

Credit Points Module

Technical Design Documentation

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Updated By | Date | Comments |
| 1.0 | Kunal Waghmare | 07/04/2016 | Initial Draft |

Document Sign-Off

|  |  |  |
| --- | --- | --- |
| Name | Role | Signature/Date |
| Gyaneshwar Dubey | Mentor |  |
| Suman Thakur | Lead |  |

Table of Contents

[1. Introduction 3](#_Toc448139924)

[1.1 Purpose 3](#_Toc448139925)

[1.2 Intended Audience 3](#_Toc448139926)

[1.3 Document Scope 3](#_Toc448139927)

[1.4 Functional Rules 3](#_Toc448139928)

[1.5 Assumptions/Restrictions 3](#_Toc448139929)

[1.6 Documents Structure 4](#_Toc448139930)

[2. High Level Design Diagram 5](#_Toc448139931)

[2.1 Overview 5](#_Toc448139932)

[2.2 Use Case Diagram 7](#_Toc448139933)

[3. Detailed Application Design 8](#_Toc448139934)

[3.1 Class Diagram 8](#_Toc448139935)

[3.2 Beans 10](#_Toc448139936)

[3.3 Strategy 10](#_Toc448139937)

[3.4 Controller 10](#_Toc448139938)

[3.5 Populator 10](#_Toc448139939)

[3.6 View/JSP 10](#_Toc448139940)

[4. Exception Handling and Logging 12](#_Toc448139941)

[4.1 Exception Scenarios 12](#_Toc448139942)

[4.2 Exception Handling: 12](#_Toc448139943)

[4.3 Logging 12](#_Toc448139944)

[5. Implementation steps 13](#_Toc448139945)

[5.1 Implementation steps for Credit Points Module 13](#_Toc448139946)

[6. Appendix A *– Glossary of terms* 17](#_Toc448139947)

1. Introduction

Purpose

This document is the Technical Design Document which provides information and steps to configure the credit points module. It covers:

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

Intended Audience

This document is aimed at Hybris developers and Technical members who want to implement the credit points module in Hybris.

Document Scope

This document covers the details about the credit points module.

Functional Rules

* Whenever new customer registers on site some credit points are given to that customer.
* My-Account section shows available credit points.
* Customer can use these points while checkout on select delivery method page.

Assumptions/Restrictions

This module has been created for version 5.7.0.5.

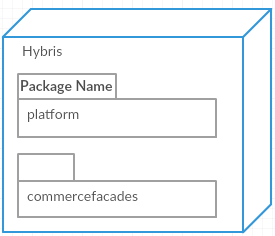
Documents Structure

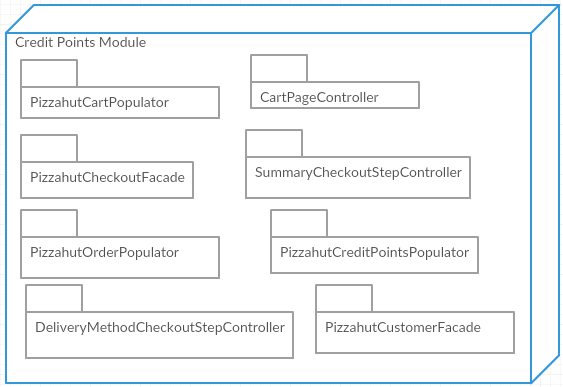
|  |  |  |
| --- | --- | --- |
| Section | | Description |
| 1 | Introduction | Describes the purpose and intended audience of the document and explains its structure. |
| 2 | High Level Design | Overview of the module. |
| 3 | Detailed Application Design | Application architecture applicable to this module. |
| 6 | Implementation steps | Steps to implement the module. |
| 7 | Appendices | Supporting information. |

1. High Level Design Diagram

Overview

Below is the High level overview of the products addons module.

