|  |
| --- |
| www.missionarogya.org |
| Getting started with SHERP |
| Environmental Setup Document |
|  |
| **Sumon Sadhukhan** |
| **7/12/2015** |

|  |
| --- |
| The document contains the outline of environment setup, required for Sherp developer |

Table of Contents

[Overview 2](#_Toc424429506)

[Assumption 2](#_Toc424429507)

[1. Eclipse Setup 3](#_Toc424429508)

[*Configuration of Git* 3](#_Toc424429509)

[*Configuration of Maven* 5](#_Toc424429510)

[*Tomcat Configuration* 7](#_Toc424429511)

[*JDK configuration* 8](#_Toc424429512)

[*Database Configuration* 8](#_Toc424429513)

[2. Build and Deploy steps 10](#_Toc424429514)

[3. Un-deploy a war packet from server 11](#_Toc424429515)

[4. Pushing changes to Git. 11](#_Toc424429516)

[5. Sync the local code base with remote repository 13](#_Toc424429517)

[6. Conclusion 14](#_Toc424429518)

[References 14](#_Toc424429519)

Introduction

The document explains the setup procedure to start a web application. Required software and tools should be kept handy before starting up the procedure. The softwares, used in the application, are open source and easily available on internet.

# Overview

Below tools have been chosen to develop the application. The document gives an overview on each of the tools to be setup in Window Environment.

|  |  |
| --- | --- |
| Framework | Spring MVC |
| Database | Postgres (version 9.4.4) |
| Application Server | Tomcat 7.0.62 |
| Repository | Maven |
| Version Control | Git |
|  |  |
| IDE | Eclipse Luna |
| Java | JDK 1.7.79 |

# Assumption

The developer should know how to setup the classpath in the windows environment. After installation of JDK and maven, the environment variable must be added in Classpath. [Click here](http://stackoverflow.com/questions/14579661/how-to-set-catalina-home-variable-in-windows-7) for more reference.

Developer should have high speed internet connection.

Getting start

For setting up the local environment you should download softwares mentioned below.

Some plugins need to be added in Eclipse for integrated development environment.

|  |  |
| --- | --- |
| Softwares | Link |
| Eclipse Juno | http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/juno/SR2/eclipse-jee-juno-SR2-win32-x86\_64.zip |
| Tomcat | http://supergsego.com/apache/tomcat/tomcat-7/v7.0.62/bin/apache-tomcat-7.0.62.zip |
| Java | <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html> ( Java SE Development Kit 7u790) |
| Postgres | http://www.enterprisedb.com/products-services-training/pgdownload#windows |

Once the download is complete extract all the zip files in proper location. For mysql it is required to install the software. For this document, following locations have been used.

|  |  |
| --- | --- |
| Eclipse | E:\Eclipse\eclipse |
| JDK 1.7.0 | F:\jdk1.7 |
| Tomcat | E:\tomcat7\apache-tomcat-7.0.62 |

Note: Try to avoid keeping multiple version of JDK in local environment. It may causes conflict sometimes.

# Eclipse Setup

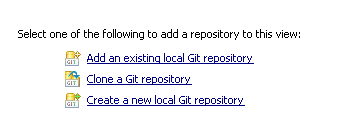
The following steps should be followed to configure Eclipse.

## *Configuration of Git*

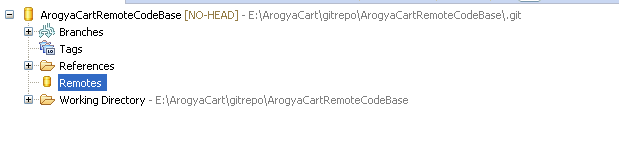
* After extracting eclipse, open the IDE.
* Enter workspace location.
* Open eclipse 🡪 Help 🡪 Eclipse Market space.
* Search with Git in Eclipse market place. (Nodeeclipse Git Addon 1.0.0)
* Once the git installation is successful, switch your eclipse to “Git”



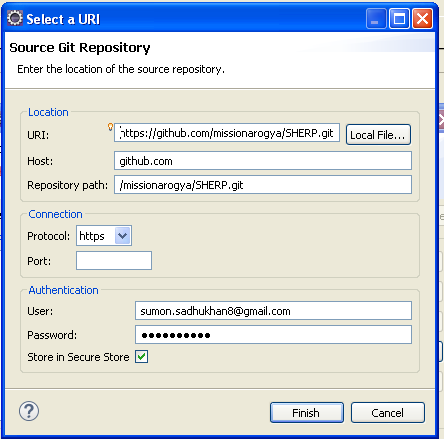
* Create a new local repository selecting the option **“Create a new local repository”**

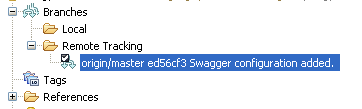


* Enter parent directory and Name on the form and click on finish.

