

Restrictions Module

Technical Design Documentation

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Updated By | Date | Comments |
| 0.1 | Shashank Bezalwar | 05/02/2015 | Initial Draft |
| 1.0 | Jayachandra Gudiwada | 20/03/2015 | Initial Draft |
|  |  |  |  |

Document Sign-Off

|  |  |  |
| --- | --- | --- |
| Name | Role | Signature/Date |
|  |  |  |

Table of Contents

[1. Introduction 3](#_Toc414880822)

[1.1 Purpose 3](#_Toc414880823)

[1.2 Intended Audience 3](#_Toc414880824)

[1.3 Document Scope 3](#_Toc414880825)

[1.4 Functional Rules 3](#_Toc414880826)

[1.5 Assumptions/Restrictions 3](#_Toc414880827)

[1.6 Documents Structure 4](#_Toc414880828)

[1.7 Additional Resources 4](#_Toc414880829)

[2. High Level Design Diagram 5](#_Toc414880830)

[2.1 Overview 5](#_Toc414880831)

[2.2 Sequence Diagram 6](#_Toc414880832)

[2.3 Use Case Diagram 8](#_Toc414880833)

[3. Detailed Application Design 9](#_Toc414880834)

[3.1 Class Diagram 9](#_Toc414880835)

[3.2 Beans 11](#_Toc414880836)

[3.3 Facade 11](#_Toc414880837)

[3.4 Strategy 11](#_Toc414880838)

[3.5 Controller 11](#_Toc414880839)

[4. Database architecture 12](#_Toc414880840)

[4.1 ER Diagram 12](#_Toc414880841)

[5. Exception Handling and Logging 14](#_Toc414880842)

[5.1 Exception Handling Best Practises: 14](#_Toc414880843)

[5.2 Logging 14](#_Toc414880844)

[6. Installation and Configuration 15](#_Toc414880845)

[6.1 Installation steps for the extension 15](#_Toc414880846)

[6.2 Configuration steps for the Extension 15](#_Toc414880847)

[7. Appendix A *– Glossary of terms* 17](#_Toc414880848)

1. Introduction

Purpose

This document is the Technical Design Document which provides information of solrsearchrestriction extension 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 solrsearchrestriction extension in Hybris.

Document Scope

This document covers the details about the sprint 1 solrsearchrestriction extesnion.

Functional Rules

* This module (a specific behaviour that does not exist in Standard Hybris) helps to create various products assortments for different customers and also manage partial catalog
* Products are included in subsets that match customer assortments, it means there is a set of products that may be suggested to some customers and not to others
* The products of these assortments may be common to many customers
* Among the customer assortment products, some may be viewable or non-viewable

Assumptions/Restrictions

This module has been created for version 5.4.0.0. No B2B Unit should be named as unknown.

**Display**: Viewable products are displayed for all B2B unit users. Non-viewable products are not viewable for the B2B unit users

**Import**: The import is done through an asynchronous interface (CSV files)

**Export:** No export is available

**Search:** Viewable products are searchable; however non-viewable products are not searchable

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

Wiki Portal: <https://wiki.hybris.com>

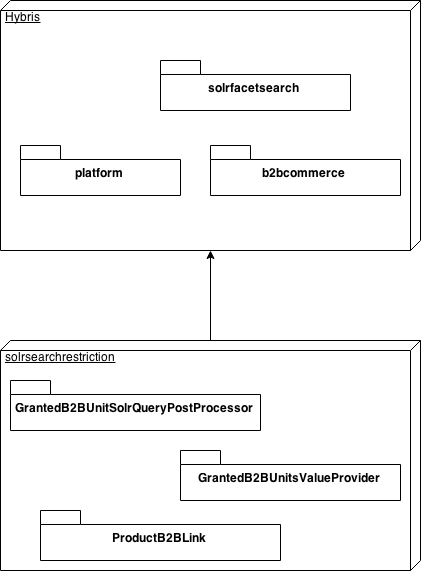
Coconet:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | URL | Purpose |
| 1 | Wiki Portal | <https://wiki.hybris.com> | To gather more information |
| 2 | Coconet | <https://coconet2.capgemini.com/sf/go/doc3343977?nav=1> | To refer functional specifications |

1. High Level Design Diagram

Overview

Below is the High level overview of the solrsearchrestriction extension.



