Apache Ofbiz
Code Ispection Document

Niccolo' Raspa, Matteo Marinelli

January 20, 2017

Software Engineering 2 Course Project

Contents

1 Apache OFBiz

The class to be inspected belongs to Apache OFBiz®, an open source product for the automation of enterprise processes that includes framework components and business applications for ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), E-Business / E-Commerce, etc.

It provides a suite of applications that are useful to integrate and automate most business processes of an enterprise.

1.1 Class Assigned

The assigned class is the following:

- SolrProductSearch.java

The class is located in the org.apache.ofbiz.solr package of the Apache OFBiz project. It belongs to the spcial purpose stack together with other secondary functions suchs as eBay Integration, Google Base Integration, Google Checkout Integration, POS System, Project Management and Scrum Management) and it's devoted to search.

The solr component includes an OFBiz service-based wrapper layer to the Apache Solr webapp queries as well as the native Apache Solr web interface itself.

1.2 Documentation

- **Apache Solr** http://lucene.apache.org/solr/
- **Apache OfBiz** https://ofbiz.apache.org/documentation.html
- **Integration of Solr in Ofbiz** https://cwiki.apache.org/confluence/display/OFBIZ/Search+Integration

1.3 Apache Solr

This OFBiz component leverages Apache Solr indexing capabilities. Apache Solr is a fast open-source Java search server. Solr enables you to easily create search engines which searches websites, databases and files. Currently, the solr component focuses on Product data.

1.3.1 Data Indexing

The solr component indexes data such as Products into the Apache Solr database using services defined in the file: servicesdef/solrservices.xml The initial indexing may need to be performed or scheduled manually, but subsequent indexing

may be partially or fully automated, though automated methods are disabled by default and must be enabled.

There are two methods for indexing data:

`* Index rebuilding service (rebuildSolrIndex):`

The rebuildSolrIndex is the most important data import service. It reindexes all OFBiz Products existing in the system into the solr index. rebuildSolrIndex MUST be run AT LEAST once after installation and also following any data load operation that loads new products.

Once the initial indexing has been performed, one can then use the Job Scheduler to invoke rebuildSolrIndex on a regular basis (every hour, every midnight, etc.) to update the Solr index.

`* ECAs/SECAs (addToSolr, for Product data):`

Although the rebuildSolrIndex is always necessary for the initial data import, one may also use ECAs and SECAs to import subsequent data changes automatically at every individual data (e.g. Product) update instead of running rebuildSolrIndex periodically. This is done by defining ECAs or SECAs that trigger the addToSolr service.

The addToSolr service simply accepts a single "instance" parameter, a GenericValue. At the time of this writing, any entity value having a valid "productId" field designating a Product value may be passed; this will trigger reindexing for the specific product.

1.3.2 Data Querying

Solr queries can be done using two methods:

*** Solr OFBiz services:**

Simply invoke (manually or in code) the query services found in the file: servicesdef/solrservices.xml These include solrProductsSearch, solrKeywordSearch and others. Note that in general, solr services can only successfully run in contexts where the solr webapp is loaded and accessible.

*** Solr native admin webapp interface:**

One can also perform native Solr queries and diagnostics using the standard admin interface, accessible as described under Configuration. Please refer to the Apache Solr documentation for usage of this interface.

1.4 Functions

*** Adds product to solr, with product denoted by productId field in instance attribute**

public static Map<String, Object> addToSolr(DispatchContext dctx, Map<String, Object> context) throws GenericEntityException {

*** Adds product to solr index.**

public static Map<String, Object> addToSolrIndex(DispatchContext dctx, Map<String, Object> context) throws GenericEntityException

*** Adds a List of products to the solr index.**

public static Map<String, Object> addListToSolrIndex(DispatchContext dctx, Map<String, Object> context) throws GenericEntityException

*** Runs a query on the Solr Search Engine and returns the results.**

public static Map<String, Object> runSolrQuery(DispatchContext dctx, Map<String, Object> context)

*** Performs solr products search.**

public static Map<String, Object> productsSearch(DispatchContext dctx, Map<String, Object> context)

*** Performs keyword search.**

public static Map<String, Object> keywordSearch(DispatchContext dctx, Map<String, Object> context)

*** Returns a map of the categories currently available under the root element.**

public static Map<String, Object> getAvailableCategories(DispatchContext dctx, Map<String, Object> context)

*** Return a map of the side deep categories.**

public static Map<String, Object> getSideDeepCategories(DispatchContext dctx, Map<String, Object> context)

Rebuilds the solr index.

public static Map<String, Object> rebuildSolrIndex(DispatchContext dctx, Map<String, Object> context) throws GenericEntityException {

2 Code Inspection

3 Effort Spent

In this section you will include information about the number of hours each group member has worked towards the fulfillment of this deadline.