**BUILD & DEPLOYMENT DOCUMENT (CELEBRATE KARO)**

1. Introduction

This document describes project configuration, how to build and deploy the Celebrate Karo project. It also includes libraries and technologies used in this project.

2. Technical Specifications

* Client Side:
  + HTML (Mixed with JSP)
  + Java Script, Jquery and Ajax
  + CSS
* Server Side:
  + Java (1.7)
  + Spring MVC
  + Hibernate (ORM modal)
  + JSP
* Database:
  + MySQL (Version 6.3)
* Build Tools & Environments:
  + Maven (3.0 or Higher)
  + Eclipse

3. Build Process

The Maven website is located here: [**http://maven.apache.org**](http://maven.apache.org/)

From this website you can download the latest version of Maven.

## What is a Build Tool?

A build tool is a tool that automates everything related to building the software project. Building a software project typically includes one or more of these activities:

* Generating source code (if auto-generated code is used in the project).
* Generating documentation from the source code.
* Compiling source code.
* Packaging compiled code into JAR files or WAR files.
* Installing the packaged code on a server, in a repository or somewhere else.

Any given software project may have more activities than these needed to build the finished software. Such activities can normally be plugged into a build tool, so these activities can be automated too.

The advantage of automating the build process is that you minimize the risk of humans making errors while building the software manually. Additionally, an automated build tool is typically faster than a human performing the same steps manually.

## Installing Maven:

To install Maven on your own system (computer), go to the [**Maven download page**](http://maven.apache.org/download.cgi) and follow the instructions there. In summary, what you need to do is:

1. Set the JAVA\_HOME environment variable to point to a valid Java SDK (e.g. Java 7).
2. Download and unzip Maven.
3. Set the M2\_HOME environment variable to point to the directory you unzipped Maven to.
4. Set the M2 environment variable to point to M2\_HOME/bin (%M2\_HOME%\bin on Windows, $M2\_HOME/bin on unix).
5. Add M2 to the PATH environment variable (%M2% on Windows, $M2 on unix).
6. Open a command prompt and type 'mvn -version' (without quotes) and press enter.

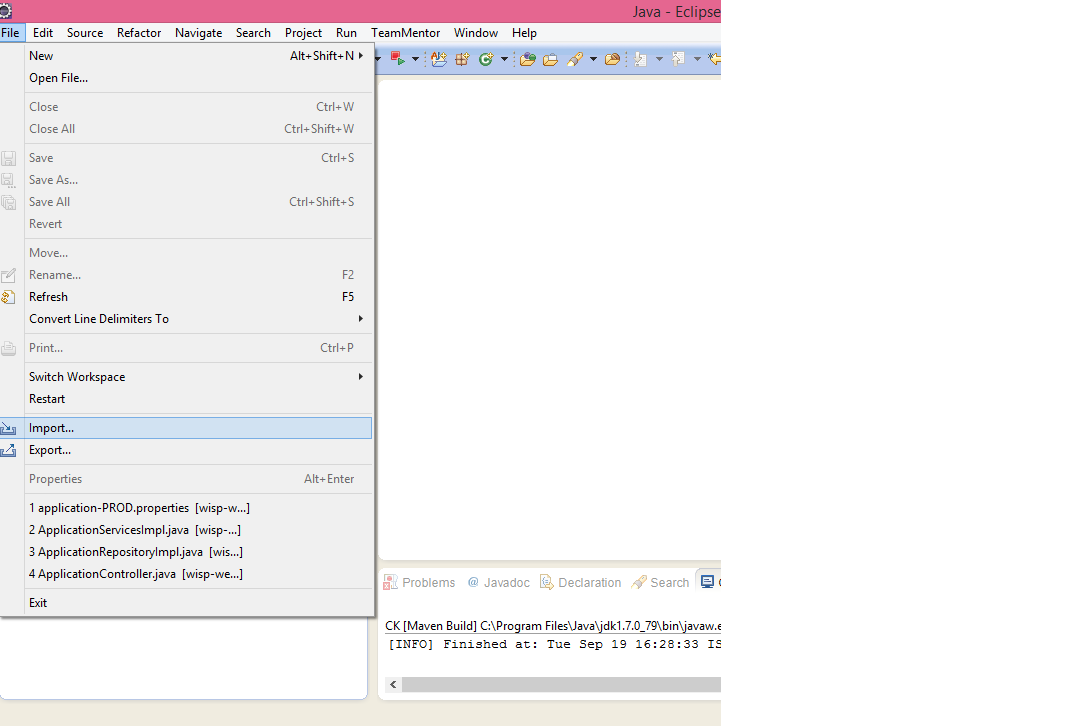
After typing in the mvn -version command you should be able to see Maven execute, and the version number of Maven written out to the command prompt.

