

Quick Order module

Technical Design documentation

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Updated By | Date | Comments |
| 1.1 | Jayachandra G | 2/09/2015 | Initial Draft |
| 1.9 | Venkatesh Dasari | 3/05/2015 | Initial Draft |
| 2.0 | Jayachandra G | 4/08/2015 | Initial Draft |

Document Sign-Off

|  |  |  |
| --- | --- | --- |
| Name | Role | Signature/Date |
| Pascal ESPINOUSE |  |  |
| Aliasgar Muchhala |  |  |
| Gyaneshwar Dubey |  |  |

Table of Contents

[1. Introduction 3](#_Toc416258348)

[1.1 Purpose 3](#_Toc416258349)

[1.2 Intended Audience 3](#_Toc416258350)

[1.3 Document Scope 3](#_Toc416258351)

[1.4 Functional Rules 3](#_Toc416258352)

[1.5 Assumptions/Restrictions 3](#_Toc416258353)

[1.6 Documents Structure 4](#_Toc416258354)

[1.7 Additional Resources 4](#_Toc416258355)

[2. High Level Design Diagram 5](#_Toc416258356)

[2.1 Overview 5](#_Toc416258357)

[2.2 Quick Order Sequence Diagram 7](#_Toc416258358)

[2.3 CSV import sequence diagram. 8](#_Toc416258359)

[2.4 Use Case Diagram 10](#_Toc416258360)

[3. Detailed Application Design 11](#_Toc416258361)

[3.1 Class Diagram 11](#_Toc416258362)

[3.2 Beans 12](#_Toc416258363)

[3.3 Facades 12](#_Toc416258364)

[3.4 Strategy 12](#_Toc416258365)

[3.5 Controller 13](#_Toc416258366)

[3.6 View/JSP 13](#_Toc416258367)

[4. Database architecture 16](#_Toc416258368)

[4.1 ER Diagram 16](#_Toc416258369)

[5. Exception Handling and Logging 18](#_Toc416258370)

[5.1 Exception Scenarios 18](#_Toc416258371)

[5.2 Exception Handling: 18](#_Toc416258372)

[5.3 Logging 19](#_Toc416258373)

[6. Installation and Configuration 20](#_Toc416258374)

[6.1 Installation steps for the addon 20](#_Toc416258375)

[6.2 Configuration steps for the addon 20](#_Toc416258376)

[6.3 How to add a new column in JSP. 21](#_Toc416258377)

[7. Appendix A *– Glossary of terms* 22](#_Toc416258378)

1. Introduction

Purpose

This document is the Technical Design Document which provides information of quick order 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 quick order addon in Hybris.

Document Scope

This document covers the details about the sprint 1 quick order addon.

Functional Rules

* Quick order module allow user to add products to cart quickly with the SKU number and the quantity. The user can import a CSV file where each line contains the SKU and the quantity.
* Quick order module contains new JSP page and contains the columns like Code, Quantity, Delete and Add to cart button to place the order quickly using.
* Once the user enters a product code, quantity it search the automatically for the product availability and based on the availability customer can add the product to cart quickly to place the order.

Assumptions/Restrictions

This module has been created for version 5.4.0.0.

* If customer enters the SKU number and the quantity and clicks on add to cart then product will be added to cart.
* Import through CSV is also possible.
* If customer enters SKU number and quantity which is not available in system. Then error message has to display.
* The number of rows in a quick order screens can be reconfigured in an impex file to based on customer needs. It is defined 5 rows in **cms-content**.impex file for visibility.

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 | Exception Management & Logging | Information about the Exception handling and logging. |
| 6 | Installation & Configuration | Steps to install and configure the module. |
| 7 | Appendices | Supporting information. |

Additional Resources

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | URL | Purpose |
| 1 | Hybris Wiki | <https://wiki.hybris.com> | For more information on we can refer hybris wiki to understand the modules used in this assets |
| 2 | Capgemini coconet server | https://coconet2.capgemini.com/sf/go/doc3313609?nav=1 | Quick order document is maintained at the capgemini coconet server. |

1. High Level Design Diagram

Overview

Below is the High level overview of the quick order addon.

