**Eclipse Maven & Tomcat Configuration**

Using this document we can able to configure maven and tomcat in eclipse IDE. To do that we need to download latest version of eclipse from the eclipse corporate site.

For 32 bit windows version.

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/neon/R/eclipse-jee-neon-R-win32.zip>

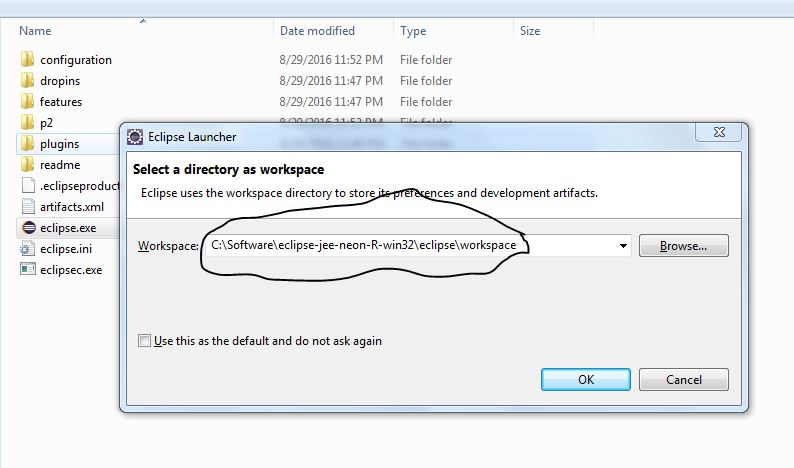
For 64 bit windows version.

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/neon/R/eclipse-jee-neon-R-win32-x86_64.zip>

Step : 1

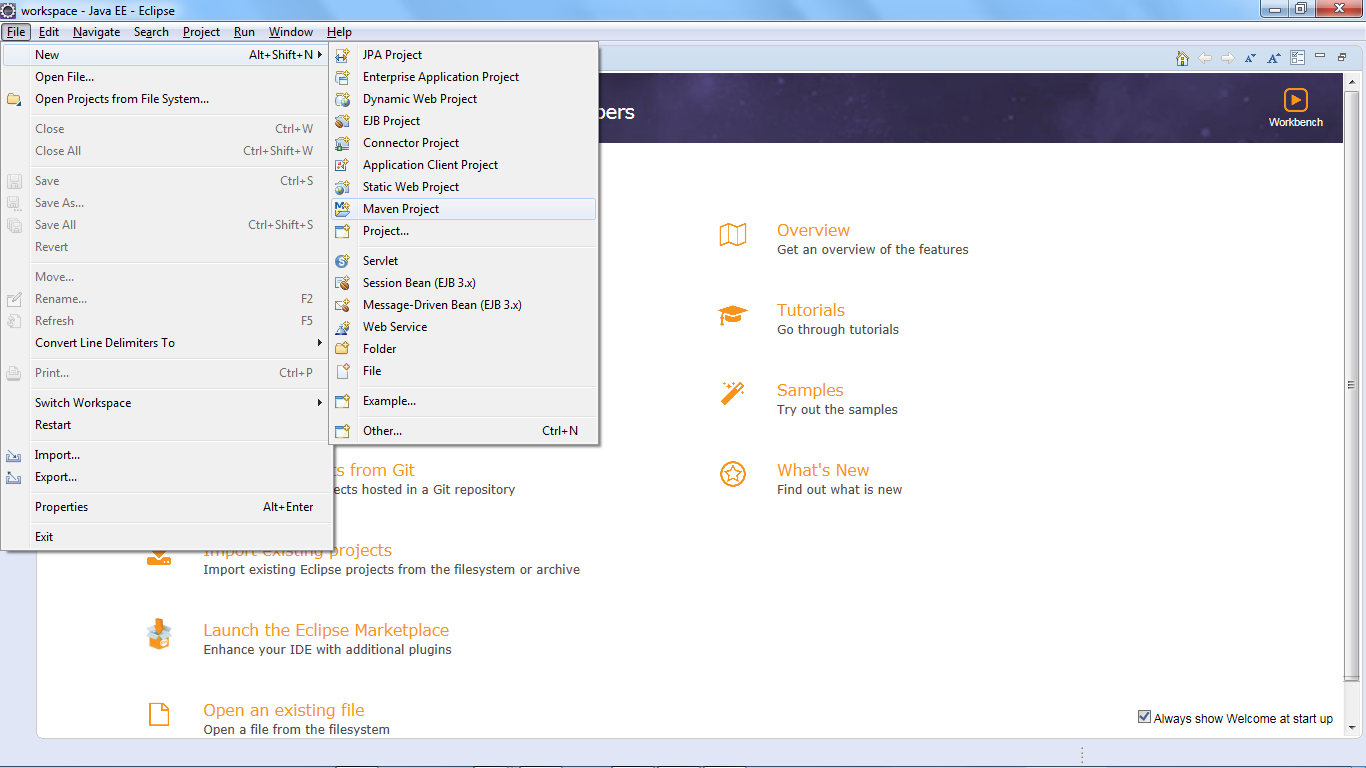
Once is downloaded, extract the eclipse zip file into specific folder. And under eclipse folder create one workspace empty directory.