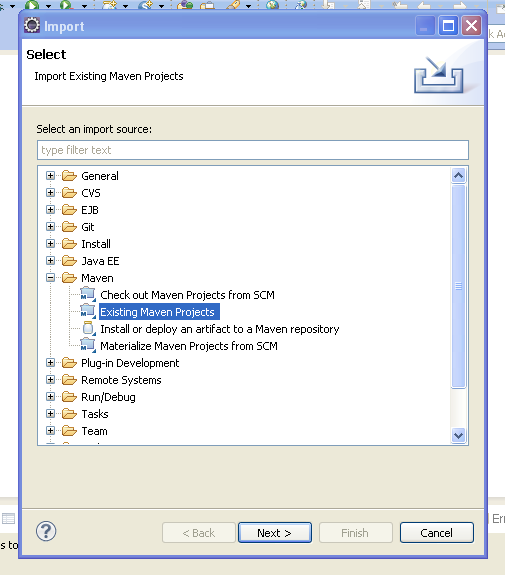
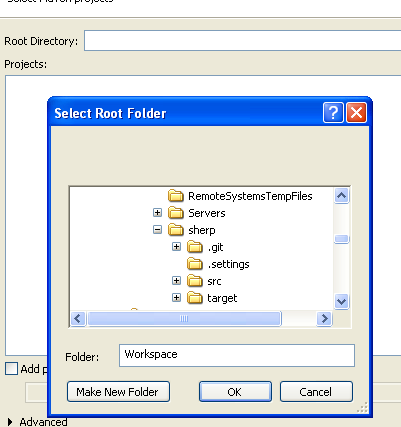


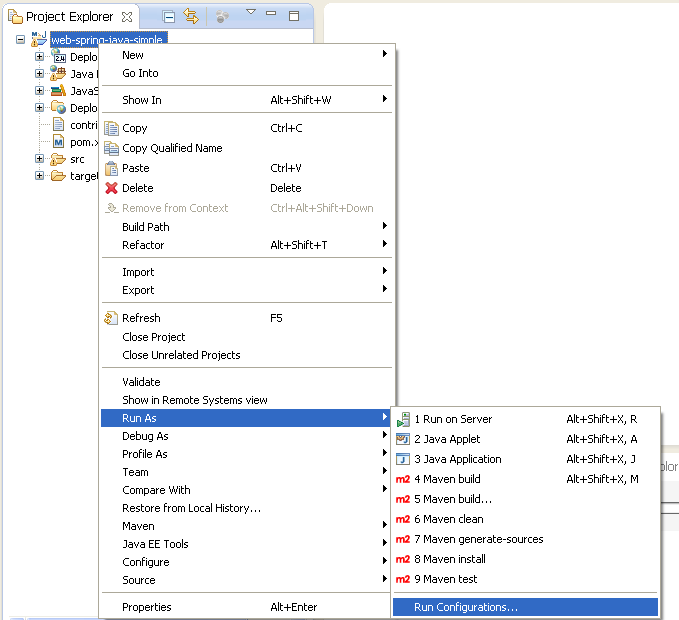
* Once the local repo is created, try to add one new Remote branch. Right Click on Remote 🡪 Create Remote then select **configure fetch**
* Remove existing ref mapping and click on change.
* Copy the git location from github and paste in URI field. The other fields will be filled automatically. Click on Finish.