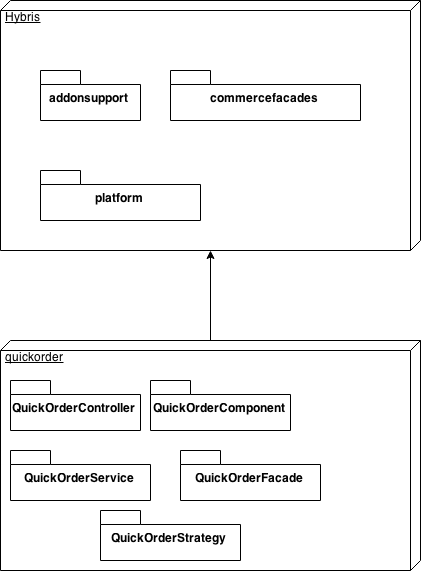


Fig: Quick Order High level design diagram

Description:

Quick order module provides the functionality to add products quickly to the cart. This module internally depends on the hybris provided extensions addonsupport, commercefacade and platform.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Extension/Component** | **Description** | |
| 1 | Addonsupport | | The hybris addonsupport extension provides a small collection of convenient classes for use when constructing own Accelerator AddOns like Quick order. |
| 2 | B2BCommerce | | The hybris b2bcommerce is an extension providing the main functionality of a hybris installation. |
| 3 | Quickorder Controller | | It is controller to handle the quick order. When customer/user enters a SKU number and the quantity this controller will be called to add the products to cart quickly. |
| 4 | Quick order Component: | | Quick order component is a CMS component it can be used to order quickly using the product SKU number and quantity. |
| 5 | Quick order Facade | | It is an interface it is used in Quick order controller to parse and find the user entered product SKU number and quantity. |
| 6 | Quick order Service | | It is an interface it contains two methods findProductsForQuery() and findProductForQuery() these methods uses the quickOrderSearchStrategy, QuickOrderDAO and ProductService and returns the List<ProductModel> and ProductModel respectively. |
| 7 | Quick order Strategy | | It is an interface used by the Default QuickOrder Service to find the user enterd products. |

Quick Order Sequence Diagram

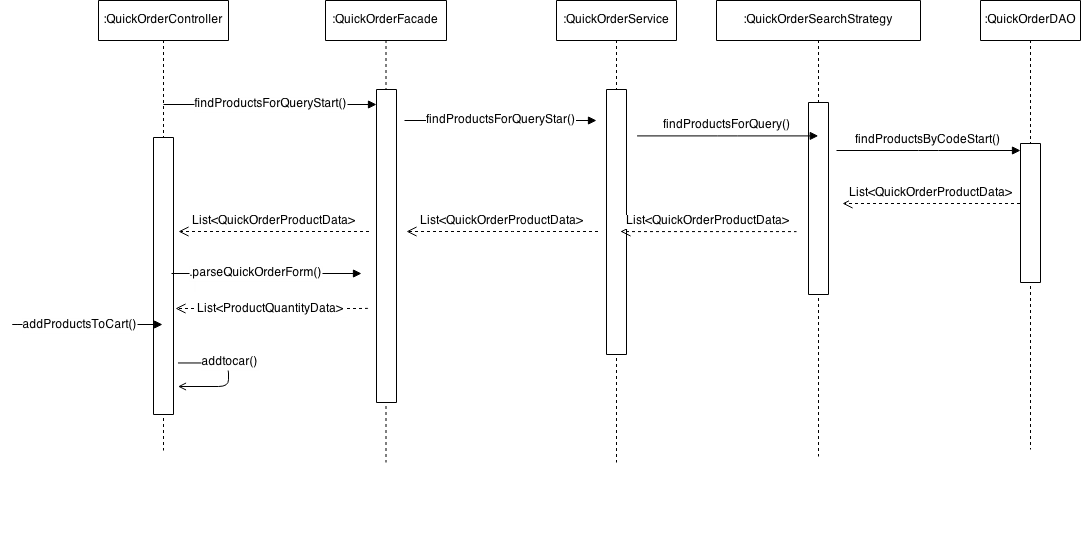


Fig: Quick order sequence diagram

Description:

|  |  |  |
| --- | --- | --- |
| **#** | **Component** | **Description** |
| **1** | QuickorderController | It is controller to handle the quick order. When customer/user enters a SKU number and the quantity this controller will be called to add the products to cart quickly. It contains the search() this will be called automatically and it will call internally indProductsForQueryStart() method. |
| **2** | QuickorderFacade | It is an interface it is used in Quick order controller to parse and find the user entered product SKU number and quantity. |
| **3** | QuickOrderService | It is an interface it contains two methods findProductsForQuery() and findProductForQuery() these methods uses the quickOrderSearchStrategy, QuickOrderDAO and ProductService and returns the List<ProductModel> and ProductModel respectively |
| **4** | **QuickOrderSearchStrategy** | This strategy is used for searh mechanism for the user entered SKU. It internally uses the quick order dao and product service for the product search. |
| **5** | **QuickOrderDAO** | Search in the database products beginning by a code entered by the B2B customer/user. |

CSV import sequence diagram.

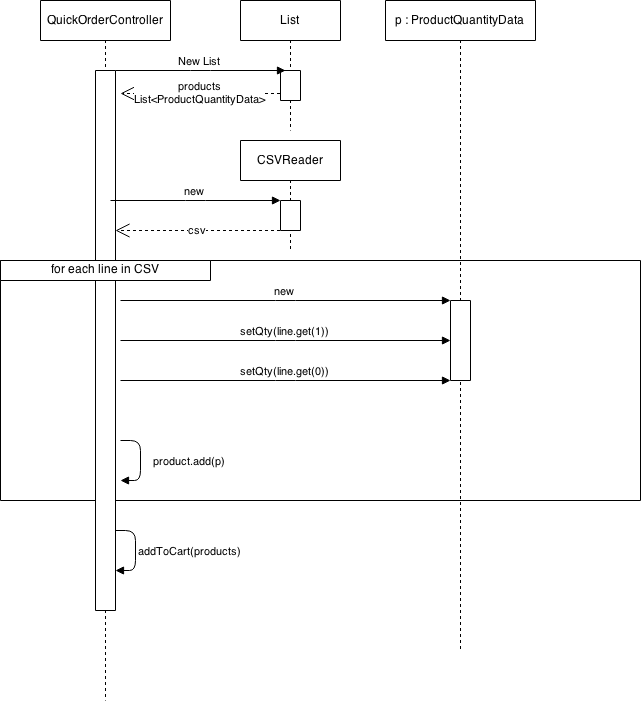


Fig: Quick order CSV Import sequence diagram

Description:

|  |  |  |
| --- | --- | --- |
| **#** | **Component** | **Description** |
| **1** | QuickorderController | It is controller to handle the quick order and CSV import. When customer/user imports csv file & this controller will be called to add the products to cart quickly by calling upload2() method and it will call internally findProductsForQueryStart() method. |
| **2** | ProductQuantityData | It is a class which contains product and quantity data |
| **3** | CSVReader | It is an interface it contains two methods findProductsForQuery() and findProductForQuery() these methods uses the quickOrderSearchStrategy, QuickOrderDAO respectively |
| **4** | cartFacade | It is used to add the products to cart. |

