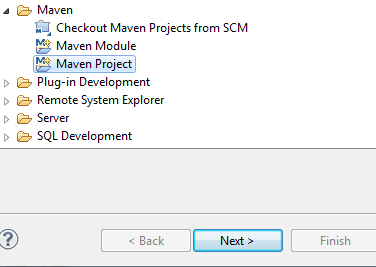
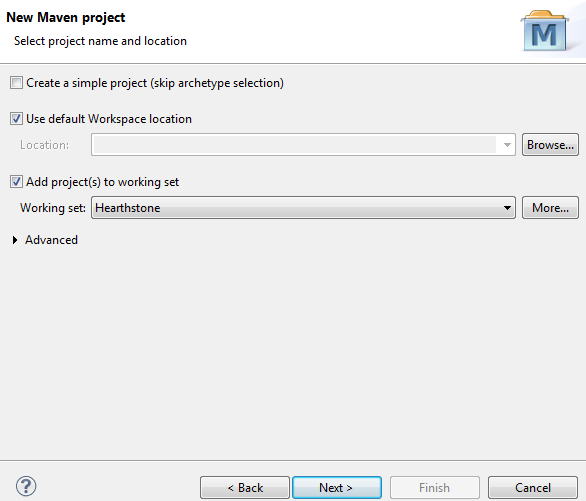
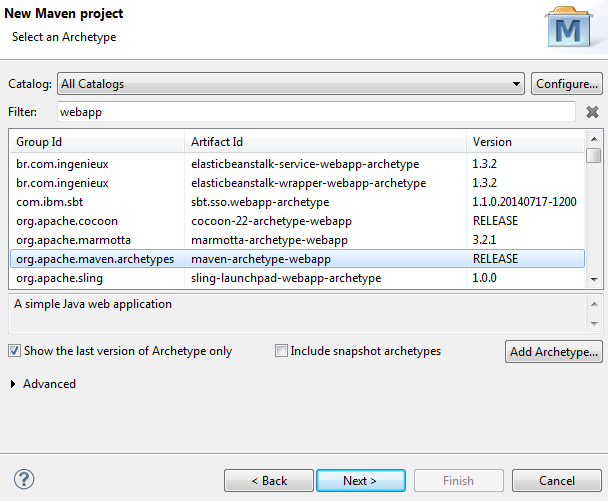
**CREATE A MAVEN PROJECT AND RUN IT IN ECLIPSE**

****

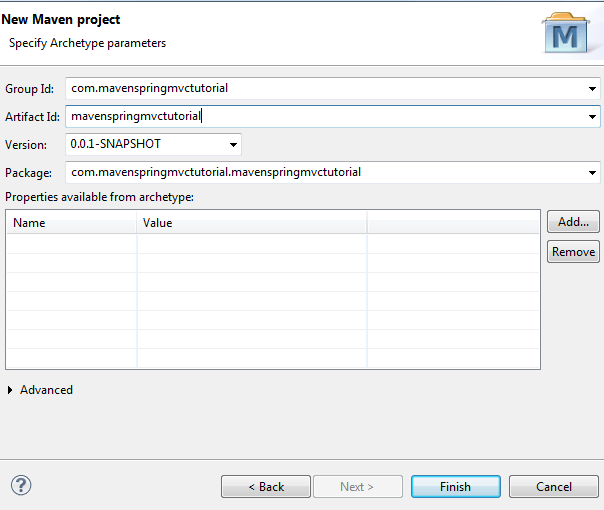
Create a new maven project and click next