**Solrfacetsearch extension:** The solrfacetsearch extension provides the simple interface that can be used to create interaction with Solr.

**Platform:** The hybris Platform consists of a standard set of extensions providing the main functionality of a hybris installation.

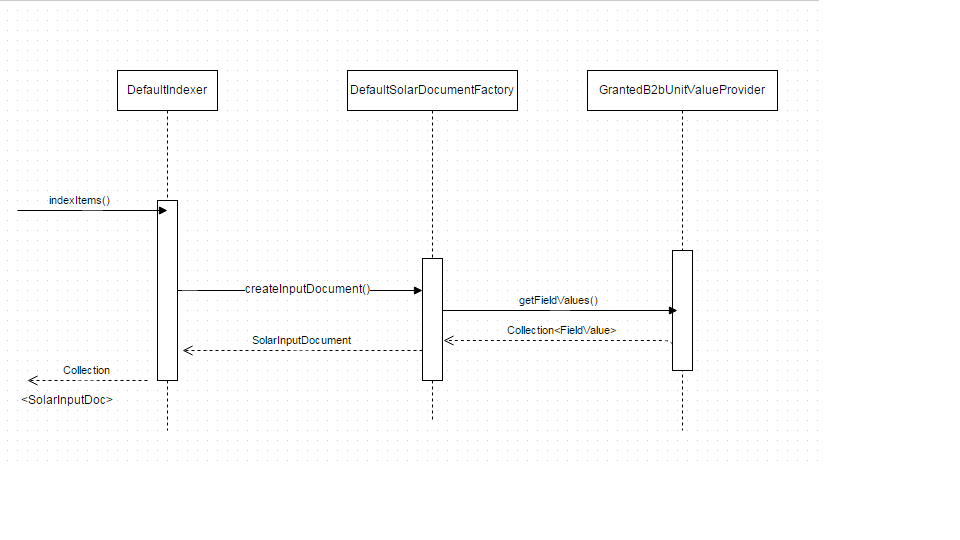
**B2BCommerce extension:** The b2bcommerce extension enables to provide B2B functionality to the organization.

**GrantedB2BUnitSolrQueryPostProcessor:** GrantedB2BUnitSolrQueryPostProcessor adds the filter query to the Solr query to enable restriction.

**GrantedB2BUnitsValueProvider:** GrantedB2BUnitsValueProvider provides the list of B2B units that are granted permission to see the product.

**ProductB2BUnitLink:** ProductB2BUnitLink table stores the isDisplayable flag which is linked to B2B unit and Product.

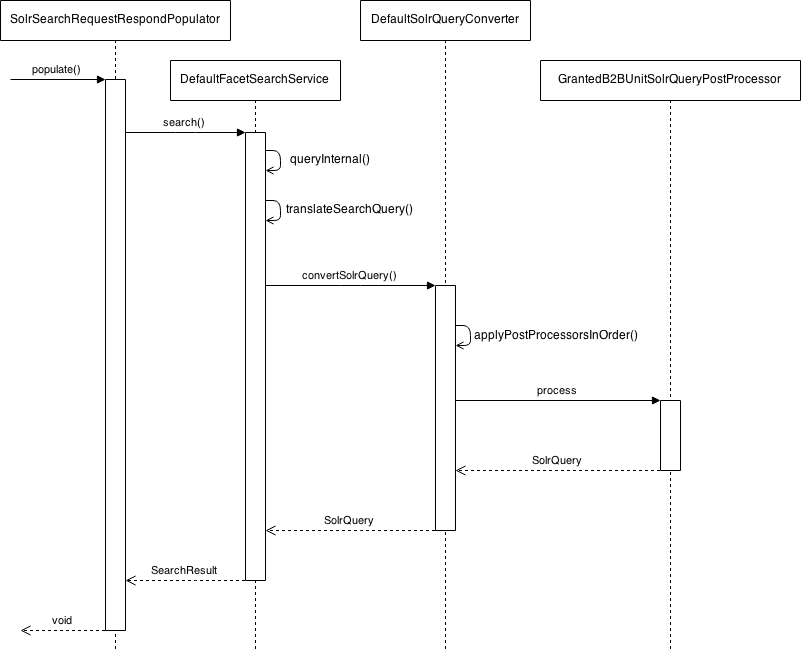
Sequence Diagram

Sequence of events during the Solr Indexation 

**DefaultIndexer:** The DefaultIndexer gets the Product Type and sends it to Indexer. DefaultIndexer will return list of Products converted to solrDocuments.