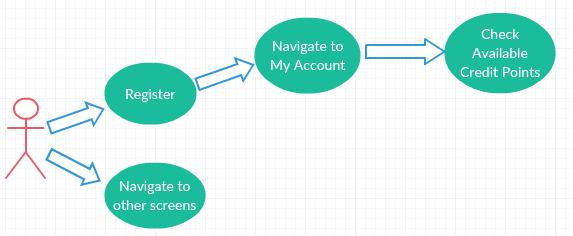


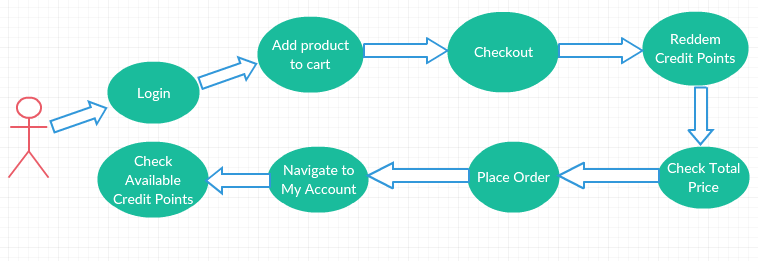


|  |  |  |
| --- | --- | --- |
| **#** | **Component/Extension/Module** | **Description** |
| 1 | **Platform** | The hybris Platform consists of a standard set of extensions providing the main functionality of a hybris installation. |
| 2 | **Commercefacades** | The hybris **commercefacades** extension provides a suite of facades that make up a unified multichannel storefront API which can be used by multiple front-ends. The facade's responsibility is to integrate existing business services from the full range of the hybris extensions and expose a Data object (POJO) response adjusted to meet the storefront requirements |
| 3 | **CartPageController** | Handles requests for checkout and calculates eligible credit points for cart |
| 4 | **PizzahutCheckoutFacade** | Overrides placeOrder() method of DefaultCheckoutFacade.java |
| 5 | **DeliveryMethodCheckoutStepController** | Handles request for selecting delivery method and recalculates cart total price |
| 6 | **SummaryCheckoutStepController** | Handles requests for placing the order and gets selected addons and pass it to order entries |
| 7 | **PizzahutRegisterPageController** | Overrides “processRegisterUserRequest()” method of “AbstractRegisterPageController.java” to call “register()” method of “PizzahutCustomerFacade.java” |
| 8 | **PizzahutCustomerFacade** | Overrides “register()” method of “DefaultCustomerFacade.java” |
| 9 | **PizzahutCartPopulator** | This populates data from CartModel to CartData |
| 10 | **PizzahutOrderPopulator** | populates data from OrderModel to OrderData |
| 11 | **PizzahutCreditPointsPopulator** | This populates data from CustomerModel to CustomerData |

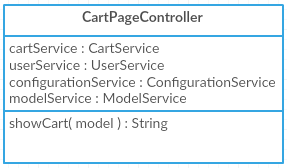
Use Case Diagram

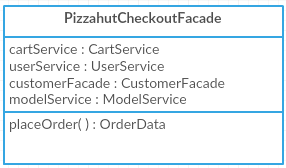
Use case diagram for the Credit Points module

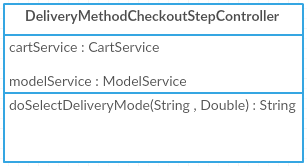


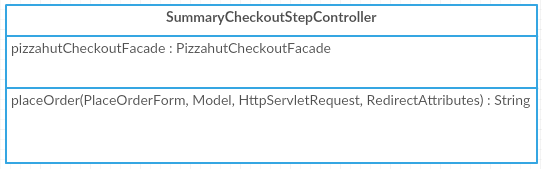


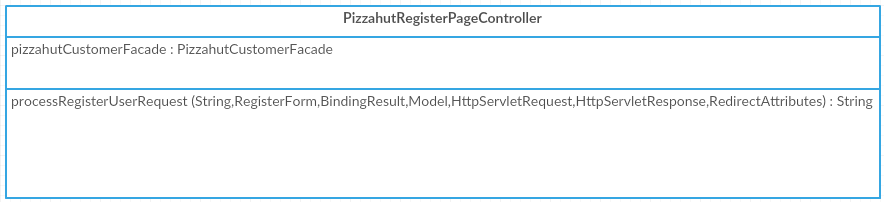
1. Detailed Application Design

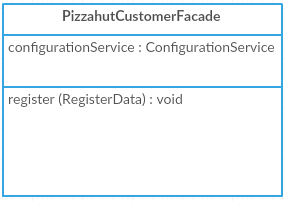
Class Diagram  
  












Beans

Strategy

N.A

Controller

**CartPageController:** Calculates eligible credit points for cart.

**DeliveryMethodCheckoutStepController:** To apply credit points to cart and reduce the total cart price according to applied credit points.

