Rogelio Alonso

MENTORS:  Masoud Sadjadi & Juan Caraballo

virtual job fair v-5.0 Installation manual

CONTENTS

[Virtual Job Fair Project 2](#_Toc417046134)

[Selenium IDE 2](#_Toc417046135)

[Solr Server 2](#_Toc417046136)

[Installation 2](#_Toc417046137)

[Configuration 2](#_Toc417046138)

[Starting and Stopping the Solr 3](#_Toc417046139)

[Reloading Configurations for the Solr Server 4](#_Toc417046140)

[Importing Data from MySQL Database 4](#_Toc417046141)

[Indexing with Solr 6](#_Toc417046142)

[Querying Solr Index 6](#_Toc417046143)

[Appendix 8](#_Toc417046144)

[solr.xml 8](#_Toc417046145)

[data-config.xml 10](#_Toc417046146)

# Virtual Job Fair Project

Instruction on how to install the virtual job fair project… (Coming Soon)

# Selenium IDE

In this section we will cover how to install Selenium IDE plugin for Mozilla Firefox browser.

1. Open a browser and navigate to <http://www.seleniumhq.org/download/>
2. Download the Selenium IDE (currently version 2.9.0) which is the \*.xpi and install it as a Firefox plugin.
3. Under Options->Options you can activate the developer tools .
4. An in the same window under Selenium IDE extensions, we can browse to were our JavaScript file is containing the data for the data driven tests. (Usually in the project folder there is a compressed file under ..\Code\Test\Student Actor\datasource.js)

Note that our GitHub repository us public and any sensitive data inside this file can be seen. What we are currently been doing is compressing the datasource.js file and putting a password on it. The password can be asked to your product owner.

# Solr Server

Solr is a search server that provides a rich REST API and runs under Apache Lucene project to provide a fast way to search by indexing large amount of content.

Below are the necessary steps to get Solr Server 5.0 up and running, including what is necessary to configure and run it in a linux or windows environment.

## Installation

Make sure your OS has a Java version 1.7 or higher. Independently from your OS, Solr can be downloaded from any of the links at:

<http://www.apache.org/dyn/closer.cgi/lucene/solr/5.1.0>

After downloading Solr, you will have to unzip the file and copy the folder to your project (Note that this is already done for you and have been uploaded to GitHub). This means, that whenever you clone your project the Solr server is already included.

This completes the installation of Solr.

## Configuration

Solr consist of Cores. Each core has a different configuration and indexes the data depending of this configuration. In order to create a Core in Solr, 4 files must be manually created as followed:

1. Navigate to your project folder. This is the folder where you copied the solr folder extracted on previous topic.
2. Notice that in this folder you have a file named: solr.xml. This file in the main entry to all the cores in your server. It also specifies the port to access your Solr server request via HTTP. By default Solr uses port 8983, and it needs to be changed, this is the place to do it. See appendix for all the lines.
3. Now, navigate to ..\solr-5.0.0\server\solr and create a folder with the name of the core you want to create.
4. Inside the above created folder we must create a file to specify the name and properties of the core we are creating, named core.properties with the following lines.
   * #Written by Rogelio Alonso
   * #Wed Apr 17 2015 at 11:34 AM EDT
   * name= “the name of Core”
   * config=solrconfig.xml
   * schema=schema.xml
   * dataDir=data
5. As you can see, the Solr Core will depend of the configuration file (solrconfig.xml), the schema file (schema.xml) and the data directory.
   * The solrconfig.xml is a configuration file and will tell the core what are the dependencies needed for the data handlers and the location of the respective jar files.
   * The schema.xml is another configuration file and will tell the core the types of data that will be indexed as well as what will be the indexing key, among much other stuff.
   * The data (which must be also manually created), is a directory where Solr will construct and maintain the search index.
6. So, now we must create three folders on the directory where you created the core.properties file.
   * conf
     1. This is where schema.xml and solrconfig.xml will be created.
   * data
     1. This is where Solr will build its index.
   * lib
     1. This is where any required library for the core will be copied.
7. Navigate to the conf folder (created above) and create the files named schema.xml and solrconfig.xml (a copy of these files can be made of the basic\_config folder provided by Solr)
8. Now we need to update the schema.xml
   * Every filed you want to index with solr should be declared here with the name, type, if you want it index or stored. i.e. <field name="title" type="string" indexed="true" stored="true"/>
9. Congratulations you have just configured your Solr Core instance.