eclipse-jee-neon-R-win32 🡪 eclipse 🡪 **<<<workspace>>>**

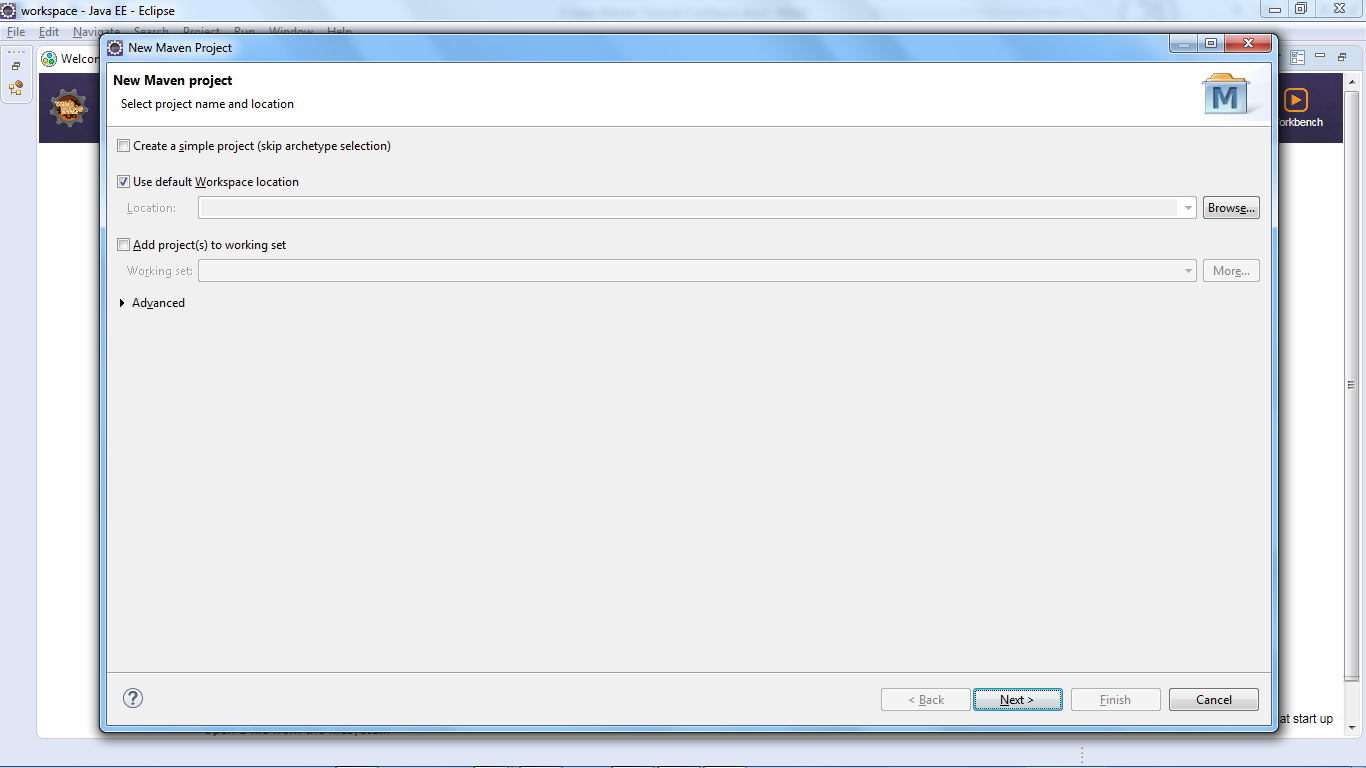
Then start the eclipse by clicking ecipse.exe file and select the workspace where u created as show in below.