**DefaultSolrDocumentFactory:** The DefaultSolrDocumentFactory converts the ProductModel to SolrInputDocument.

**GrantedB2BUnitsValueProvider:** The DefaultSolrDocumentFactory uses the GrantedB2BUnitsValueProvider to get the list of B2B Units that can view the product.

Sequence of events during the retrieval of the products for B2B Units

**SolrSearchRequestResponcePopulator:** The SolrSearchRequestResponcePopulator creates the search query and sends it to DefaultFacetSearchService.

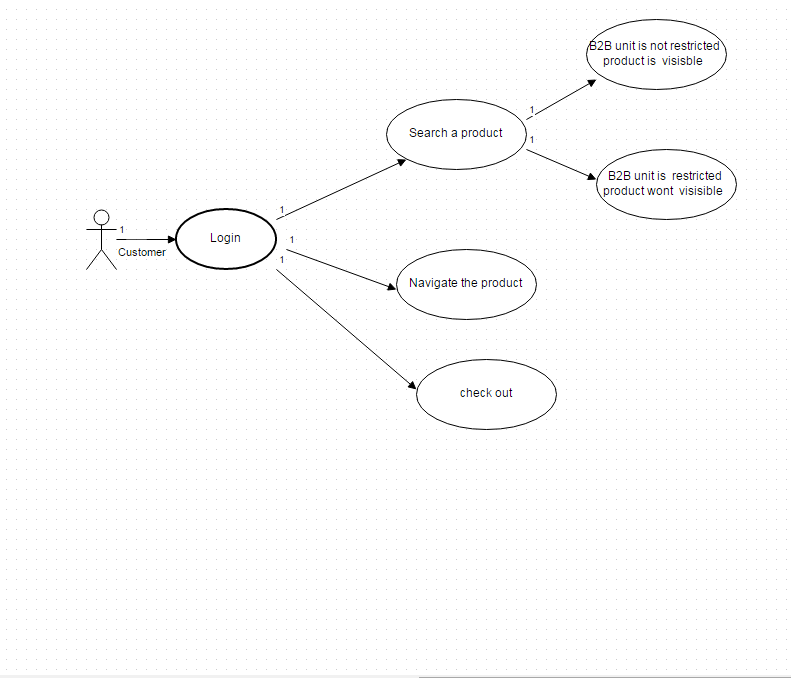
**DefaultFacetSearchService:** The DefaultFacetSearchService does the search with the specific query received.

**DefaultSolrQueryConverter:** The DefaultSolrQueryConverter converts the [SearchQuery](https://download.hybris.com/api/5.4.0/commercesuite/de/hybris/platform/solrfacetsearch/search/SearchQuery.html) instance into valid SolrQuery.

**GrantedB2BUnitSolrQueryPostProcessor:** The GrantedB2BUnitSolrQueryPostProcessor adds the filter to the query to restrict the result.

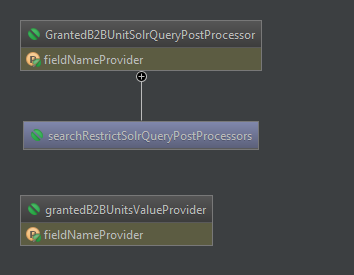
Use Case Diagram

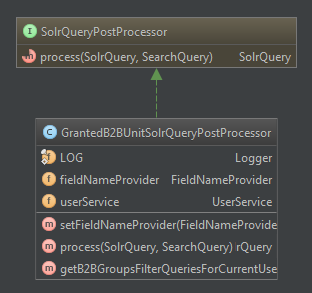
Use case diagram for the solrsearchrestriction extension. B2B customer has to loging using the credentials. Then customer can find the products if the B2BUnit is not restricted. If the B2BUnit is restrictied then it won’t visible to the customer.

