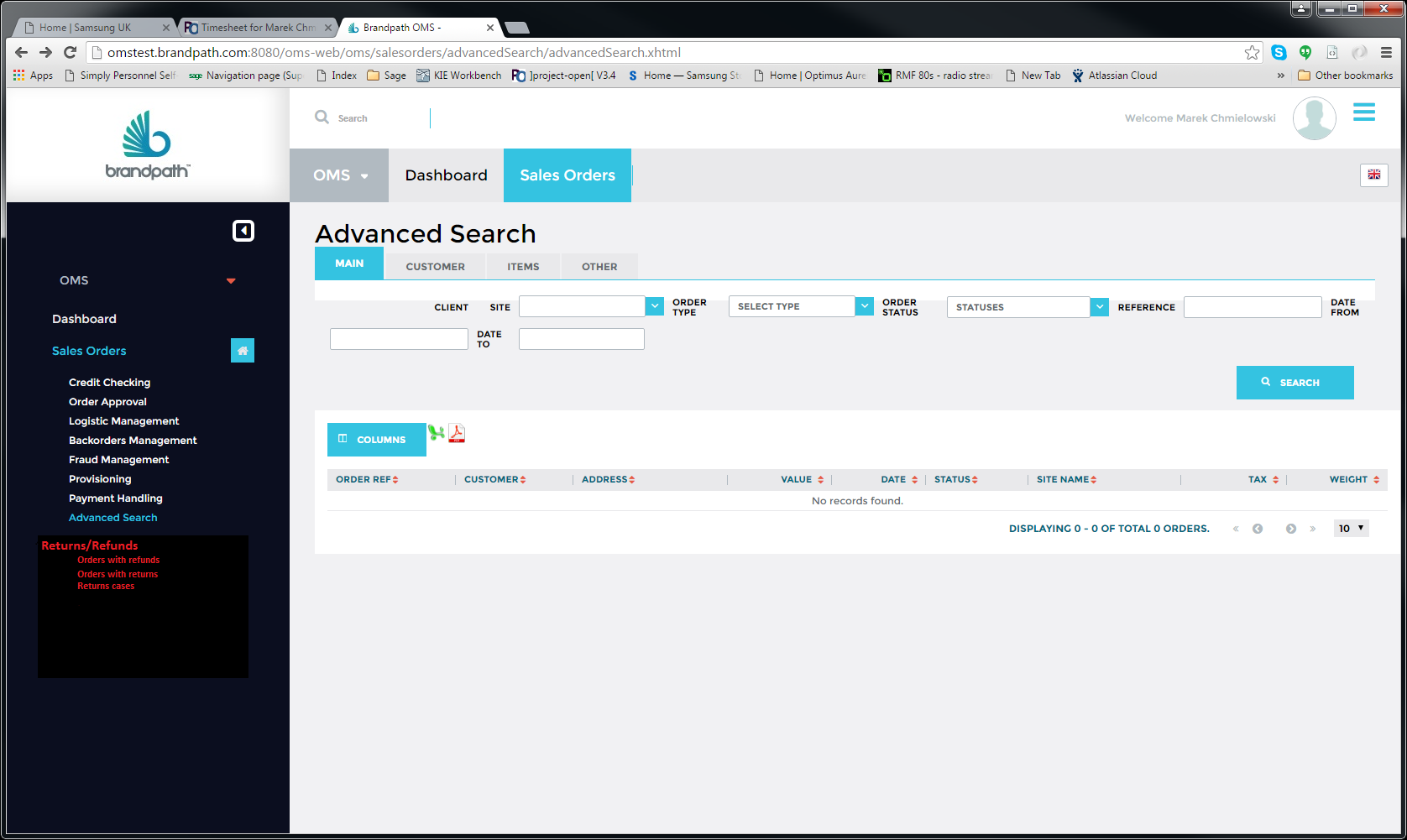
Id (unique)

timestamp

status (return status from return.status)

### Main menu modifications

Left menu – additional positions (marked in red):



Advance Search

Search for orders with given RMA number should be added – orders should be selected independently if there are returns associated with given RMA number.

Search should look for matching order property RMANUMBER

Returns/Refunds

Sub-menu: orders with refunds

List (link to) of all orders with pending refunds (i.e. returns filtered with default filter: status != COMPLETED)

Sub-menu: Orders with returns

List (link to) of all orders with pending returns (i.e. returns filtered with default filter: status != COMPLETED)

Sub-menu: Return cases

List of all pending return case (i.e. return cases filtered with default filter: status != COMPLETED) – table of return cases (with info as for each return in order details) + link to order (order number)

For each return case sub-table of return products

Search capabilities should be expanded for refund and returns.

Search by:

Return Status

Refund Status

RMA number

Preserve existing search options i.e. Date, client,…..

Remark

Search for RMA number here should be restricted to orders with returns/refunds. Search should be done on return.case field RMA

## Update Order Tracking Service

Order self-tracking page is available for customer via unique link provided on confirmation page in inside confirmation e-mail (see HLD\_Tech – Optimus Sabre 2.0 eCommerce – fig. 3). Data for order tracking page are provided by Sabre 2.0 web service:

package com.brandpath.sabre.uk.site.service.rest.cart

interface ICartRS

method: getOrderStatus

In turn, data for Sabre are provided by OMS webService:

package com.brandpath.oms.ws.sei

interface ISalesOrderStatus

method: getOrderStatus

implementation is in:

package com.brandpath.oms.ws.so

class SalesOrderStatus

method: getOrderStatus

The above method should be updated to provide information about returns/refund status (if available). All other order information currently presented should be unchanged.

For returns the following information should be provided:

#### For return case:

Id

returnRef (Tetra orderNo)

RMA

created (date)

received (date)

accepted (date)

userFriendlyName (from return.status corresponding to statusId)

refundRequired

Comments

#### For returned product

Id

**productId**

**serialNo**

**quantity**

**productGrade** (Product Grading)

**packageGrade** (Packaging Grading)

**reason** (Returns Reason )

**StatusDesc** (last Detailed description)

**customerRequest** (possible values: refund / replace / credit / repair)

userFriendlyName (from return.status corresponding to statusId)

refundRequired

Comments (can be manually added in OMS)

#### For each refund connected with order in question:

Date

Amount

Status

reason

comments

**Remarks**

1. In general refunds can be unrelated to returns.
2. However, return can trigger refund
3. This SDD does not cover how front-end should presented visually information about returns/refunds – only how ensure that relevant information about returns/refunds are available on demand from front-end application

## Reporting

Returns related reporting are handled entirely by Tetra. Two report are generated:





According to reporting workshop summary : there is no need for changes.