Step : 2

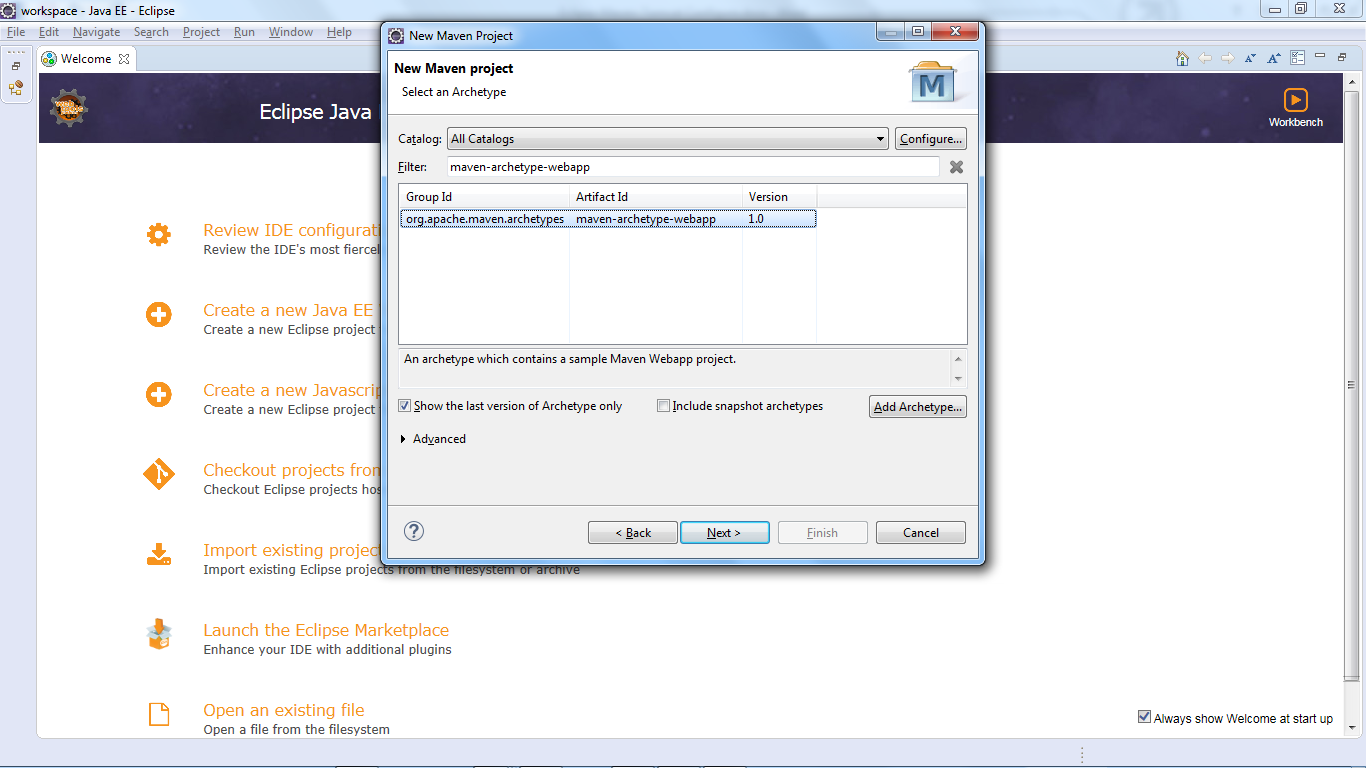
Create new maven project by selecting File 🡪 New 🡪 Maven Project.



In **New Maven Project** window, click **next** button without modifying anything.



In second window we need to select Archetype by typing in Filter as : **maven-archetype-webapp**