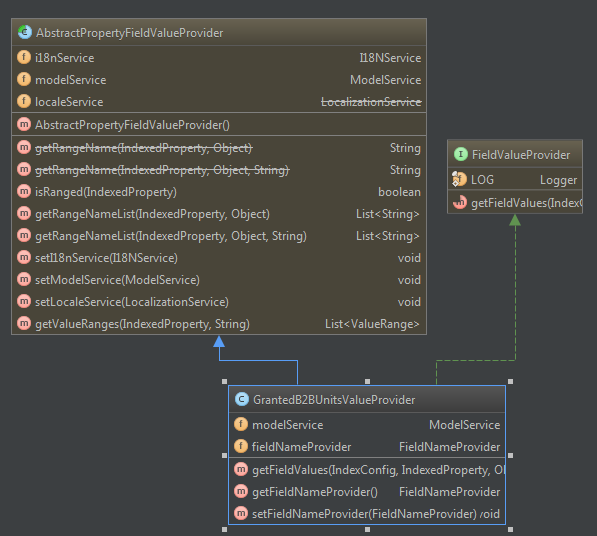


1. Detailed Application Design

Class Diagram







In the solrsearchrestriction two classes will have to be created.

1. GrantedB2BUnitSolrQueryPostProcessor
2. GrantedB2BUnitsValueProvider

GrantedB2BUnitSolrQueryPostProcessor : The GrantedB2BUnitSolrQueryPostProcessor should implement SolrQueryPostProcessor, which provides a direct access to the query right before performing search on Solr's index. This will enable to add a filter query to Solr query to retrieve only products that current user can see depending on its B2Bunit. If user is anonymous or is not in any B2BUnit, then the filter query should restrict in a way that no product can be seen.

GrantedB2BUnitsValueProvider : The GrantedB2BUnitsValueProvider should extend abstract class AbstractPropertyFieldValueProvider. This class will provide the list of B2BUnits that has access to the product. This will be done for each product during the Solr Indexation.

For each B2BUnit and product, the module will have to set the displayable status. For this the module will create a new SQL table where the displayable status is saved for each product and B2BUnit. The details of the table are specified in section 4 of this document.

For the read access, a restriction is to be added in Hybris via impex.

Beans

N.A

Facade

N.A

Strategy

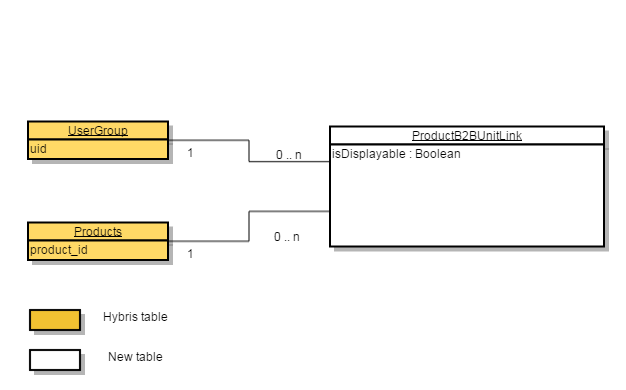
N.A

Controller

N.A

1. Database architecture

ER Diagram



Below are the tables used in this module. The Hybris tables are not impacted and only one new table ProductB2BUnitLinkis created. The module uses three types B2BUnit, Products and ProductB2BUnitLink. The B2BUnit type extends Company which extends UserGroup type. There is no physical table for the B2BUnit type.

**Table ProductB2BUnitLink:**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Mandatory | Description |
| isDisplayable | Boolean | Yes | Information to display the right products to the client |
| b2bunit | B2BUnit | Yes | Unique identifier for the B2B Unit |
| product | Product | Yes | Unique identifier for the product |

**Table Products:**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Mandatory | Description |
| code (Products\_id ) | Product | Yes | Unique identifier for the product |

**Table UserGroup:**

|  |  |  |
| --- | --- | --- |
| Column | Data Type | Description |
| Uid | B2BUnit | Unique identifier group for the B2B Unit |

1. Exception Handling and Logging

Exception Handling Best Practises:

* 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 solrsearchrestriction module Log4j will be used.
* The logging will be set at INFO level.

