

Single View Of All Orders

Technical Design Documentation

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Updated By | Date | Comments |
| 0.1 | Amit Mandal | 27/04/2015 | Initial Draft |
|  |  |  |  |
|  |  |  |  |

Document Sign-Off

|  |  |  |
| --- | --- | --- |
| Name | Role | Signature/Date |
| Pascal ESPINOUSE |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[1. Introduction 3](#_Toc418679185)

[1.1 Purpose 3](#_Toc418679186)

[1.2 Intended Audience 3](#_Toc418679187)

[1.3 Document Scope 3](#_Toc418679188)

[1.4 Functional Rules 3](#_Toc418679189)

[1.5 Assumptions/Restrictions 4](#_Toc418679190)

[1.6 Documents Structure 4](#_Toc418679191)

[1.7 Additional Resources 4](#_Toc418679192)

[2. High Level Design Diagram 5](#_Toc418679193)

[2.1 Overview 5](#_Toc418679194)

[2.2 Sequence Diagram 6](#_Toc418679195)

[2.3 Use Case Diagram 8](#_Toc418679196)

[3. Detailed Application Design 9](#_Toc418679197)

[3.1 Class Diagram 9](#_Toc418679198)

[3.2 Beans.xml 9](#_Toc418679199)

[3.3 Facade 10](#_Toc418679200)

[3.4 Strategy 10](#_Toc418679201)

[3.5 Populators 10](#_Toc418679202)

[3.6 Controller 10](#_Toc418679203)

[3.7 View/JSP/Tag 10](#_Toc418679204)

[3.8 Localization 10](#_Toc418679205)

[3.9 Beans.xml 11](#_Toc418679206)

[4. Database architecture 12](#_Toc418679207)

[4.1 ER Diagram 12](#_Toc418679208)

[5. Installation and Configuration 13](#_Toc418679209)

[5.1 Installation steps for the addon 13](#_Toc418679210)

[5.2 Configuration steps for the addon 13](#_Toc418679211)

[Open accountOrderHistoryPage.jsp – Use ${ order} variable to show added columns by populators(ERP number, author etc.) 13](#_Toc418679212)

[7. Appendix A *– Glossary of terms* 14](#_Toc418679213)

1. Introduction

Purpose

This document is the Technical Design Document which provides information of singlevieworders addon and steps to configure the module. It covers:

1. High Level Design
2. Application Architecture
3. Database Architecture
4. Installation Steps

Intended Audience

This document is aimed at Hybris developers and Technical members who want to implement the singlevieworders addon in Hybris.

Document Scope

This document covers the details about the sprint 1 hot folder addon.

Functional Rules

* Order history List Page – this will show all the orders of a B2B unit for all the users including offline orders. Order History should contain at least Columns like: Order Number, ERP Order Number, Origin, Order Status, P.O Number, Date Placed, Author, Actions. Also, Orders list can be sorted by Date, Order Status, Order Number, P.O. Number, ERP Order Number
* Order History Detail Page – This will show all the order details and below each order entry there will be tracking details (if there is any). Tracking details will have following details:

In the section Your Order, for each order entry, the table shows:

* Order Quantity
* Shipped Quantity
* Open Quantity: Quantity that has not been delivered yet
* Expected Release Date
* Delivery Status
* Delivery Method

The *Shipped* table shows line by line:

* + Date: Delivery Date
  + Quantity: Quantity delivered on this date
  + Tracking Number: Clickable, linked to the third party carrier platform.
* Order History Detail page – If Product is offline i.e only available in master catalog then product will not be clickable and cannot be bought on site.
* When clicking on the Reorder button, the customer will be redirected to the cart page. In the case that not all the products are available in the local catalog, a message should be displayed in the cart page: Some products of the historical order are not available online.

Assumptions/Restrictions

This module has been created for version 5.4.0.0.

**Display:** Changes will show in My Account > Order History Link

**Import**: The different template we provide in CSV files filled, these will be files where offline orders will be imported.