Use Case Diagram

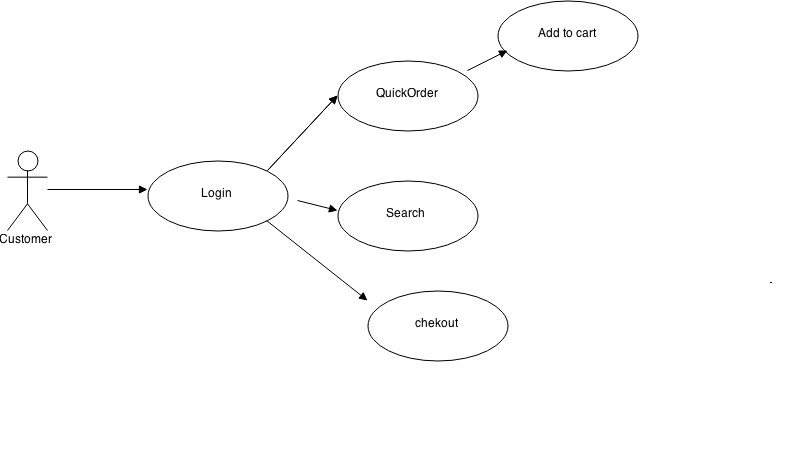
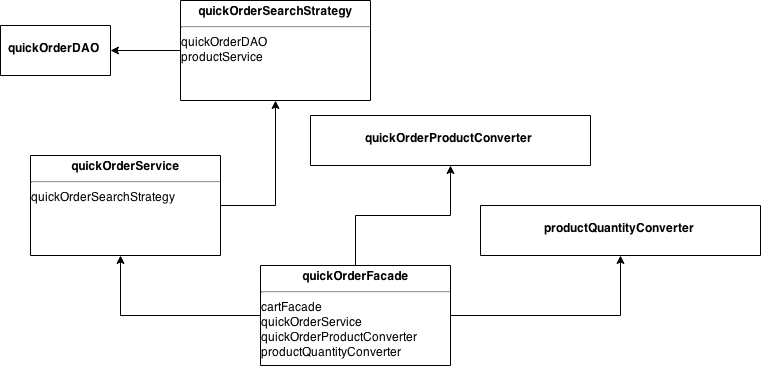


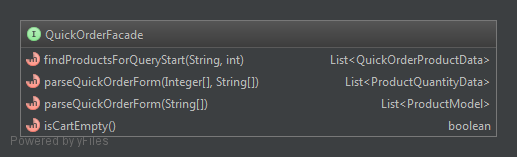
Fig: Quick order use case diagram.

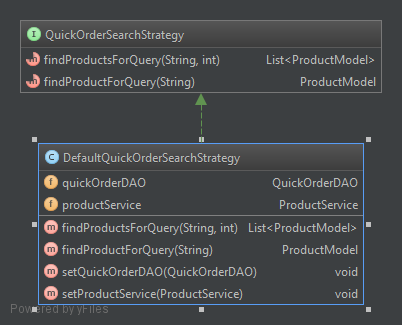
**Description:** B2B customer can login using his credentials and he can place order quickly by entering product quantity and SKU number.

|  |  |  |
| --- | --- | --- |
| # | Use case/Activity | Description |
| 1 | Login | B2B customer/user has to login using his credentials to place the quick order. |
| 2 | QuickOrder | By entering the Product SKU number user can add the product quickly to cart. To place an order quickly. |
| 3 | Search | B2B customer can search the product and he adds the product to cart. |
| 4 | Checkout | Once product is added to cart the B2B customer can proceed to check out the product. |

1. Detailed Application Design

Class Diagram





Description: This quick order module is an extension which will be used to add the products quickly to the cart to place an order quickly. From the QuickOrderController quick order facade is called. This facade uses the quickOrderService this service uses Quick order search strategy again this will uses quickOrderDAO to search the product availability using the SKU. If the product is available this will be then that product will be allowed to add to the cart.

Beans

**QuickOrderProductData**: This bean is used for the quick order. This bean is not auto generated as of now. In future implementation it has to configure in xml file to generate automatically.

Facades

**QuickorderFacade:** It is an interface it is used in Quick order controller to parse and find the user entered product SKU numb and quantity.

Strategy

**QuickOrderSearchStrategy:** This strategy is used for searh mechanism for the user entered SKU. It internally uses the quick order dao and product service for the product search.

