**DOJO MAVEN INTEGRATION: -**

Before MAVEN: -

We have already developed a web project in the traditional manner, like using servlets, JSP, without maven.

1. Here we have included all the external jar files to the /web-inf/lib folder manually.
2. Jar files are downloaded manually.
3. Dependent jars also we need to include the manually based on it needs.
4. Generating WAR file is manually.

**To Overcome above MAVEN Is introduced into my project: -**

**Steps to follow to Integrate Maven for Existing project: -**

1. Go to eclipse, right click on Your already existing project and go to CONFIGURE 🡪 Convert to maven project.
2. It will open and create POM.xml file and target folder for your project.

**Some common debugging things you need to check to ensure MAVEN IS WORKING:**

* 1. When you say Clean project from eclipse or UPDATE PROJECT from MAVEN option, you should be able to see in settings.xml file under .m2 Folder which created in your local machine.
  2. If not please include it from externally.
  3. And it should contain the plugins that it uses and also the dependencies files that you have in POM.xml file.
  4. If it is not able to download any of your plugins like, error like maven-resources-\*\*\*.jar file not able to download or not able to find.
  5. Then it may be because of certicates i.e PROXY file. i.e it is not able to connect maven artifactory location on web. That means your internet is restricting to access those sites.
  6. To solve it you need to add proxy certificates related to your organization to JAVA INSTALLATION/JRE/LIB/Security/Cacerts file. Then you may be able to solve it.

1. But to really to run our project as maven project, you need to follow the maven architecture i.e folder structure we need to follow.
   1. Like JAVA SOURCES will be put inside com/main/java + package & java files of your project.
   2. Test Cases should be put inside the com/main/test + package & java files of your project.
   3. And All the web-Content code should be placed inside the com/main/webapp folder. (inside that only web-inf and web.xml file exists).
2. Once you placed all the files properly, Add dependencies(library jar files that you need to have ). Delete the web-inf/lib directory where you have included those manually previously.
3. Once you added the dependencies, then you right click on the project, go to maven and say UPDATE PROJECT. ( it will try to look your pom.xml file and see all dependencies are downloaded into your local .m2/repository/ folder or not. If not it will automatically download that particular jar and dependent jars for that.
4. Once it is done, then you right click on the project 🡪 Run as 🡪 mvn clean.
5. Project 🡪 run as 🡪 mvn install or any other commands based on your need.
6. Once build is success, then it will create a war file for you on TARGET folder.
7. You can look into it.
8. Then you can publish and start/restart the TOMCAT server that you already have. Then access application and see whether it is running or not. ( max cases IT WILL WORK).

**Some common Debugging steps when you are doing setup. : -**

1. Check whether your Maven Dependencis added to your DEPLOYMENT ASSESMBLY OR NOT. ( right click on project 🡪 properties 🡪 deployment assembly ).
2. If not click on add, then choose “Java Build Path Entries”, then choose maven dependencies. (Maven dependencies only available if it created .m2/repository/all jars.)
3. And In the DEPLOYMENT ASSEMBLY, you also check the SOURCE & DEPLOY PATH.



if src/main/test is available it should also present there in the list.

And JAVA BUILD PATH SHOULD contain following minimum.