## Starting and Stopping the Solr

Now, this depends on what OS you are using, so I will discuss about Linux and Windows, which are the ones I consider the most important ones.

#### Windows

1. Make sure you have added the “java” as an environmental variable. (If not please read this tutorial in how to do so: <https://docs.oracle.com/javase/tutorial/essential/environment/paths.html>)
2. Open a Command Prompt window and navigate to ..\solr-5.0.0\server
3. Type java –jar start.jar and press Enter
4. Open a browser and navigate to <localhost:8983/solr> to make sure your server is running. You can also select your core name from the core drop down menu and make sure is running with no problems.
5. To stop the server simply press Ctrl+C or close the Command Prompt.

#### Linux

Since most of linux is done from SSH terminal, you want to make sure that whenever you exit your terminal your server is still running. We will accomplish this with the help of “nohup”. Moreover, we will need an alias for the process in case we want to stop it, it will be easier for us to identify it.

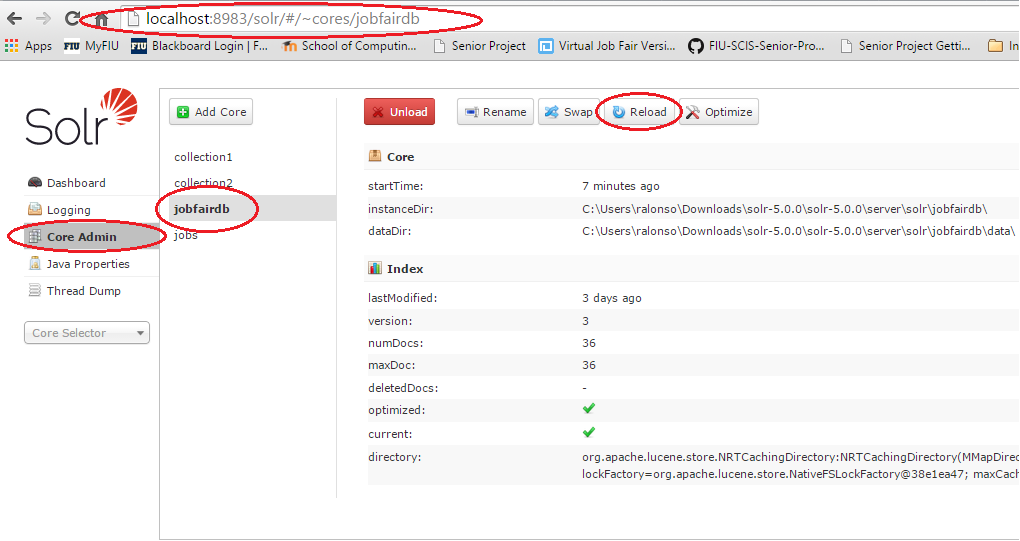
1. Navigate to ..\solr-5.0.0\server
2. Type sudo nohup java –Dsolr –jar start.jar & and press Enter
3. To stop the server you will need to look for the process and kill it with any of following commands:
   * sudo pkill -f Dsolr
   * sudo pkill -9 -f Dsolr
   * Or you can list the processes by jps –l and then kill $PID

## Reloading Configurations for the Solr Server

Note that you do not need to start and stop the server every time you make an update in the schema or the configuration file. Solr provides a graphic way to manage your cores as follows:

1. Open a browser and navigate to <http://localhost:8983/solr> or your server i.e. <http://vjf-dev.cis.fiu.edu:8983/solr>
2. Click on Core Admin
3. Select your Core
4. Click on Reload