Controller

It is controller to handle the quick order.

* When customer/user enters a SKU number and the quantity this controller will be called to add the products to cart quickly. It contains the search() this will be called automatically and it will call internally findProductsForQueryStart() method.
* When customer/user imports csv file & this controller will be called to add the products to cart quickly by calling upload() method.

View/JSP

1. Quickordercomponent.jsp is a custom JSP this will allow the b2b customers/users to add the products quickly to the cart.

The JSP page consists 3 sections:

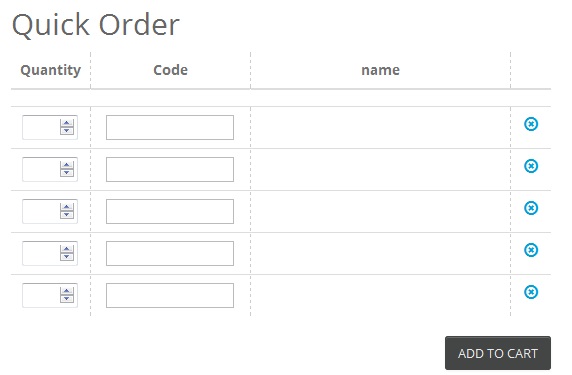
1. Header section
2. Input section and delete button.
3. Submit button for Add to cart.

The header section contains the column names like Quantity, product SKU (code), Name.

Input section contains the fields that can be used for searching the product using SKU and quantity field this can be used to set the quantity of the of the product to place the order.

Once the user enters SKU code it will check automatically for the product stock availability. If the product is available then it will allow the users to add to cart.

Below is the mock up of the JSP page for quick order component. The page will be enhanced in future releases.



.

Fig: Quick order component JSP.

1. csvordercomponent.jsp is a custom JSP this will allow the b2b customers/users to import csv file & add the products quickly to the cart.

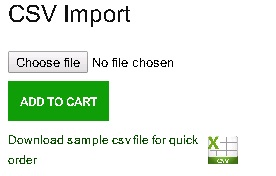
The JSP page consists 3 sections:

1. Input section to import a file.
2. Submit button for Add to cart.
3. Link for sample file download.

Input section contains only one field that can be used for searching the product using SKU and quantity field this can be used to set the quantity of the of the product to place the order.

Once the user imports csv file it will read the each and every line of CSV file & checks for the product stock availability. If the product is available then it will add the products to a cart. If the product is not available it will show the error messages.

Below is the mock up of the JSP page for csv order component. The page will be enhanced in future releases



Description:

Using download sample CSV file for quick order user can down load the sample CSV file and use the can the data to import file for quick order.

Using Choose file user can import the CSV file.

1. Database architecture

ER Diagram

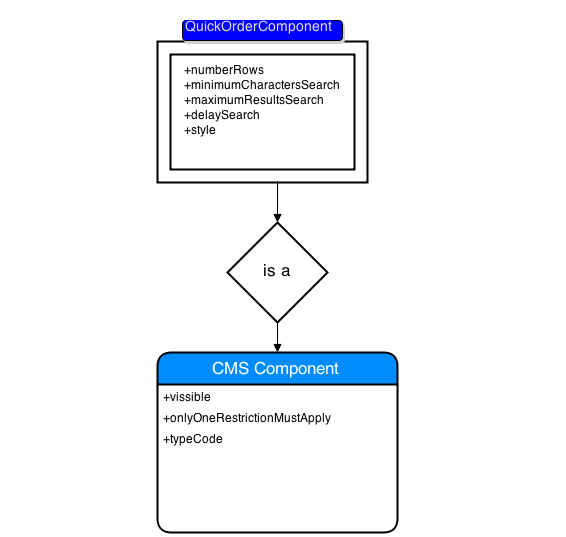


Fig: Quick order ER diagram

**ER Diagram Description:**

Quick order component is an item type. This component itself can’t exist in the database as it is a weak entity and it will available as logically and not present physically in the database. It is associated with CMS Component table. It will be having below attributes.

|  |  |
| --- | --- |
| numberRows | Number of rows in the table. |
| minimumCharactersSearch | Number of characters for auto-completion to start. |
| maximumResultsSearch | Maximum number of results to display by auto-completion. |
| delaySearch | Delay in milliseconds before displaying auto-completion results. |
| styleClass | CSS style class of this quick order component. |

1. Exception Handling and Logging

Exception Scenarios

CommerceCartModification can occur when user modify the quantity of the product.