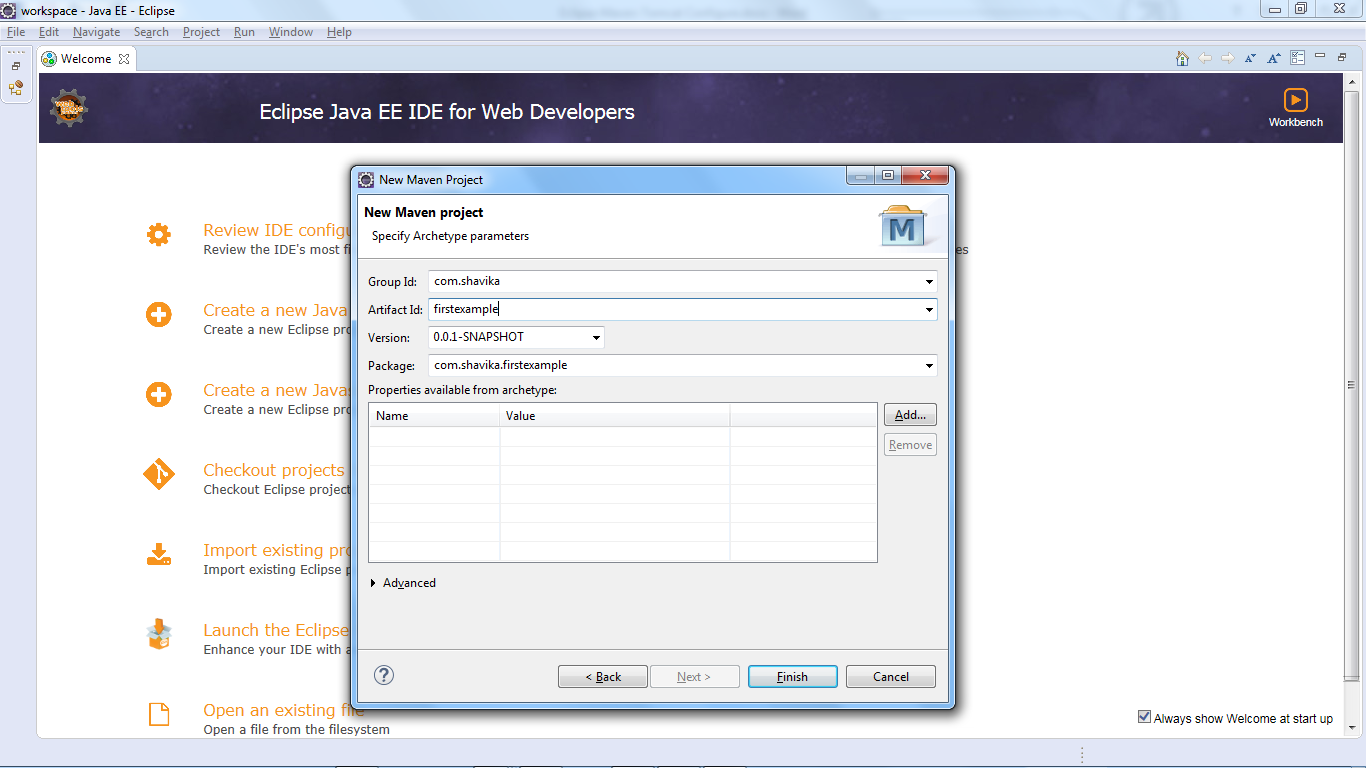
And click next button.

In next window, we need to specify Archetype parameters. As show in below screenshot. And in that some important parameters should know before set it.

It should be noted that there should be a single POM file for each project.