**Export**: No export is available

Documents Structure

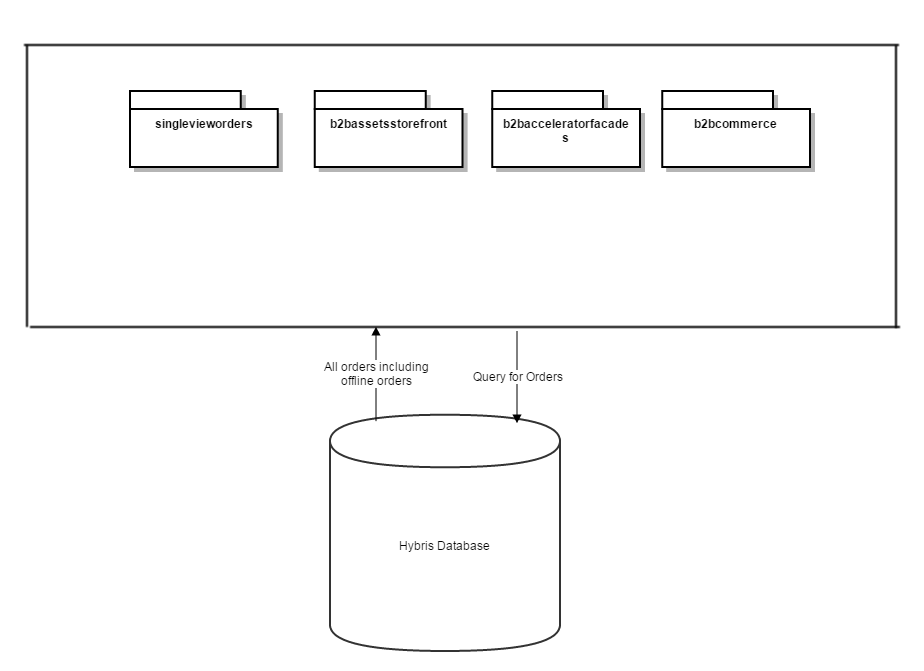
|  |  |  |
| --- | --- | --- |
| Section | | Description |
| 1 | Introduction | Describes the purpose and intended audience of the document and explains its structure. |
| 2 | High Level Design | Overview of the module. |
| 3 | Detailed Application Design | Application architecture applicable to this module. |
| 4 | Detailed Database Design | Conceptual and logical architecture applicable to this module. |
| 5 | Installation & Configuration | Steps to install and configure the module. |
| 6 | Appendices | Supporting information. |

Additional Resources

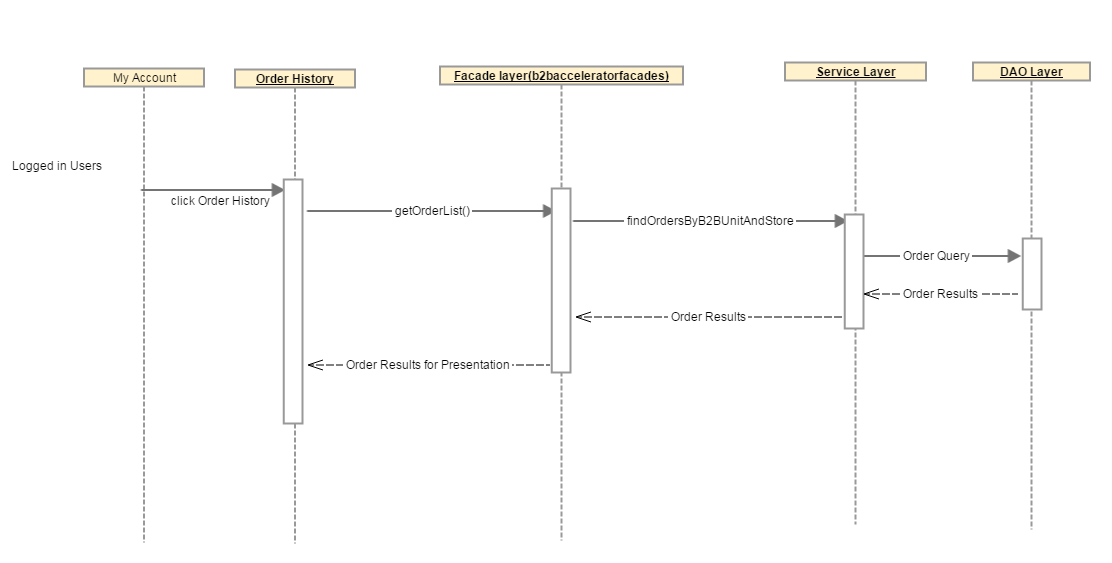
|  |  |  |  |
| --- | --- | --- | --- |
|  | Description | URL | Purpose |
| 1 | Wiki Portal | <https://wiki.hybris.com> | To gather more information |
| 2 | Coconet | https://coconet2.capgemini.com/sf/go/doc3384622?nav=1 | To refer functional specifications |