IllegalState exception can occur when the product price is not defined for user entered SKU.

UnknownIdentifierException occur when it finds unknown product which is not available in the database.

NumberFormatException can occur when quantity is defined or its incorrect format for user entered SKU

|  |  |
| --- | --- |
| Exception | Message |
| CommerceCartModification | Error occurred while adding to cart. |
| IllegalState | Products were not added to the cart due to missing prices. |
| UnknownIdentifierException | Error occurred due to unknown product. |
| NumberFormatException | Products were not added to the cart due to quantity was wrong format. |

Exception Handling:

Below is the Exception handling best practice used in the module

* Never swallow the exception in catch block
* Declare the specific checked exceptions that your method can throw
* Do not catch the Exception class rather catch specific sub classes
* Never catch Throwable class
* Always correctly wrap the exceptions in custom exceptions so that stack trace is not lost
* Either log the exception or throw it but never do the both
* Never throw any exception from finally block
* Always catch only those exceptions that you can actually handle
* Don't use printStackTrace() statement or similar methods
* Use finally blocks instead of catch blocks if you are not going to handle exception
* Remember "Throw early catch late" principle
* Always clean up after handling the exception
* Throw only relevant exception from a method
* Never use exceptions for flow control in your program
* Validate user input to catch adverse conditions very early in request processing
* Always include all information about an exception in single log message
* Pass all relevant information to exceptions to make them informative as much as possible
* Always terminate the thread which it is interrupted
* Use template methods for repeated try-catch
* Document all exceptions in your application in javadoc

Logging

* For logging in the quick order module Log4j will be used.
* The logging will be set at INFO level.

1. Installation and Configuration

Installation steps for the addon

1. Add the quickorder module dependency in your localextension.xml. It is available in hybris\config\localextensions.xml

<extension name=”quickorder”/>

1. Open the command and go to the folder hybris\bin\platform.
2. Execute the commands
   1. set antenv.bat
   2. ant addoninstall -Daddonnames="quickorder" - DaddonStorefront yb2bacceleratorstorefront ="yb2bacceleratorstorefront"

WARNING: yb2bacceleratorstorefront must be replaced by your custom storefront

1. Update the project data by selecting the quick order using HAC.
2. Update the b2bassets initial data using HAC.

Configuration steps for the addon

1. Quick order beans and base package has to configure in quickorder-spring.xml file. This file is available at hybris\bin\custom\quickorder\resources\quickorder-spring.xml
2. Quick Order component has to add in essentialdata\_quickorder.impex. This file is available at hybris\bin\custom\quickorder\resources\quickorder\import\essentialdata\_quickorder.impex

INSERT\_UPDATE ComponentTypeGroups2ComponentType;source(code)[unique=true];target(code)[unique=true]

;narrow;QuickOrderComponent

;wide;QuickOrderComponent

;narrow;HomepageCsvOrderComponent

;wide;HomepageCsvOrderComponent

1. To Add the CSV order and Quick order components in home page add the below content in /b2bassetsinitialdata/resources/b2bassetsinitialdata/import/contentCatalogs/powertoolsContentCatalog/cms-content.impex

INSERT\_UPDATE QuickOrderComponent;$contentCV[unique=true];uid[unique=true];name;numberRows;minimumCharactersSearch;maximumResultsSearch;delaySearch;styleClass;&componentRef

;;HomepageQuickOrderComponent;Homepage Quick Order Component;5;3;4;300;;HomepageQuickOrderComponent