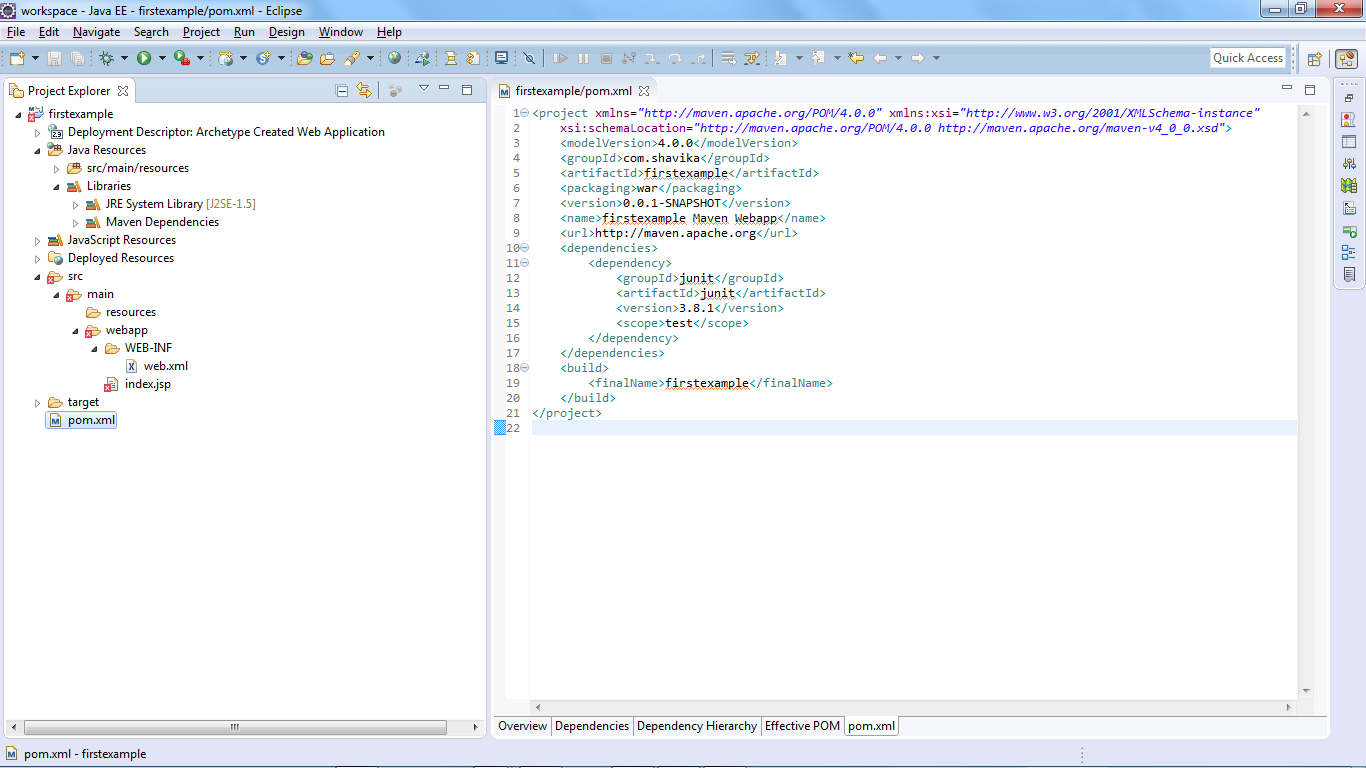
* All POM files require the **project** element and three mandatory fields:**groupId, artifactId,version.**
* Projects notation in repository is **groupId:artifactId:version.**
* Root element of POM.xml is **project** and it has three major sub-nodes :

|  |  |
| --- | --- |
| **Node** | **Description** |
| groupId | This is an Id of project's group. This is generally unique amongst an organization or a project. For example, a banking group com.company.bank has all bank related projects. |
| artifactId | This is an Id of the project.This is generally name of the project. For example, consumer-banking. Along with the groupId, the artifactId defines the artifact's location within the repository. |
| version | This is the version of the project.Along with the groupId, It is used within an artifact's repository to separate versions from each other. For example:   *com.company.bank:consumer-banking:1.0*  *com.company.bank:consumer-banking:1.1.* |



Once you click **Finish,** required configuration and dependent file (or) jar will download in to **.m2** directory**. (c:/Users/<<userName>>/.m2)**

And file structurewill display in **eclipse 🡪 Project Explorer.**



**Note: -** index.jsp file will show error, due to servlet.jar is not configured. Once is done this error will go.

Now you can open the pom.xml file and modify as specified below.

In pom.xml file there are 3 groups are there.

1. Model :- In this block we can see project related information like ‘groupId’, ‘archetypeId’, ‘package’, ‘version’… etc.
2. Dependencies :- In this block we can add projected dependent jar artifact.

Ex : <dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

Like this we can add lot of dependence library which is useful for project.

1. Build :- In this block we can specify the plugins.

There are 3 major useful plugins we will use for Web Application applications those are ‘Build Project’, ‘Eclipse Configure’ & ‘Tomcat plugin’.

As below specified.

<build>

<finalName> PROJECT\_NAME</finalName>

{{{{ Add the below plugins here}}}}

</build>

**<plugins>**

**<!-- Build project -->**

**<plugin>**

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

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

**<version>3.1</version>**

**<configuration>**

**<source>1.8</source>**

**<target>1.8</target>**

**</configuration>**

**</plugin>**

**<!-- Eclipse project -->**

**<plugin>**

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

**<artifactId>maven-eclipse-plugin</artifactId>**

**<version>2.9</version>**

**<configuration>**

**<!-- Always download and attach dependencies source code -->**

**<downloadSources>true</downloadSources>**

**<downloadJavadocs>false</downloadJavadocs>**

**<!-- Avoid type mvn eclipse:eclipse -Dwtpversion=2.0 -->**

**<wtpversion>2.0</wtpversion>**

**</configuration>**

**</plugin>**

**<!-- Tomcat plugin -->**

**<plugin>**

**<groupId>org.apache.tomcat.maven</groupId>**

**<artifactId>tomcat7-maven-plugin</artifactId>**

**<version>2.2</version>**

**<configuration>**

**<port>9090</port>**

**<path>/</path>**

**</configuration>**

**</plugin>**

**</plugins>**

Finally Project structure will appear as show below.

