**How to deploy Maven based war file to Tomcat**

**Using maven-tomcat-plugin**

**1. Tomcat Authentication**

First, add an user with administrator access right for Tomcat. To add Tomcat user, edit this file – “%TOMCAT\_PATH%/conf/tomcat-users.xml“.

*File : tomcat-users.xml*

<?xml version='1.0' encoding='utf-8'?>

<tomcat-users>

<role rolename="manager"/>

<role rolename="admin"/>

<user username="admin" password="password" roles="admin,manager"/>

</tomcat-users>

**2. Maven Authentication**

In Maven side, you need to add the same user authentication information in “%MAVEN\_PATH%/conf/settings.xml“.

*File : settings.xml*

<server>

<id>TomcatServer</id>

<username>admin</username>

<password>password</password>

</server>

**3. Maven-Tomcat-Plugin**

Declare “Maven-Tomcat plugin” and related Tomcat server detail in your pom.xml file.

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>tomcat-maven-plugin</artifactId>

<configuration>

<url>http://<your ip>/manager</url>

<server><some name></server>

<path>/<some name for path></path>

</configuration>

</plugin>

**4. Deploy to Tomcat**

Run “mvn tomcat:deploy” to package your project in a WAR file, and deploy it to Tomcat server. To verify it, just access to the Tomcat’s manager page and make sure “**/<some name for path>**” path is existed.

*URL : http://<your ip>:8080/manager/*

**Using maven-tomcat-plugin**

Add the following entry into the pom.xml. This is jus a WAR file packing plugin so it can be moved to any location on install phase

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<configuration>

<warName>mavenproject1</warName> (Optional, used when u want a constant name to your project)

<outputDirectory>E:\apache-tomcat-7.0.5\webapps</outputDirectory>

</configuration>

</plugin>