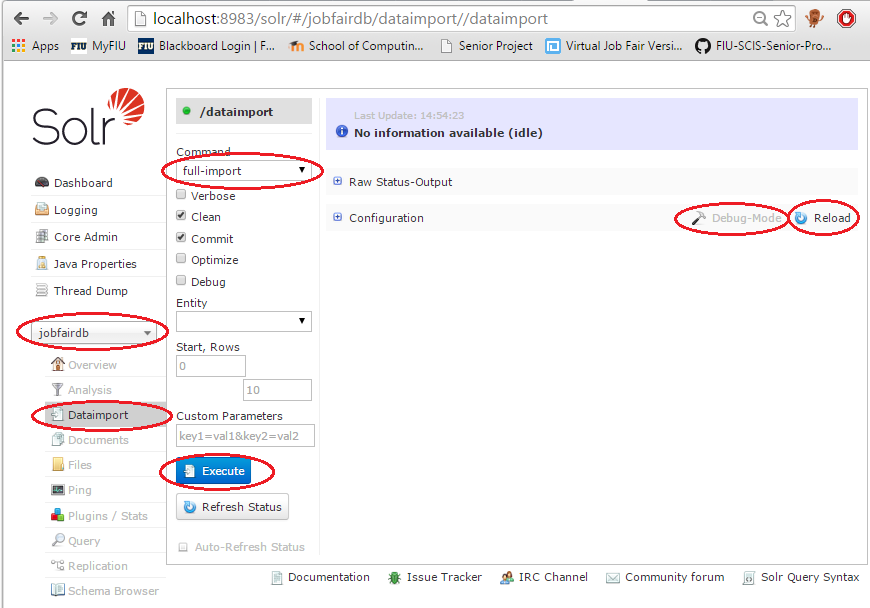
This will reload all the files needed and will restart the Core.



## Importing Data from MySQL Database

Initially it is necessary to import data to be indexed. This data is the desired tables on the jobfairdb database in MySQL. I will now document how to integrate and import data from the MySQL database into Solr and how to query the data.

1. Download mysql-connector-java-\*-bin.jar (version 5 and up) from Oracle web site: <http://dev.mysql.com/downloads/connector/j/>
2. Copy the appropriate mysql-connector-java-5.1.35-bin.jar in the lib folder created in the configuration process.
3. Copy the solr-dataimporthandler-5.0.0.jar and solr-dataimporthandler-extras-5.0.0.jar from the ..\solr-5.0.0\dist folder to the lib folder above mentioned. This two jar files will use the MySQL connector jar to access and retrieve rows from your MySQL database.
4. Add the following lines to the solrconfig.xml file:
   * Before the <dataDir></dataDir> tag add:
     + <!-- Dependency Libraries-->
     + <lib dir=**"lib"** regex=**".\*\.jar"** />
     + <lib dir=**"lib"** regex=**"solr-dataimporthandler-\d.\*\.jar"** />
     + <lib dir=**"lib"** regex=**"mysql-connector-java-\d.\*\.jar"** />
   * At the end of the file add:
     + <requestHandler name=**"/dataimport"** class=**"org.apache.solr.handler.dataimport.DataImportHandler"**>
     + <lst name=**"defaults"**>
     + <str name=**"config"**>**data-config.xml**</str>
     + </lst>
     + </requestHandler>
5. Create a file name data-config.xml to save the queries to be run to MySQL Database for full and incremental (Delta) indexing as example in the Appendix.
   * To touch up upon this file. Here is where you want to map your MySQL Database fields to those defined on your schema.xml. This file will contain not only the information for the type of query you want run to your database, either Delta/Incremental or Full Import; but also, the link between data types and field names.
6. Open a browser and navigate to <http://localhost:8983/solr> or your server i.e. <http://vjf-dev.cis.fiu.edu:8983/solr>
7. Select your Core
8. Click on Dataimport
9. Select the full-import Command
10. Click Execute
11. After execution you can see how many documents or rows where imported into the index.



If you want to make changes to your schema.xml file you can do so on the Debug-Mode button and then reload the schema.xml file on the Reload button.

## Indexing with Solr

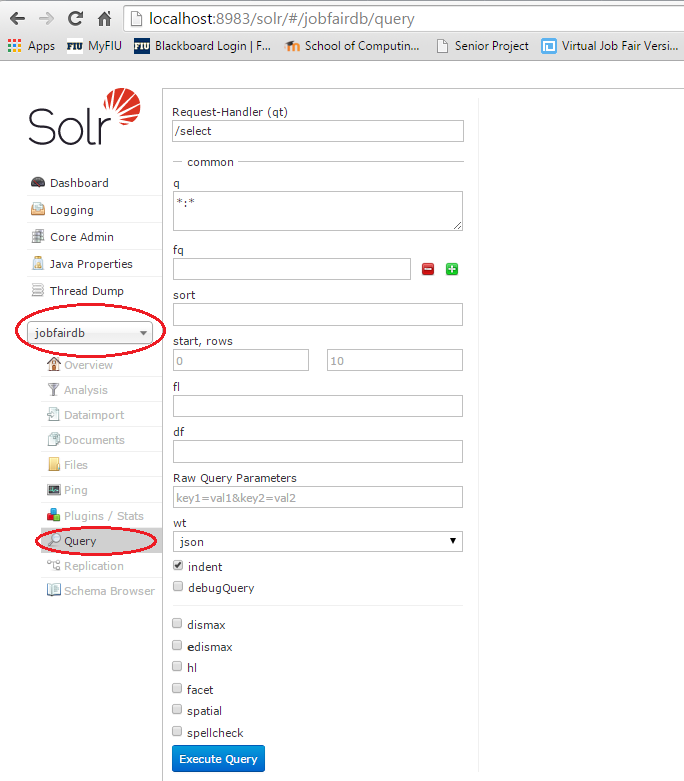
Solr REST API accepts document in the JSON, XML, PDF and a number of different formats to be indexed via POST HTTP request. I strongly recommend to read this page for information on how to index new documents.