**SummaryCheckoutStepController:** To call new placeOrder() method.

**PizzahutRegisterPageController:** To call new register() method of PizzahutCustomerFacade.java

Populator

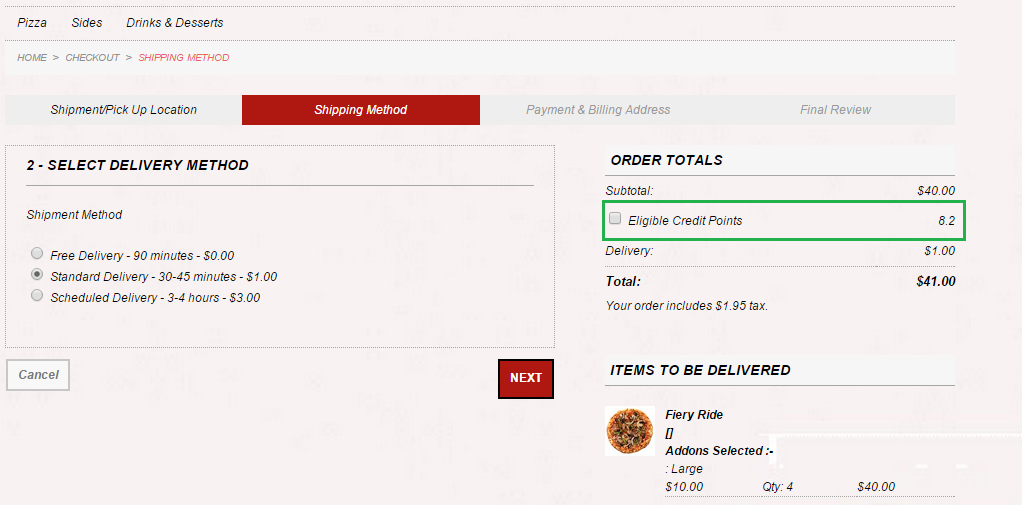
**PizzahutCartPopulator:** This populates data from CartModel to CartData

**PizzahutOrderPopulator:** This populates data from OrderModel to OrderData

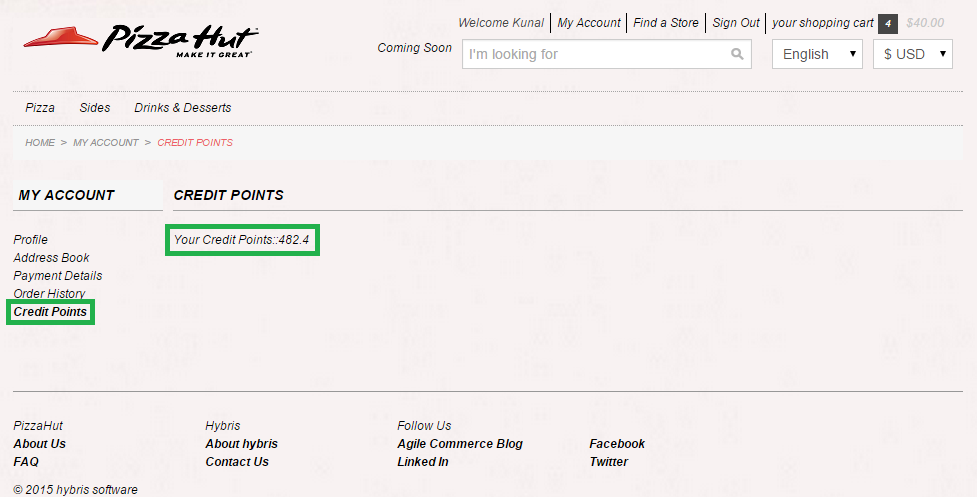
**PizzahutCreditPointsPopulator:** This populates data from CustomerModel to CustomerData

View/JSP

**chooseDeliveryMethodPage.jsp:** This jsp displays Eligible Credit Points on Select Delivery Method Page.



**accountCreditPointsPage.jsp:** This jsp displays Available Credit Points on MyAccount section



1. Exception Handling and Logging

Exception Scenarios

N.A

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 advanced product search module Log4j will be used.
* The logging level will be set at INFO level.
* The exceptions are logged using Log.error() method.

1. Implementation steps

Implementation steps for Credit Points Module

1. Create new attributes for Customer, Cart and Order item type in pizzahutcore-items.xml

|  |
| --- |
| <typegroup name=*"Customer"*>  <itemtype code=*"Customer"* autocreate=*"false"* generate=*"false"*>  <description>Extending Customer type with additional attributes.  </description>  <attributes>  <attribute autocreate=*"true"* qualifier=*"availableCreditPoints"* type=*"java.lang.Double"*>  <modifiers optional=*"false"* />  <persistence type=*"property"* />  <description>It holds information about customer creditPoints.  </description>  </attribute>  </attributes>  </itemtype>  </typegroup> |

|  |
| --- |
| <typegroup name=*"Cart"*>  <itemtype code=*"Cart"* autocreate=*"false"* generate=*"false"*>  <description>Extending Cart type with additional attributes for credit points functionality</description>  <attributes>  <attribute autocreate=*"true"* qualifier=*"eligibleCreditPoints"* type=*"java.lang.Double"*>  <modifiers optional=*"true"* />  <persistence type=*"property"* />  <description>It holds information about customer creditPoints.  </description>  </attribute>  </attributes>  </itemtype>  </typegroup> |

|  |
| --- |
| <typegroup name=*"Order"*>  <itemtype code=*"Order"* autocreate=*"false"* generate=*"false"*>  <description>Extending Order type with additional attributes.</description>  <attributes>  <attribute qualifier=*"usedCreditPoints"* type=*"java.lang.Double"*>  <description>Used Credit Points</description>  <persistence type=*"property"* />  </attribute>  </attributes>  </itemtype>  </typegroup> |