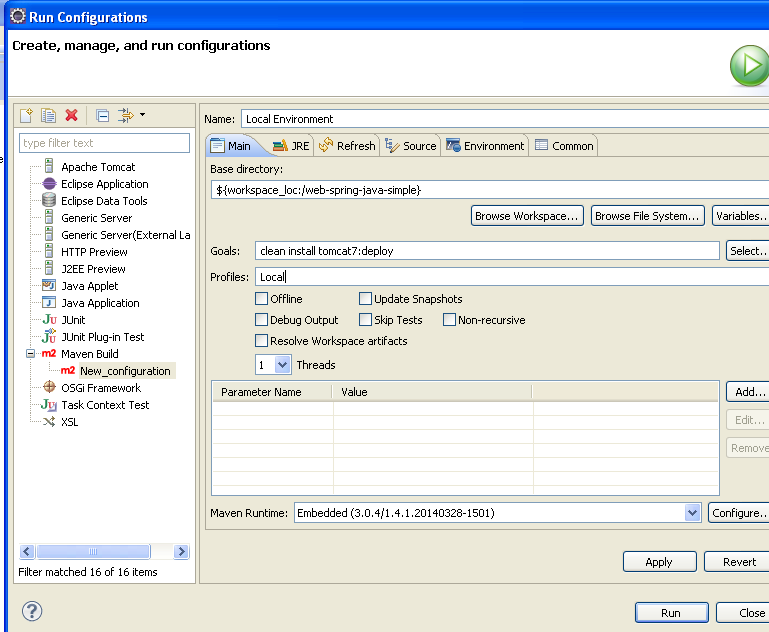


* add a new reference mapping after click on Add.
* Enter master in the branch field and choose master.
* Click on Save and Fetch.
* Click on OK. it will fetch the updated code from github
* Click on remote Tracking and click on checkout. The code will be downloaded in local IDE.
* 
* Once the checkout is completed, the **Working Directory** folder will be updated.
* Switch the workspace J2EE perspective.

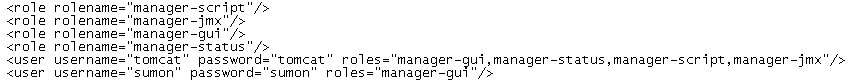
## *Configuration of Maven*

* Open eclipse market place. Install Maven integration plugins from eclipse market place search result.
* Install maven integration plugins for juno. If facing issues with eclipse marketplace for installation of maven plugin for Juno, go to Help->Install New software-><http://download.eclipse.org/technology/m2e/releases> . Try installing the maven plugins from there.
* Once installation is done, Right click on **Project Explorer 🡪 Import 🡪 Maven 🡪 Existing Maven Project**
* 
* Click on Next button then select the Git project checkout location, after click on Browse button.
* 
* It will start analyzing the project pom.xml file.
* Click on next button and Click on finish. It will fetch all the dependency of the project.



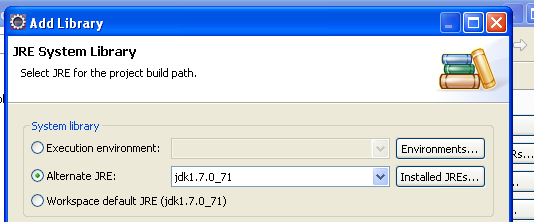
* Go to Run Configuration after right clicking on the project 🡪 Maven 🡪 Run configuration.
* Click on Browse Workspace and select the current project.
* In Goals enter **clean install tomcat7:deploy.**
* Enter **Local** in Profile.
* 
* Click on Apply to save the configuration and then close.

## *Tomcat Configuration*

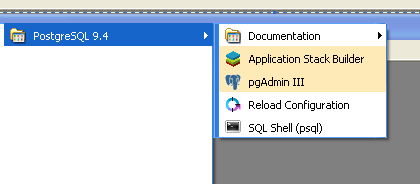
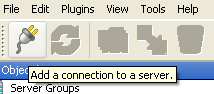
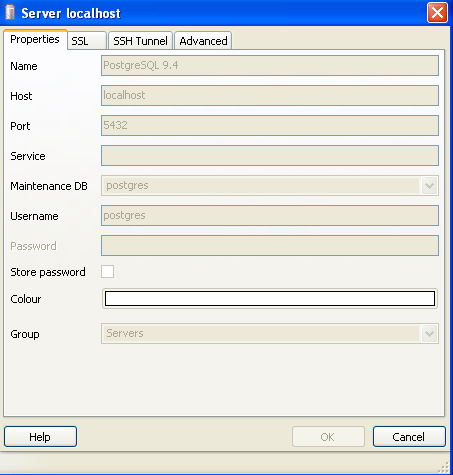
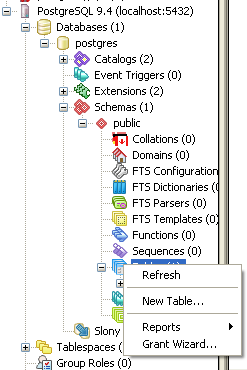
* Extract tomcat7 to your local machine.
* Go to E:\tomcat7\apache-tomcat-7.0.62\conf location.
* Open tomcat-users.xml file.
* Add a new user with below configuration.
* Open command line. Enter the below text and press enter. It will start the server.
* 

Note : Don’t start the server from Eclipse.