INSERT\_UPDATE CsvOrderComponent;$contentCV[unique=true];uid[unique=true];name;file;&componentRef

;;HomepageCsvOrderComponent;Homepage Csv Order Component;;HomepageCsvOrderComponent

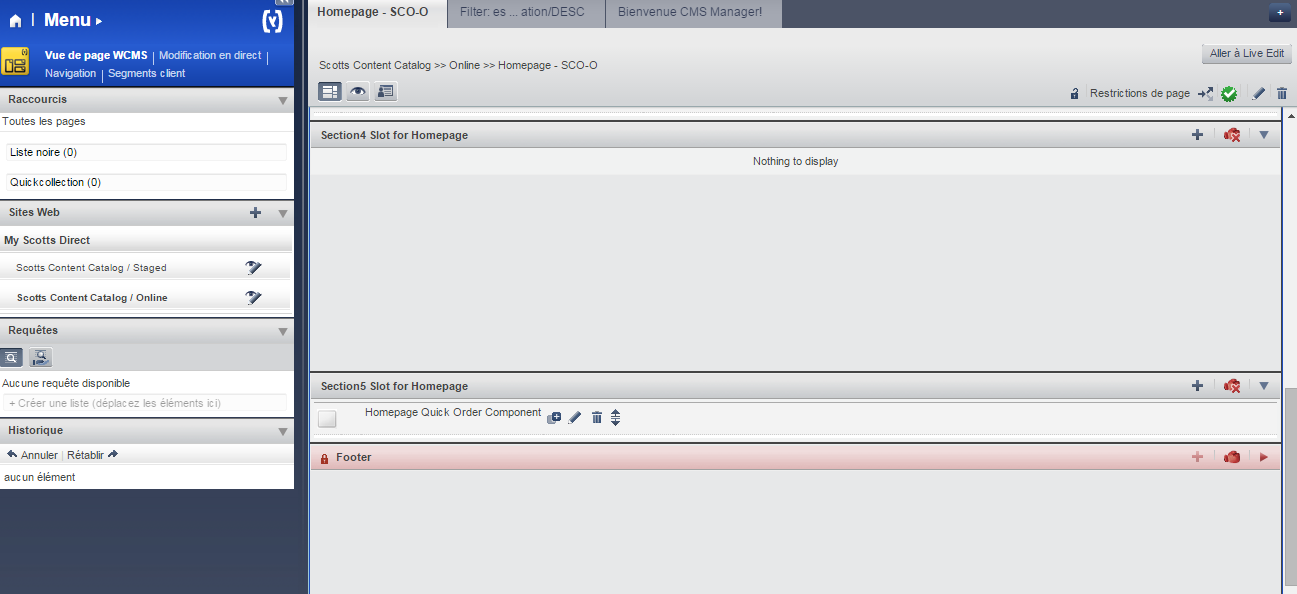
INSERT\_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(&componentRef)

;;Section2ASlot-B2BAssetsHomepage;Section2A Slot for Homepage;true;HomepageQuickOrderComponent, HomepageCsvOrderComponent

INSERT\_UPDATE ContentSlotForPage;$contentCV[unique=true];uid[unique=true];position[unique=true];page(uid,$contentCV)[unique=true][default='homepage'];contentSlot(uid,$contentCV)[unique=true]

;;Section2A-Homepage;Section2A;;Section2ASlot-B2BAssetsHomepage

1. Open the WCMS and select he home page template of your project and insert this quickorderComponet. The sample screen shoot is given below.(optional)



How to add a new column in JSP.

1. Add the new column in quickordercomponent.jsp.
2. Add new filed in QuickOrderProductData bean.
3. Modify the quickOrderProductConverter to populate the new filed.
4. Modify the JS to fill the DOM document to show the new column in the JSP.
5. Appendix A *– Glossary of terms*

|  |  |
| --- | --- |
| **Term/Abbreviation/Acronym** | **Meaning** |
| SKU | Stock Keeping Unit |
| CSV | Comma Separated Values |
| HAC | Hybris Administrative console |