1. Add following properties in “local.properties” file.

|  |
| --- |
| hybris\config\local.properties |
| pizzahut.credit.discount=20  pizzahut.credit.points=500 |

1. Create “PizzahutCartPopulator.java” to populate new cart attribute
2. Create “PizzahutOrderPopulator.java” to populate new order attribute
3. Create “PizzahutCreditPointsPopulator.java” to populate new customer attribute

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. Following JSP and TAG files of “pizzahutstorefront” extension used to display Eligible Credit points for logged-in customer

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. Override “register()” method of “DefaultCustomerFacade.java” in “PizzahutCustomerFacade.java”. This “register()” method is used to give some credit points to customer when he/she registers on store.

|  |  |
| --- | --- |
|  |  |

1. “CartPageController.java” is used to calculate eligible credit points for cart

|  |
| --- |
|  |

1. “DeliveryMethodCheckoutStepController.java” is used to apply credit points to cart and reduce the total cart price according to applied credit points

|  |
| --- |
|  |

1. Create “PizzahutCheckoutFacade.java” which extends “DefaultCheckoutFacade.java” and override “placeOrder()” method
2. This “placeOrder()” method is used to set total order price and to reduce credit points from customer’s account.

|  |
| --- |
|  |

1. “SummaryCheckoutStepController.java” is used to call above mentioned new “placeOrder()” method.

|  |
| --- |
|  |

1. Following files required to display available credit points on My Account section:

**cms-content.impex**

|  |
| --- |
| hybris\bin\custom\pizzahut\pizzahutinitialdata\resources\pizzahutinitialdata\import\sampledata\contentCatalogs\pizzahutContentCatalog\**cms-content.impex** |
| INSERT\_UPDATE CMSLinkComponent;$contentCV[unique=true];uid[unique=true];name;url;&linkRef;&componentRef;target(code)[default='sameWindow'];$category;$product;  ;;AccountCreditPointsLink;AccountCreditPointsLink;/my-account/credit-points;AccountCreditPointsLink;AccountCreditPointsLink;;;;  # ContentSlot  INSERT\_UPDATE ContentSlot;$contentCV[unique=true];uid[unique=true];name;active;cmsComponents(&componentRef);;;  ;;SideContent-credit-points;Side Content Slot for credit-points;true;AccountLeftNavigationComponent;;;  ;;BodyContent-credit-points;Body Content Slot for credit-points;true;AccountCreditPointsComponent;;;  # ContentSlotForPage  INSERT\_UPDATE ContentSlotForPage;$contentCV[unique=true];uid[unique=true];position[unique=true];page(uid,$contentCV)[unique=true][default='credit-points'];contentSlot(uid,$contentCV)[unique=true];;;  ;;SideContent-credit-points;SideContent;;SideContent-credit-points;;;  ;;BodyContent-credit-points;BodyContent;;BodyContent-credit-points;;;  INSERT\_UPDATE ContentPage;$contentCV[unique=true];uid[unique=true];name;masterTemplate(uid,$contentCV);label;defaultPage[default='true'];approvalStatus(code)[default='approved'];homepage[default='false'];previewImage(code, $contentCV)[default='ContentPageModel\_\_function\_preview']  ;;credit-points;credit-points Page;AccountPageTemplate;/credit-points |

**cms-content\_en.impex**

|  |
| --- |
| hybris\bin\custom\pizzahut\pizzahutinitialdata\resources\pizzahutinitialdata\import\sampledata\contentCatalogs\pizzahutContentCatalog\**cms-content\_en.impex** |
| # CMS Link Components  UPDATE CMSLinkComponent;$contentCV[unique=true];uid[unique=true];linkName[lang=$lang]  ;;AccountCreditPointsLink;"Credit Points" |

**accountCreditPointsPage.jsp**

|  |
| --- |
| hybris\bin\custom\pizzahut\pizzahutstorefront\web\webroot\WEB-INF\views\desktop\pages\account\accountCreditPointsPage.jsp |
|  |

1. Appendix A *– Glossary of terms*

|  |  |
| --- | --- |
| **Term/Abbreviation/Acronym** | **Meaning** |
|  |  |
|  |  |
|  |  |