Choose default setting and click next

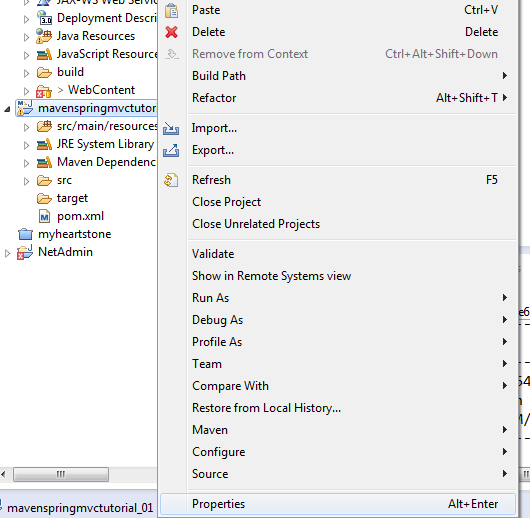
****

Choose webapp archetype and click next

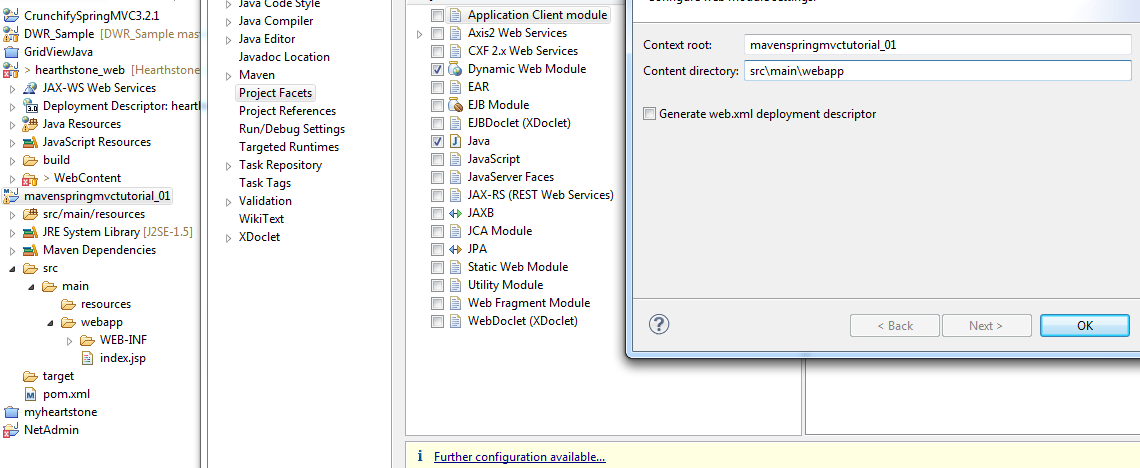
****

Add Group ID and Artifact ID and click finish

Wait for maven to complete the project for the chosen archetype



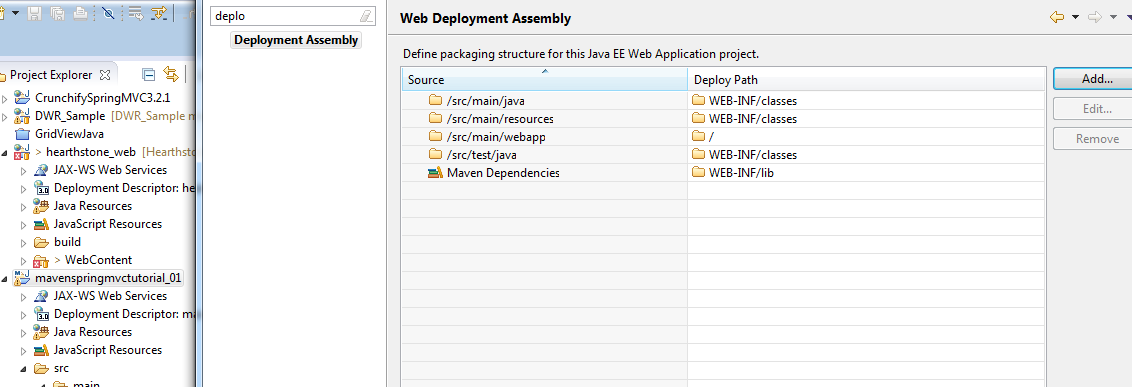
Right click properties and click project facets



Click the **Dynamic Web Module** and replace the content directory: **src\main\webapp**

And you can now run the maven web project to your tomcat server

To access the maven dependencies you must right click the properties again and search for the deployment assembly, add the Maven Dependencies and click OK



For the POM.xml

Add the following dependencies

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>mavenspringmvctutorial</groupId>

<artifactId>mavenspringmvctutorial</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>mavenspringmvctutorial Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<!-- Spring 4 dependencies -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>4.0.6.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>4.0.6.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>4.0.6.RELEASE</version>

</dependency>

</dependencies>

<build>

<finalName>mavenspringmvctutorial</finalName>

</build>

</project>

to include the spring MVC libraries

Troubleshooting:

http://stackoverflow.com/questions/7715260/java-compiler-level-does-not-match-the-version-of-the-installed-java-project-fac

If you are using m2e plugins for eclipse you need to specify the target version as 1.6

e.g.: Java compiler level does not match the version of the installed Java project facet.

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<source>1.6</source>

<target>1.6</target>

</configuration>

</plugin>

</plugins>

</build>

Alternatively, you can specify the maven.compiler.source and maven.compiler.targetproperties with values of 1.6, that happen to be the equivalent:

<properties>

<maven.compiler.target>1.6</maven.compiler.target>

<maven.compiler.source>1.6</maven.compiler.source>

</properties>