Note: Maven uses Java when executing, so you need Java installed too (and the JAVA\_HOME environment variable set as explained above). Maven 3.0.5 needs a Java version 1.5 or later. I use Maven 3.3.3 with Java 7.

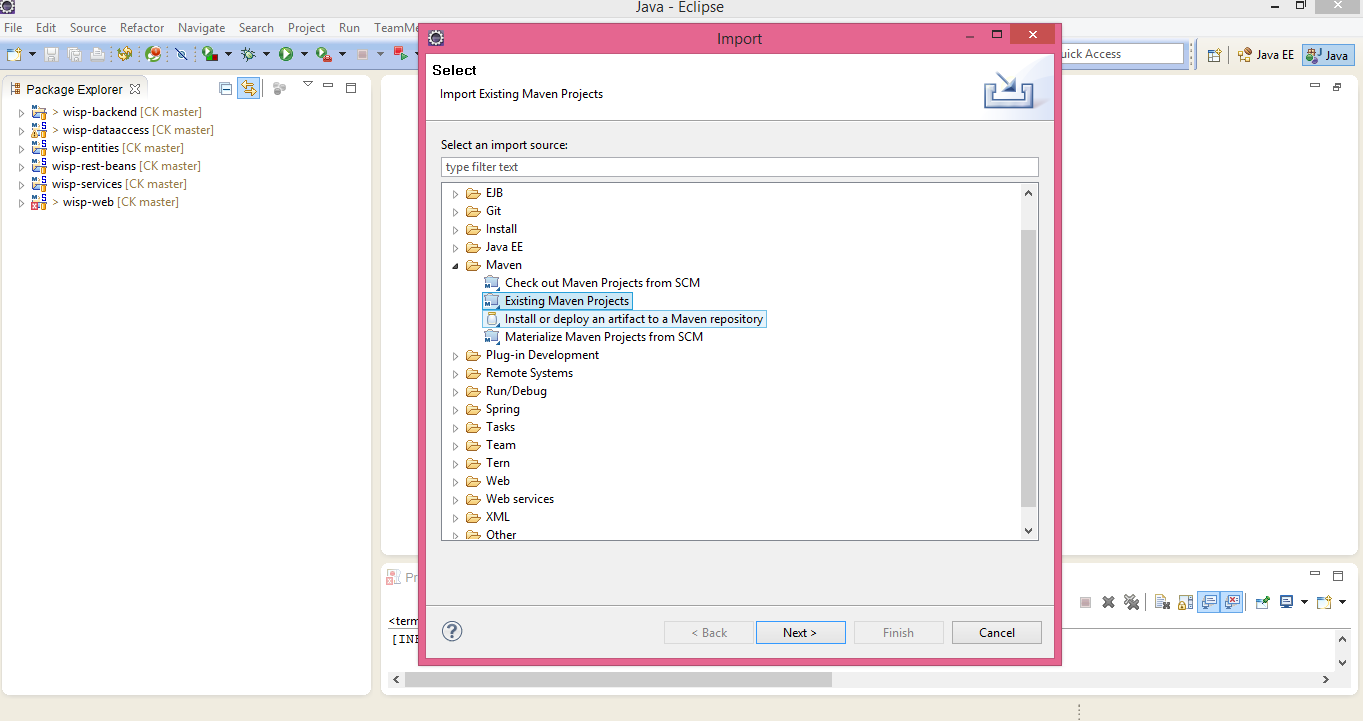
## Import Project:

Un-Zip the project source code and follow the steps below,

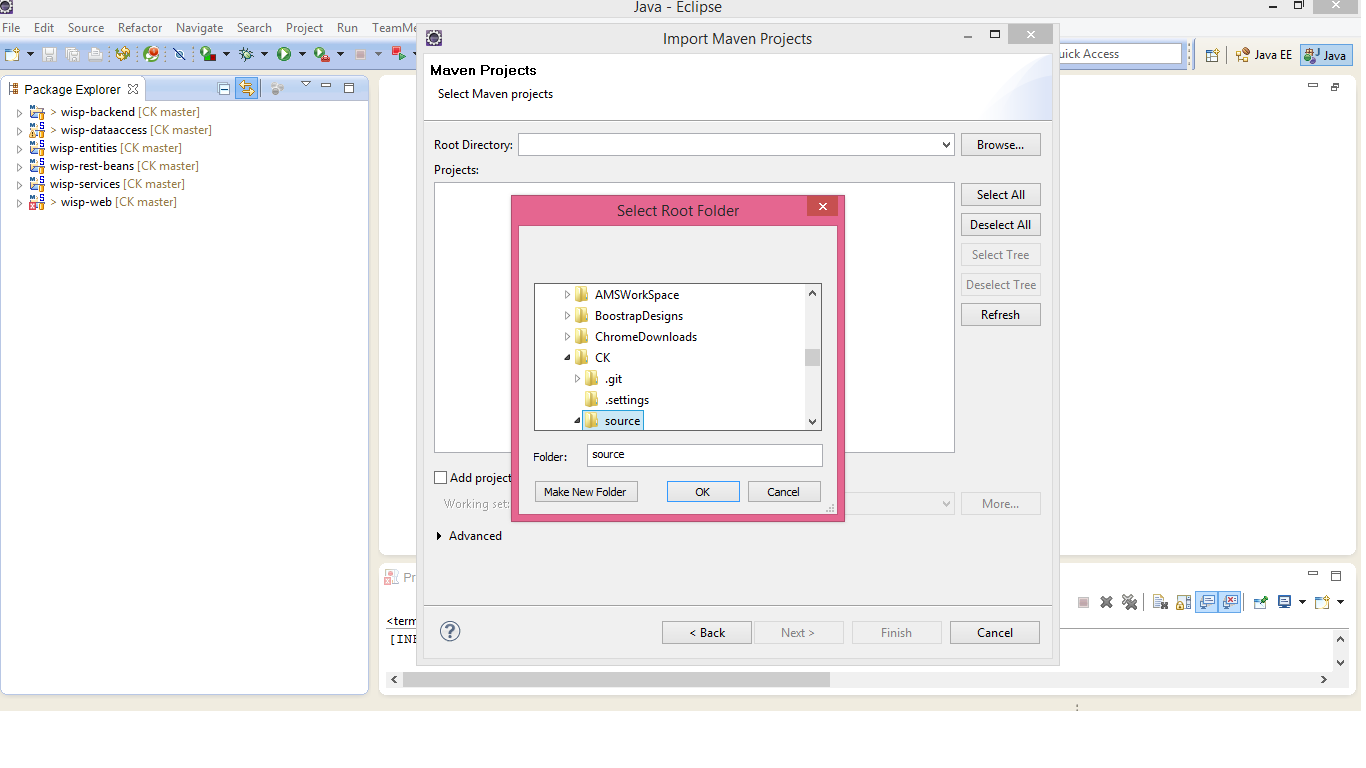
Eclipse --> File --> Import