1. Installation and Configuration

Installation steps for the extension

1. Add the solrsearchrestriction module dependency in your localextension.xml

<extension name=" solrsearchrestriction "/>

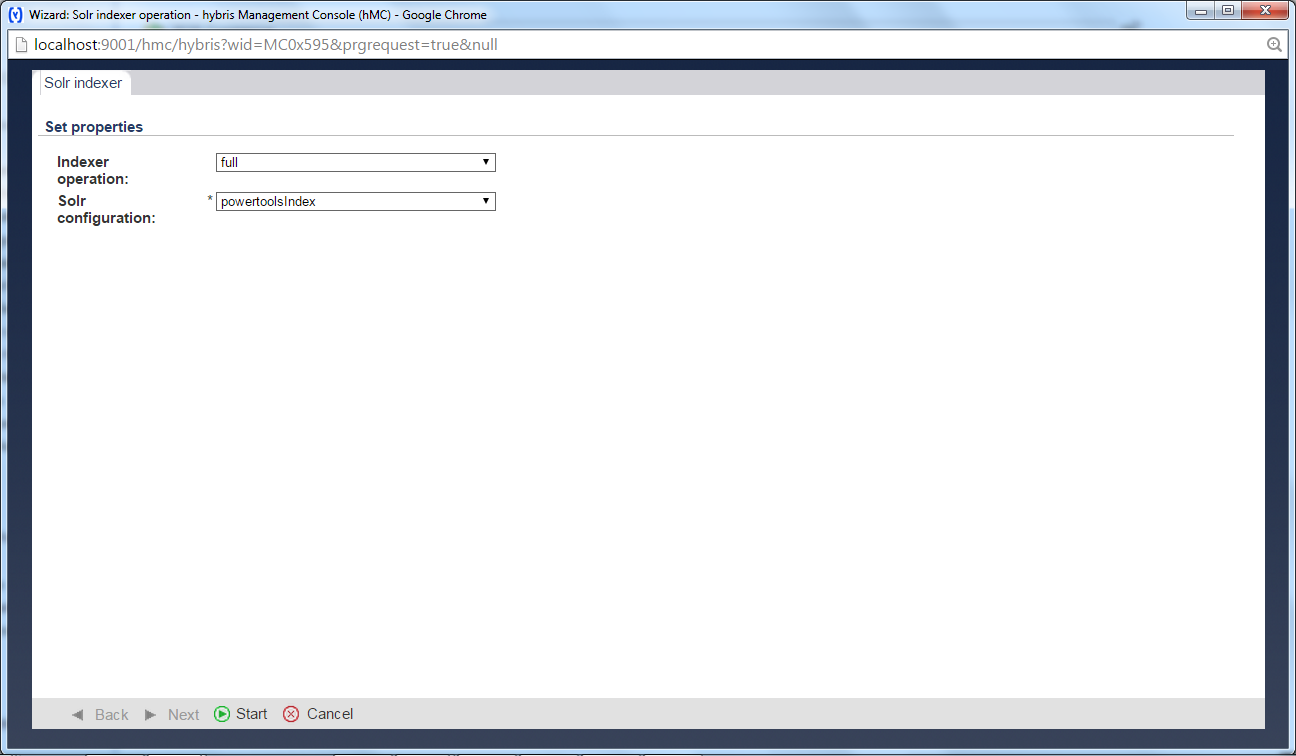
1. Go to platform directory where hybris is installed using command prompt. As shown in the below example.

Cd D:\B2BAssets\hybris\bin\platform

1. Execute the below command.
   1. Setantenv.bat
   2. Ant all
   3. Hybrisserver.bat
2. Go to admin console using http://<hostname>:9001
3. In admin console go to platform🡪 update
4. Project data settings select solrsearchrestrition and click on update.

Configuration steps for the Extension

1. Login to the HMC.
2. Go to catalog🡪 products🡪 search🡪select catalog version as powertoosproduct catalog version as staged🡪 click on search.
3. Select the product and go to administration tab🡪 got to unbound section🡪 B2BunitLink🡪created b2bunit product link.
4. Follow the same steps for the products to which b2bunit have to restrict.
5. Synchronize the products.
6. Go to HMC🡪 System🡪 Facet Search🡪 indexer operation wizard🡪 set the properties as shown below screen shoot.



1. Click on start. Wait until the index finish.
2. Click on done.
3. Go the store front and login using B2BCustomer of the restricted B2BUnit? If the product is displayable is true. Then products will display in the store front.
4. Appendix A *– Glossary of terms*

|  |  |
| --- | --- |
| **Term/Abbreviation/Acronym** | **Meaning** |
| Solr | Solr is an open source enterprise search platform used for text based search in Hybris |
| HMC | Hybris Management Conole. |
| HAC | Hybris Administration console. |
| B2B | Business to Business |