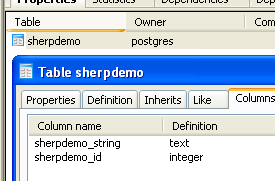
## *JDK configuration*

* Extract JDK 1.7 in some location.
* Select the project and right click on it and goto **Properties**
* Select Java Build path 🡪 Library.
* Add JDK 1.7.0 under JRE System Library.
* 
* Click on finish.

## *Database Configuration*

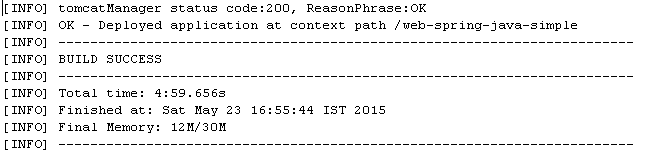
* Install postgres in your vm. (Make sure to take a note of your root password). Check reference section for help.
* Start pgAdmin III.
* 
* Add your local postgres server.
* 
* Fill the form with your local server details.
* 
* Once you click on OK button , the server will be added.
* Create a new table which is used in the same setup.
* 

Add two columns mentioned in the below picture.

* 
* Open application.properties from eclipse, change username/password according to that.

# Build and Deploy steps

Before deploying the code check if tomcat home page is rendering or not by opening the browser(http://localhost:8080).

* Open pom.xml in eclipse.
* Check the username/password for **Local** profile is matching with server configuration or not. If it is not matching change it.
* Again go to **run configuration** (Right Click on the project 🡪 Run As 🡪 Run configuration.
* Select **Local Environment**.
* Check the JRE is selected as JDK 1.6.0 or not. If it is not selected change it to JDK 1.6
* Click on Apply and Run.
* You will receive a successful message once the build is completed.

Note: One of the reasons of Build failure is availability of old war packet in webapps folder. To remove old packet, you have to un-deploy the old packet from server.

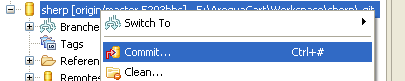
# Un-deploy a war packet from server

* Open the browser, navigate to <http://localhost:8080>
* Click on Tomcat Manager on your left side.
* Enter username/password. (tomcat/tomcat)
* If the war file is already deployed on the server, the application will be displayed on the application list.
* Click on un-deploy.

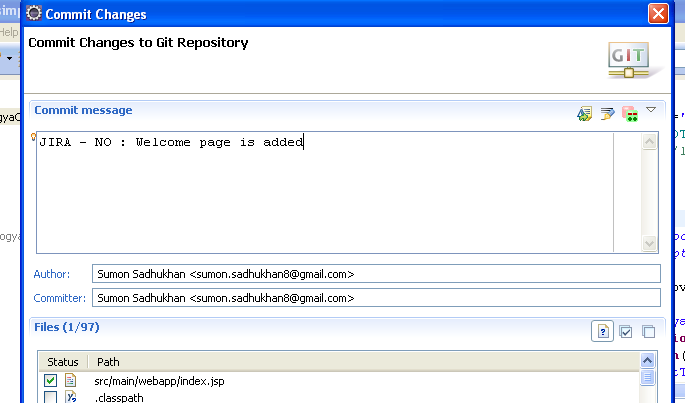
# Pushing changes to Git.