Under Maven select Existing maven project,

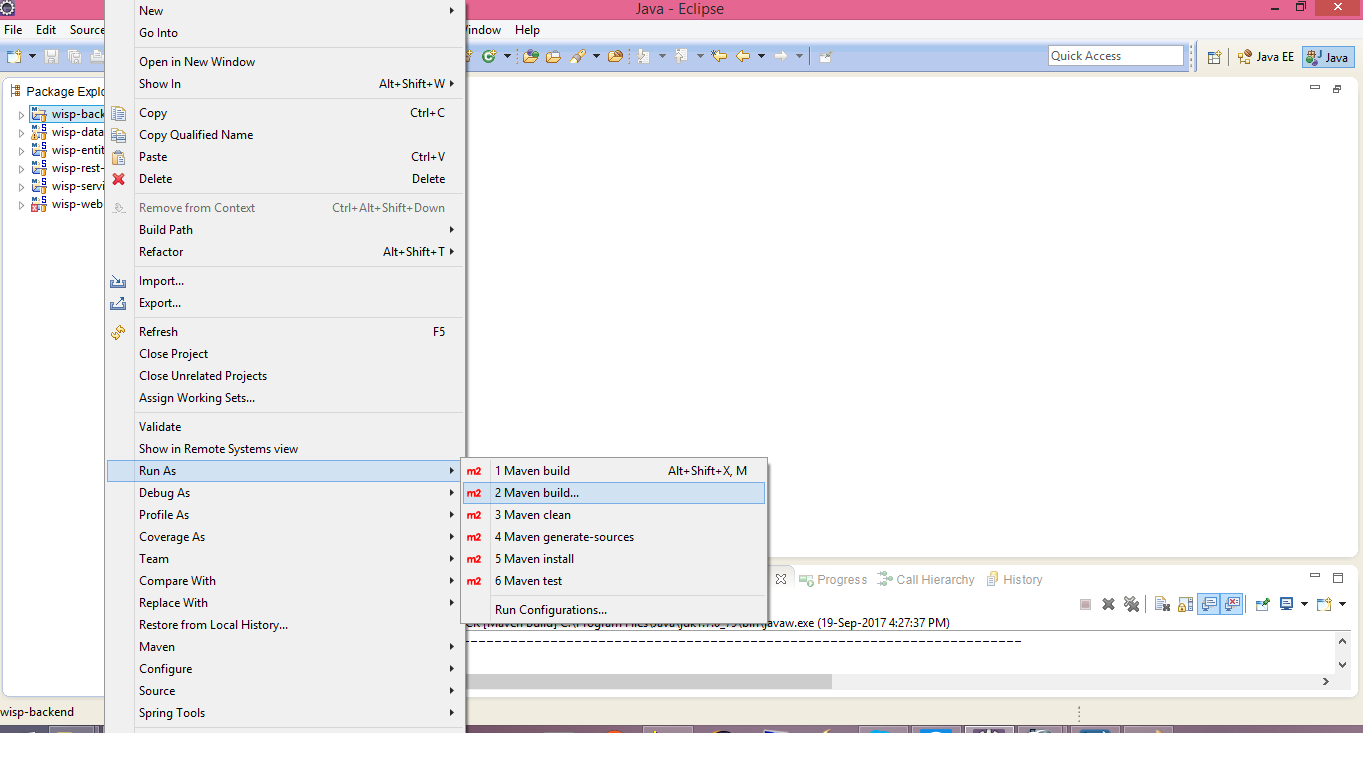


Click on Next, select the source folder from the folder.