1. High Level Design Diagram

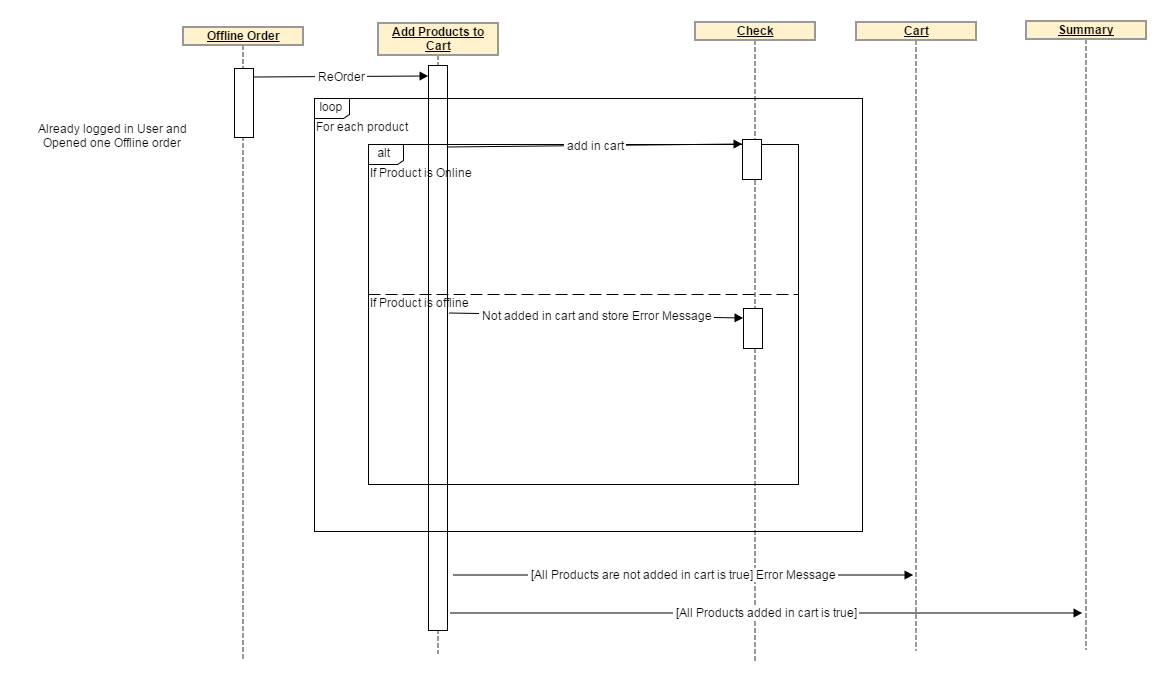
Overview



Sequence Diagram

Sequence of events during Getting All the Orders for a B2BUnit

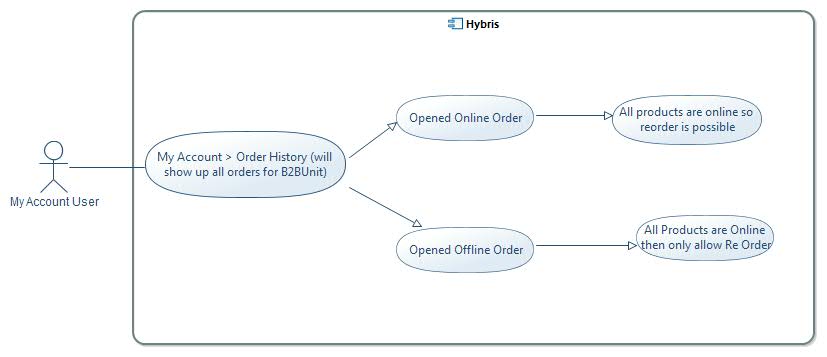
Sequence of events during Reorder



|  |  |  |
| --- | --- | --- |
| **#** | **Component** | **Description** |
| 1 | **My Account** | My Account Home Page after login |
| 2 | **Order History** | List view of all orders of the B2B Unit (including offline orders) |
| 3 | **Façade Layer (b2bacceleratorfacades)** | Hybris b2b façade layer which has access to service layer |
| 4 | **Service Layer** | Hybris Service Layer (here we are using customer service) |
| 5 | **DAO Layer** | Customer Account DAO layer which will run database query to find all the orders for all the users of B2B unit |
| 6 | **Offline Order** | B2B customer logged in and opened any offline order placed by any user of the same B2BUnit |
| 7 | **Add Products to Cart** | This event will add products to the cart |
| 8 | **Check** | This will check that product is available online i.e. product is available in Online version of catalog or not? |
| 9 | **Cart** | Shopping Cart (if it lands down in above sequence diagram then it will render with error messages that product/s were not available online) |
| 10 | **Summary** | If products are added in cart successfully then it will land on summary page. |