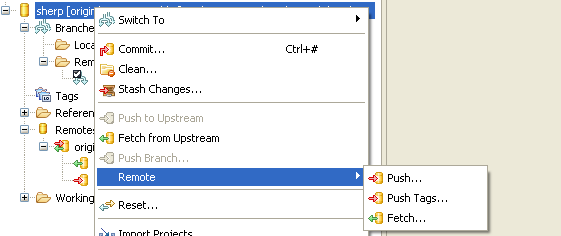
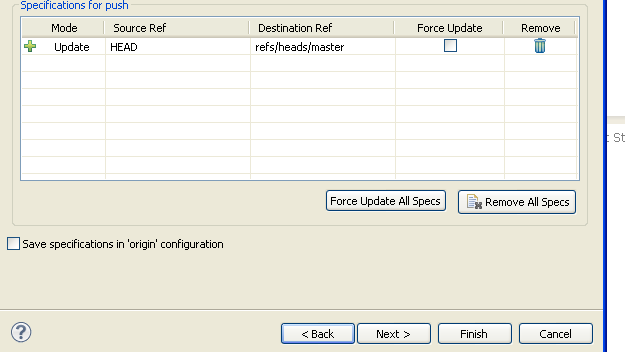
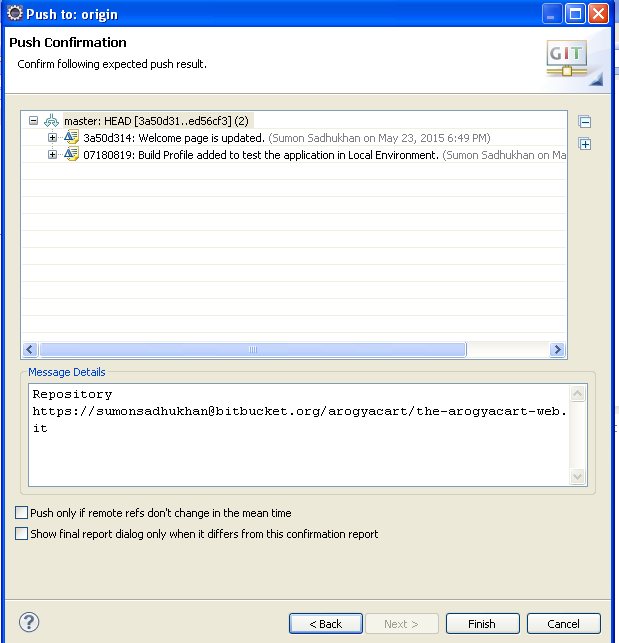
Before committing any change make sure, your change is not breaking any other functionality. Deploy the packet to the local tomcat server. Validate the change.

* Switch to Git Repository Exploring .
* Right click on the local repository, click on **Commit.**
* Select all the files related to particular JIRA. Try to commit all the files together for a particular jira along with a proper comment. The comment should contain the jira number and the remarks. Don’t commit the target folder into the repository.



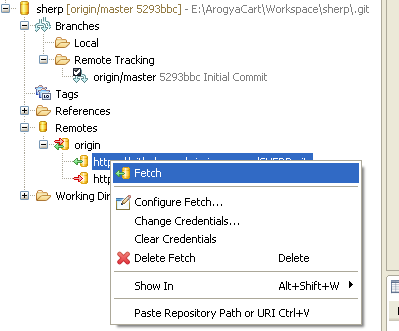
* Now the local git is updated with all the changes. The change can be pushed to remote branch by click on Push.



* Right click on the branch. Click on push.
* 
* Check the remote and click on Next button.
* Check the **specification for push** if it is correct, navigate to next page clicking on next button
* 
* Click on next button and Click on Finish.
* 

Note: Before pushing any change always update your local repository, otherwise it will give error message.

# Sync the local code base with remote repository

* Click on Git Repository Exploring.
* Go to remotes and right click on origin.
* Click on fetch.
* Click on Finish. It will sync the local codebase with remote repo. Sometimes you may face conflict of files. You have to resolve with your changes or remote branch changes.
* 

# Conclusion

You have completed the initial local environment setup. You can go through the links mentioned below to know more about spring, Hibernate & Rest.

# References

1. Class path setup for JDK and tomcat.

[http://stackoverflow.com/questions/14579661/how-to-set-catalina-home-variable-in-windows-7](http://stackoverflow.com/questions/14579661/how-to-set-catalina-home-variable-in-windows-7%20)

1. MySQL installation:

[www.dbasehost.com/tutorials/mysql2.php](http://www.dbasehost.com/tutorials/mysql2.php)

1. Getting started with REST and Spring

<https://spring.io/guides/gs/rest-service/>

1. Getting started with Hibernate

<http://www.tutorialspoint.com/hibernate/hibernate_examples.htm>

1. Getting started with Spring MVC

<http://www.javacodegeeks.com/2012/09/spring-dao-and-service-layer.html>