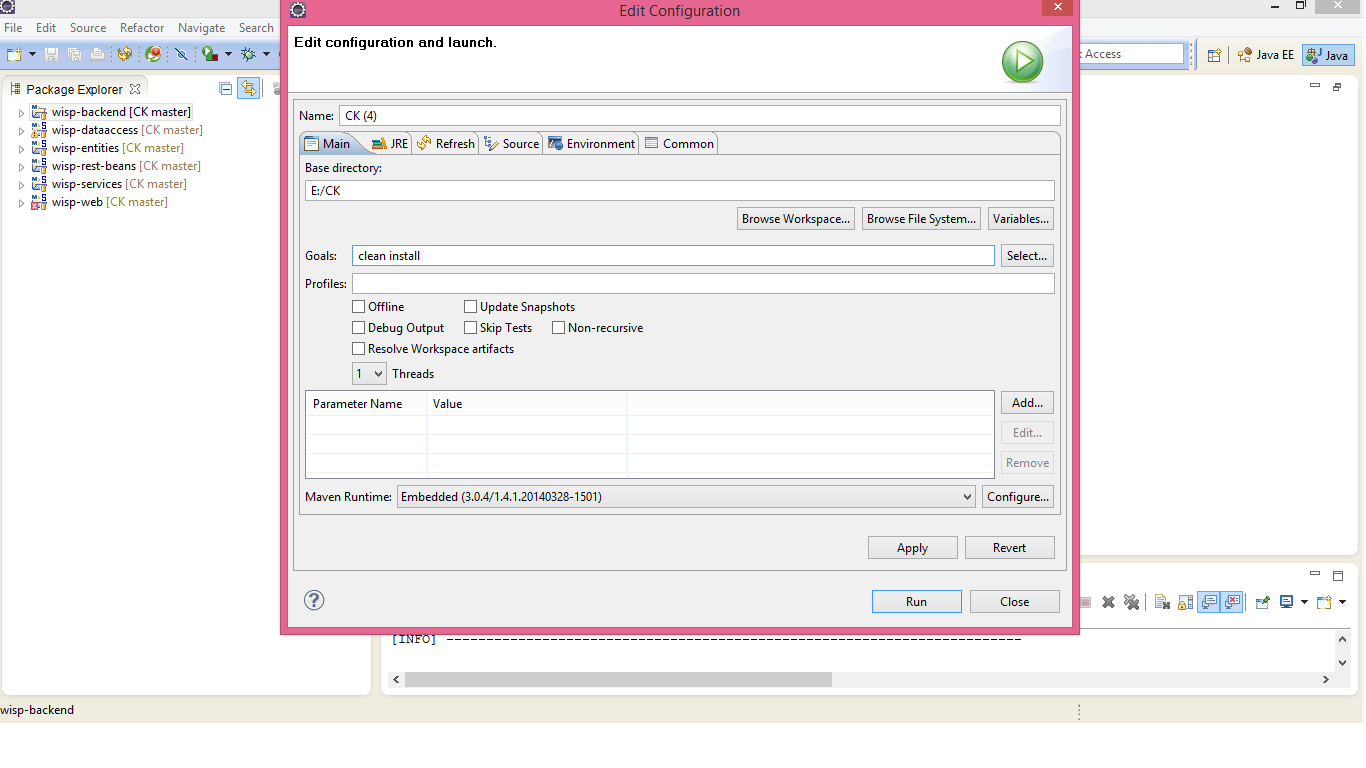
Click to finish.