Use Case Diagram

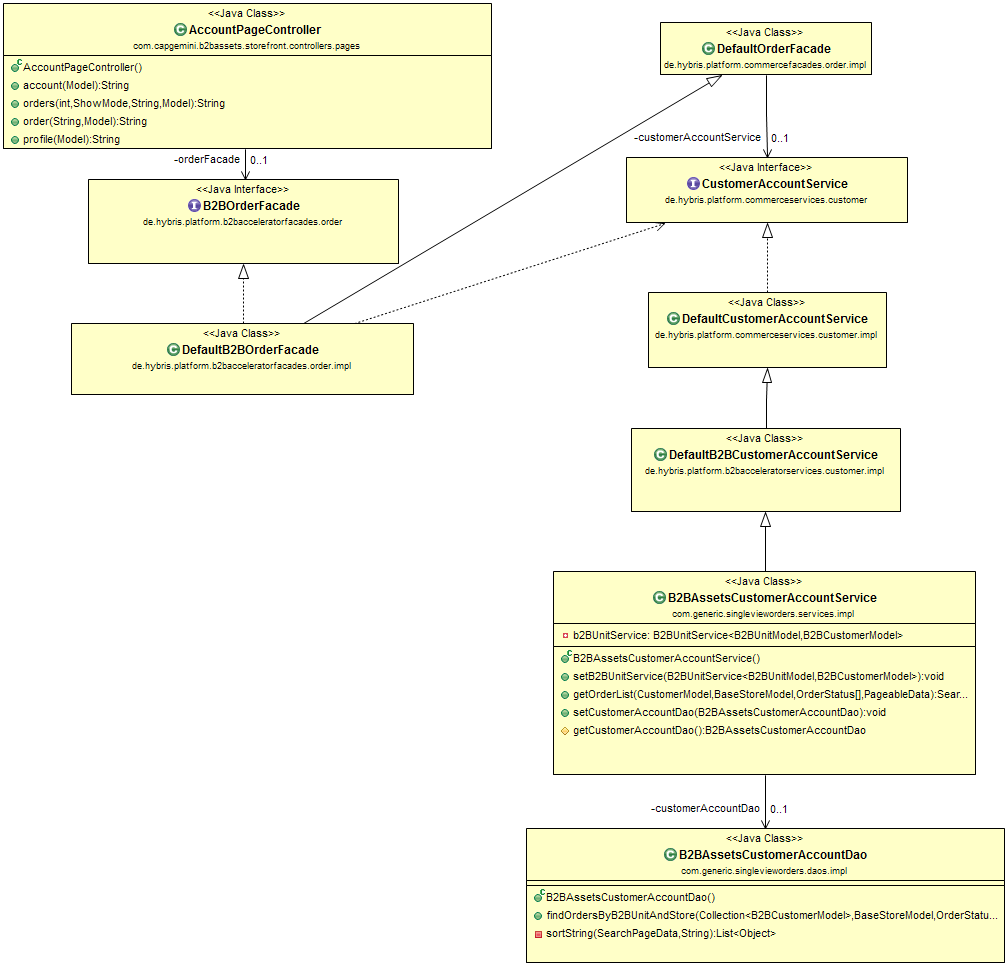
Use case diagram for the hot folder singlevieworders addon



**Description:** In Order History all orders will show up of the user’s parent B2BUnit. Reorder is always possible for the orders placed on Hybris. Reorder of offline orders will only be possible for online products.

1. Detailed Application Design

Class Diagram



Beans.xml

**AccountPageController:** orders() function will fetch all the orders (placed on Hybris and non-Hybris orders) and will send to JSP for rendering

**DefaultOrderFacade:** There is no change in this class just to show dependency here.

**B2BOrderFacade:** No change in getPagedOrderHistoryForStatuses(….)

**DefaultB2BOrderFacade:** No change in getPagedOrderHistoryForStatuses(….)

**CustomerAccountService:** There is no change in this class just to show the relation of call getOrderList(…) function.

**DefaultCustomerAccountService:** There is no change in this class just to show the dependency.

**DefaultB2BCustomerAccountService:** There is no change in this class just to show the dependency.

**B2BAssetsCustomerAccountService:** This class is overriding getOrderList(…) function so that it can fetch all the users of B2BUnit using b2BUnitService

**B2BAssetsCustomerAccountDAO:** findOrdersByB2BUnitAndStore – new function which will fetch all the orders for the given users which belongs to a B2BUnit.

Facade

N.A

Strategy

N.A

Populators

B2BOrderEntryPopulator

1. Populating order tracking data at each Order Entry level which was imported using order tracking interface. For details please Order Tracking TDD.
2. Setting offline flag if product is not available in Online catalog so that product link will be disabled from Order Details page because PDP cannot be shown for that product since product is offline.