<http://lucene.apache.org/solr/quickstart.html>

## Querying Solr Index

To test your imported data or newly indexed data you can navigate in a browser window to:

1. <http://localhost:8983/solr> or your server i.e. <http://vjf-dev.cis.fiu.edu:8983/solr> depending where you installed the server.
2. Then select your core
3. Click on Query
4. Click Execute Query



There is a variety of features here in this page such as the format you want solr to return your data or what fields you actually want or the typo of query you want to run (SELECT/UPDATE/etc), or ever if you want to add facets to your results and so forth. For more information about how to query solr please download this pdf:

<https://www.apache.org/dyn/closer.cgi/lucene/solr/ref-guide/>

# Appendix

## solr.xml

<?xml version="1.0" encoding="UTF-8" ?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

-->

<!--

This is an example of a simple "solr.xml" file for configuring one or

more Solr Cores, as well as allowing Cores to be added, removed, and

reloaded via HTTP requests.

More information about options available in this configuration file,

and Solr Core administration can be found online:

http://wiki.apache.org/solr/CoreAdmin

-->

<solr>

<solrcloud>

<str name="host">${**host:**}</str>

<int name="hostPort">${**jetty.port:8983**}</int>

<str name="hostContext">${**hostContext:solr**}</str>

<bool name="genericCoreNodeNames">${genericCoreNodeNames:true}</bool>

<int name="zkClientTimeout">${zkClientTimeout:30000}</int>

<int name="distribUpdateSoTimeout">${distribUpdateSoTimeout:600000}</int>

<int name="distribUpdateConnTimeout">${distribUpdateConnTimeout:60000}</int>

</solrcloud>

<shardHandlerFactory name="shardHandlerFactory"

class="HttpShardHandlerFactory">

<int name="socketTimeout">${socketTimeout:600000}</int>

<int name="connTimeout">${connTimeout:60000}</int>

</shardHandlerFactory>

</solr>

## data-config.xml

<?xml version = "1.0" encoding = "UTF-8" ?>

<dataConfig>

<dataSource type="JdbcDataSource" driver="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/jobfairdb" user="root" password="" batchSize="1" />

<document name="jodfairdb">

<entity name="solr"

query="SELECT \* FROM solr"

deltaImportQuery= "SELECT \* FROM solr WHERE id = '${dih.delta.id}'"

deltaQuery = "SELECT id FROM solr WHERE modified > '${dih.last\_index\_time}'">

<field column="id" name="id"/>

<field column="username" name="username"/>

<field column="email" name="email"/>

<field column="registration\_date" name="registration\_date"/>

<field column="first\_name" name="first\_name"/>

<field column="last\_name" name="last\_name"/>

<field column="image\_url" name="image\_url"/>

<field column="type" name="type"/>

<field column="title" name="title"/>

<field column="post\_date" name="post\_date"/>

<field column="deadline" name="deadline"/>

<field column="description" name="description"/>

<field column="compensation" name="compensation"/>

<field column="other\_requirements" name="other\_requirements"/>

<field column="matches\_found" name="matches\_found"/>

<field column="posting\_url" name="posting\_url"/>

<field column="comp\_name" name="comp\_name"/>

<field column="poster\_email" name="poster\_email"/>

<entity name = "student\_skills" query= "SELECT skillid FROM student\_skill\_map WHERE student\_skill\_map.userid = '${solr.id}'">

<entity name = "skills" query= "SELECT name FROM skillset WHERE skillset.id = '${student\_skills.skillid}'">

<field column="name" name="skillname" />

</entity>

</entity>

</entity>

</document>

</dataConfig>