Then right click on 'wisp-backend' folder and select Run As --> Maven Build,

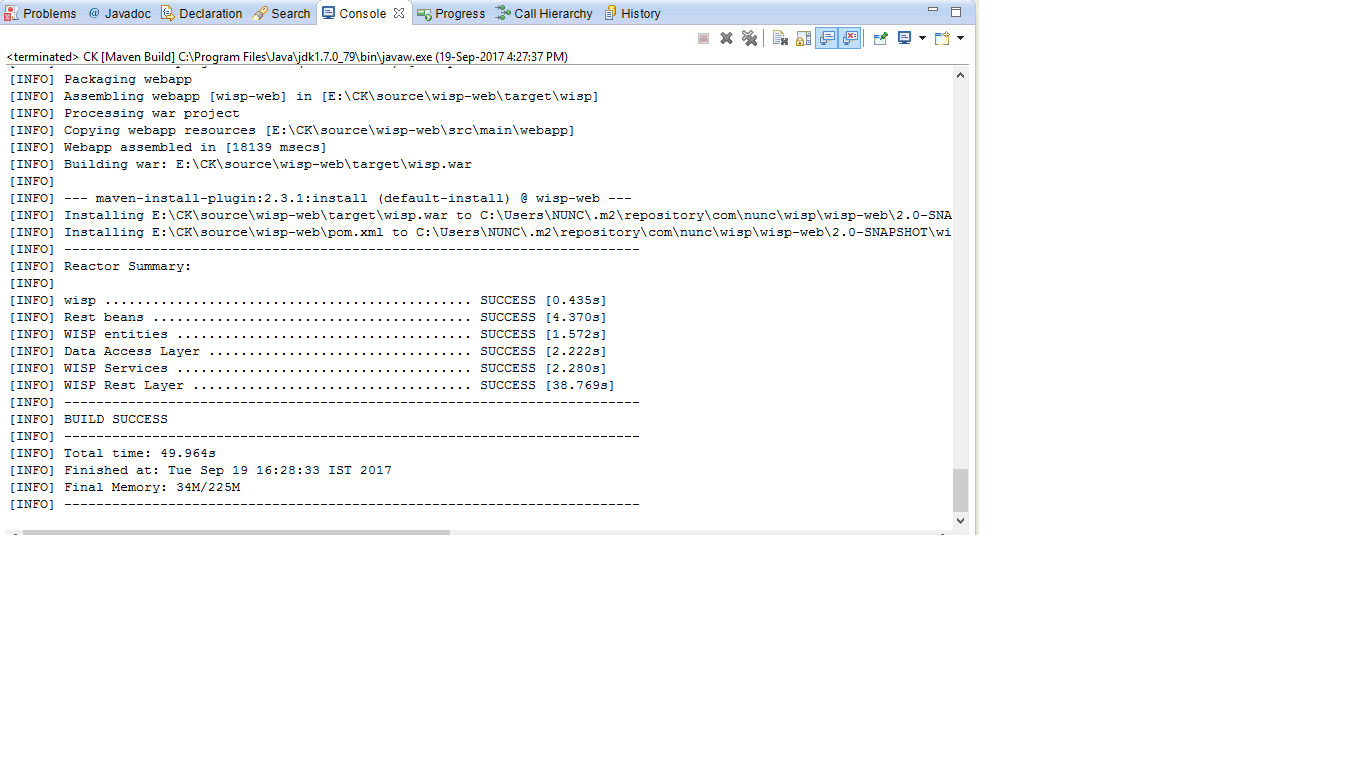


Provide the goals as "clean install" and click on Run.

This command first executes the clean build life cycle, which removes compiled classes from the Maven output directory, and then it executes the install build phase.



If build successful you will see below screen.



Now expand "wisp-web" folder --> Target, under target folder you will find wisp.war file. This is the final build which we want to deploy in server.

4. Deployment:

## Install tomcat server and config data source:

Install Tomcat server 7, for tomcat installation steps please follow the below link,

<http://www.yorku.ca/jhuang/examples/tomcat-install.html>

Config data source for tomcat in server.xml file like below,

<Context path="/wisp" docBase="wisp" crossContext="true" reloadable="true" debug="1">

<Resource name="jdbc/wisp" auth="Container" type="javax.sql.DataSource" maxActive="100" maxIdle="30" maxWait="10000" username="\*\*\*\*\*\*\*" password="\*\*\*\*\*\*\*\*" driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/wisp"/>

</Context>

Note: Please provide the database username and password.