B2BAssetsB2BOrderHistoryPopulator – populating attribute ERPOrderNumber, Origin and Author.

Controller

N.A

View/JSP/Tag

orderDetailsItemWithTracking.tag – created separated tag to show order item details with tracking details and this tag is used instead of <order:orderDetailsItem tag in accountOrderPage.jsp.

accountOrderPage.jsp - orderDetailsItemWithTracking.tag, created separated tag to show order item details with tracking details and this tag is used instead of <order:orderDetailsItem tag in accountOrderPage.jsp.

accountOrderHistoryPage.jsp – updated table headers with ERP Number, Origin, Authors, etc. and populated values in respective <td> tags.

Localization

Following were added in add-on properties file:

# put localizations or messages used in your web application into this file

#

text.account.orderHistory.erpOrderNumber=ERP Order Number

text.account.orderHistory.origin=Origin

text.account.orderHistory.author=Author

#For Order Hisotry Page Sort By

text.account.orderHistory.page.sort.byStatus=Order Status

text.account.orderHistory.page.sort.byPONumber=PO Number

text.account.orderHistory.page.sort.byERPNumber=ERP Number

text.account.orderHistory.page.shipping.info=Shipping Information for **{0}**

text.account.orderHistory.page.scheduledQuantity=Scheduled Quantity

text.account.orderHistory.page.deliveredQuantity=Delivered Quantity

text.account.orderHistory.page.deliveryDate=Delivery Date

text.account.orderHistory.page.scheduledDate=Scheduled Date

text.account.orderHistory.page.trackingNumber=Tracking Number

text.account.orderHistory.page.trackinglink=<a href="http://www.fedex.com/Tracking?action=track&tracknumbers?**{0}**" target="new">**{1}**</a>

basket.page.message.update.reducedNumberOfItemsAdded.offline=**{0}** of the historical order are not available online

Beans.xml

Following beans added/modifed to show single view orders and used in above mentioned populators.

<bean class=*"de.hybris.platform.commercefacades.order.data.OrderHistoryData"*>

<property name=*"ERPOrderNumber"* type=*"String"*/>

<property name=*"origin"* type=*"String"*/>

<property name=*"author"* type=*"String"*/>

</bean>

<bean class=*"de.hybris.platform.commercefacades.order.data.OrderTrackingData"*>

<property name=*"scheduledDate"* type=*"java.util.Date"*/>

<property name=*"scheduledQty"* type=*"java.lang.Integer"*/>

<property name=*"deliveryDate"* type=*"java.util.Date"*/>

<property name=*"deliveryQty"* type=*"java.lang.Integer"*/>

<property name=*"trackingNumber"* type=*"java.lang.String"*/>

</bean>

<bean class=*"de.hybris.platform.commercefacades.order.data.OrderEntryData"*>

<property name=*"offlineProduct"* type=*"boolean"*/>

<property name=*"orderTracking"* type=*"java.util.List&lt;de.hybris.platform.commercefacades.order.data.OrderTrackingData>"*/>

</bean>

1. Database architecture

ER Diagram

**singlevieworders ER Diagram:**

**AbstractOrder**

| **Name** | **Description** | **Type** | **Default Value** | **Mandatory** | **Example** |
| --- | --- | --- | --- | --- | --- |
| ERPOrderNumber | ERP Order Number for offline orders | String |  | FALSE | 1234 |
| origin | Origin of the order – from phone, pos, fax etc. | String |  | FALSE | phone |

1. Installation and Configuration

Installation steps for the addon

Add the hotfolder module dependency in hybris/Assets/hybris/config/localextension.xml file as follows.

<extension name=" singlevieworders "/>

Next, Copy singlevieworders module directory into hybris/bin/custom.

To finish, execute the following commands marked in green using command prompt.

Cd hybris.directory/bin/platform

# **Note:** hybris.directory is the installation location of the hybris

ant addoninstall -Daddonnames="singlevieworders" -DaddonStorefront.yb2bacceleratorstorefront="<YOUR\_B2B2STOREFRONT\_EXTENSIONNAME>"

setantenv.bat

ant all

Configuration steps for the addon

Open accountOrderHistoryPage.jsp – Use ${ order} variable to show added columns by populators(ERP number, author etc.)

Open accountOrderPage.jsp – use this tag <orderTracking:orderDetailsItemWithTracking order="${orderData}" isOrderDetailsPage="true" /> to show tracking details.

7. Appendix A *– Glossary of terms*

|  |  |
| --- | --- |
| **Term/Abbreviation/Acronym** | **Meaning** |
| B2B | Business to Business |