## Environment Variables:

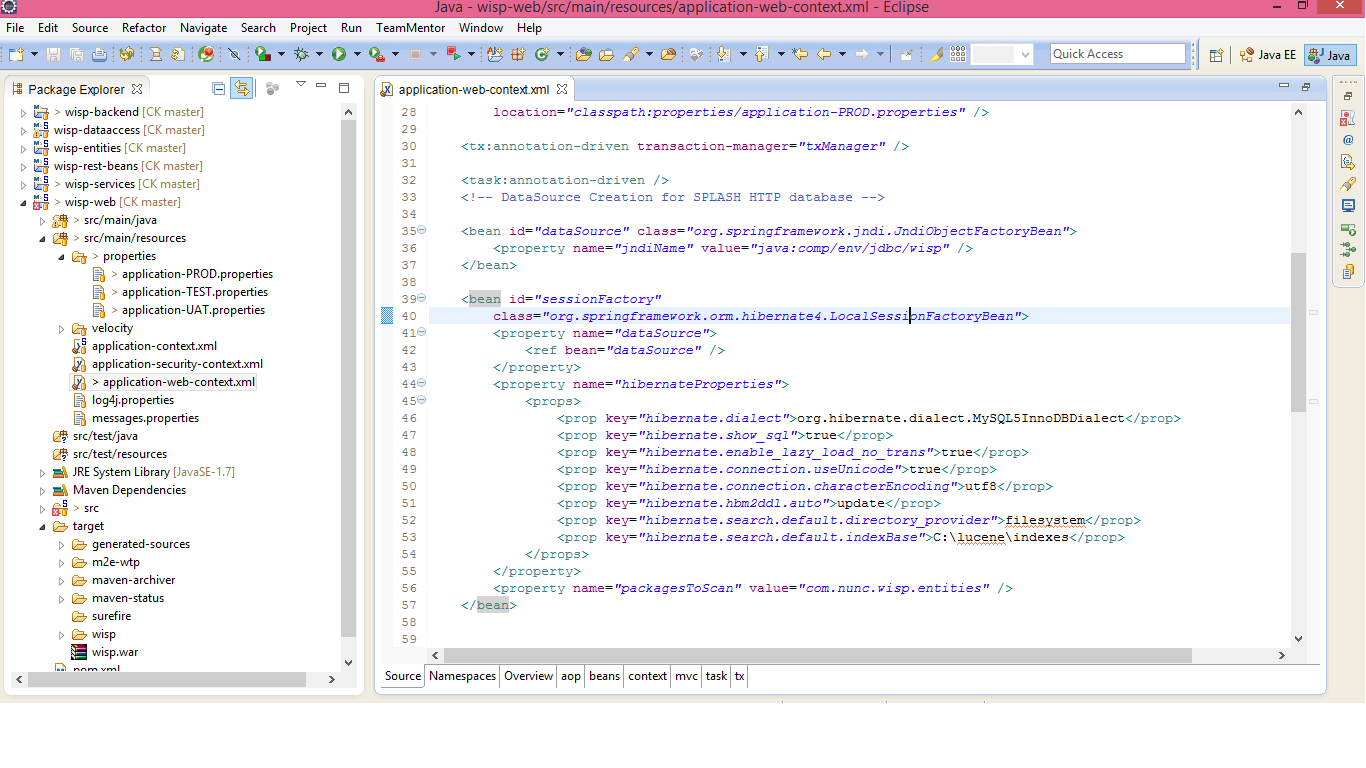
In context.xml specify the environment.

<Environment name="wisp.environment" value="PROD" type="java.lang.String" override="false"/>

This value is to select corresponding properties file based on provided environment.

**Note: In Case of Shared server which doesn't have access to edit configuration file/ don't have JNDI support then remove the data source and environment from server and place it in application configuration file.**

**To find application configuration file, explore wisp-web ---> src/main/resource. Under resource folder you will find 'application-web-context.xml' file**



**Remove the JNDI to modify data source values.**

**To find application properties in same directory check for 'application-PROD.properties' file and update the properties for different environments.**

Finally, to deploy build in server copy wisp. War file from target folder,

## Windows:

Copy the .war file (E.g.: wisp. War) to %CATALINA\_HOME%\webapps (E.g.: C:\tomcat\webapps)

Run %CATALINA\_HOME%\bin\startup.bat

Your .war file will be extracted automatically to a **folder** that has the same name (without extension) (E.g.: wisp)

Go to %CATALINA\_HOME%\conf\server.xml and take the **port** for the HTTP protocol. <Connector port="8080" ... />. The default value is **8080**.

Access the following URL:

[<protocol>://] local host :< port>/folder/

(E.g.: local host: 8080/folder)

## Linux:

1) Get [WinSCP](https://winscp.net/eng/download.php) for transferring files to your instance

2) With WinSCP, transfer the war file (assuming the file name is wisp. War) to the following location:

/home/ec2-user/wisp. War

3) Using Putty, enter the following command

sudo -s (for root access)

cp /home/ec2-user/wisp.war /var/lib/tomcat7/webapps

4) Start / Restart your tomcat with the following command

sudo service tomcat7 start (to start server)

sudo service tomcat7 stop (to stop server)

sudo service tomcat7 restart (to restart, if your tomcat has already started)

5) Verify that it has been uploaded at the following location

<http://instanceURL:8080/wisp>

4. Change Social App ID:

To change Facebook and google application ids for social authentication or google analytics, find socialauth.js file under,

Wisp-web --> src --> main --> webapp --> resources --> js --> socialauth.js

**Note: To effect the changes in socialauth.js or application properties or data source please rebuild the application and redeploy in server.**