***FirstApplication***

|-- pom.xml

`-- src

`-- main

`-- resources

|-- META-INF

| `-- maven

| `--archetype.xml

`-- archetype-resources

|-- pom.xml

`-- src

|-- main

| `-- java

| `-- App.java

`-- test

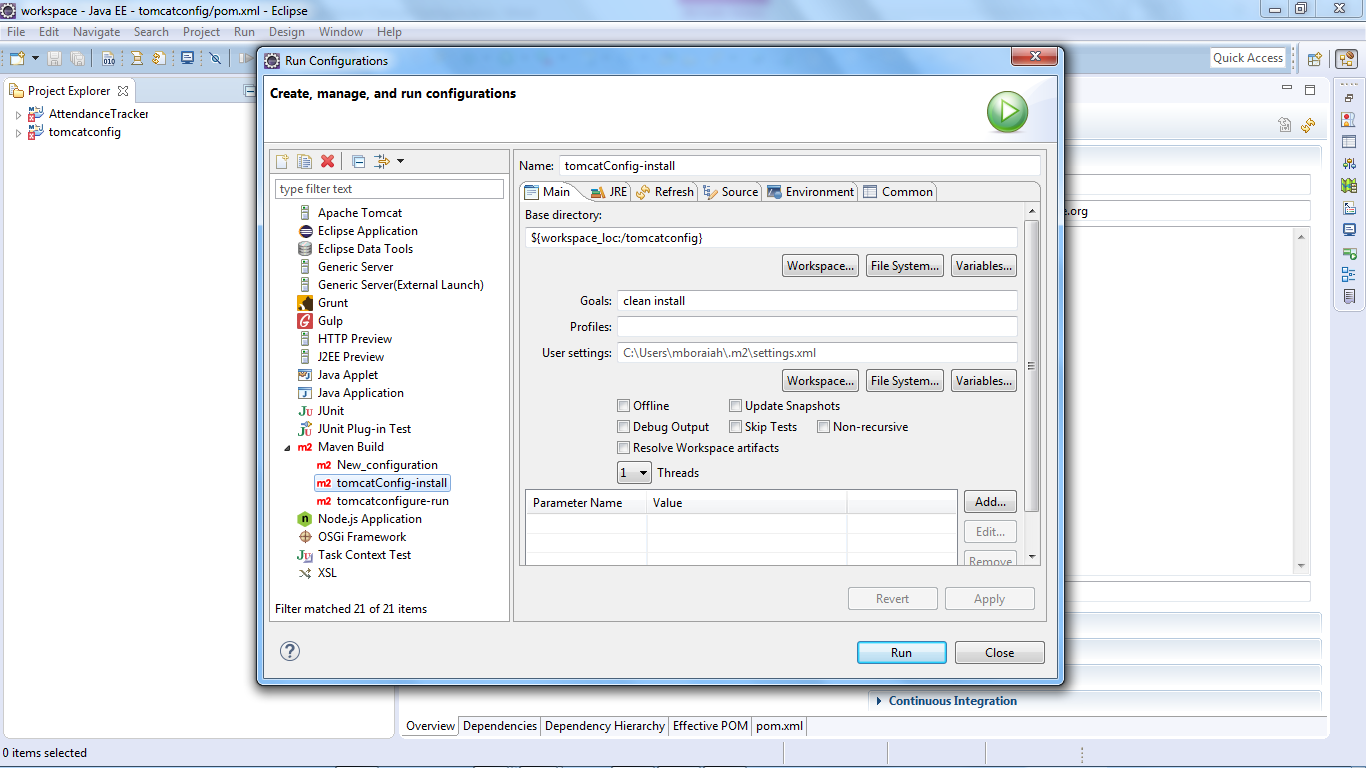
`-- java

`-- AppTest.java

Step : 3

Now Application configuration is ready to compile & Run the project.

====🡺 Clean & Build the project

Configure the ‘**Run Configuration’** by selecting ***Run menu*** 🡪 ***Run Configuration.*** Then you are able to see the Properties window as show below.

In that select maven Build 🡪 right click and ***New***,

then provide

Name : as run configuration

Base directory : click WorkSpace [above of Goals text box] and select project name.

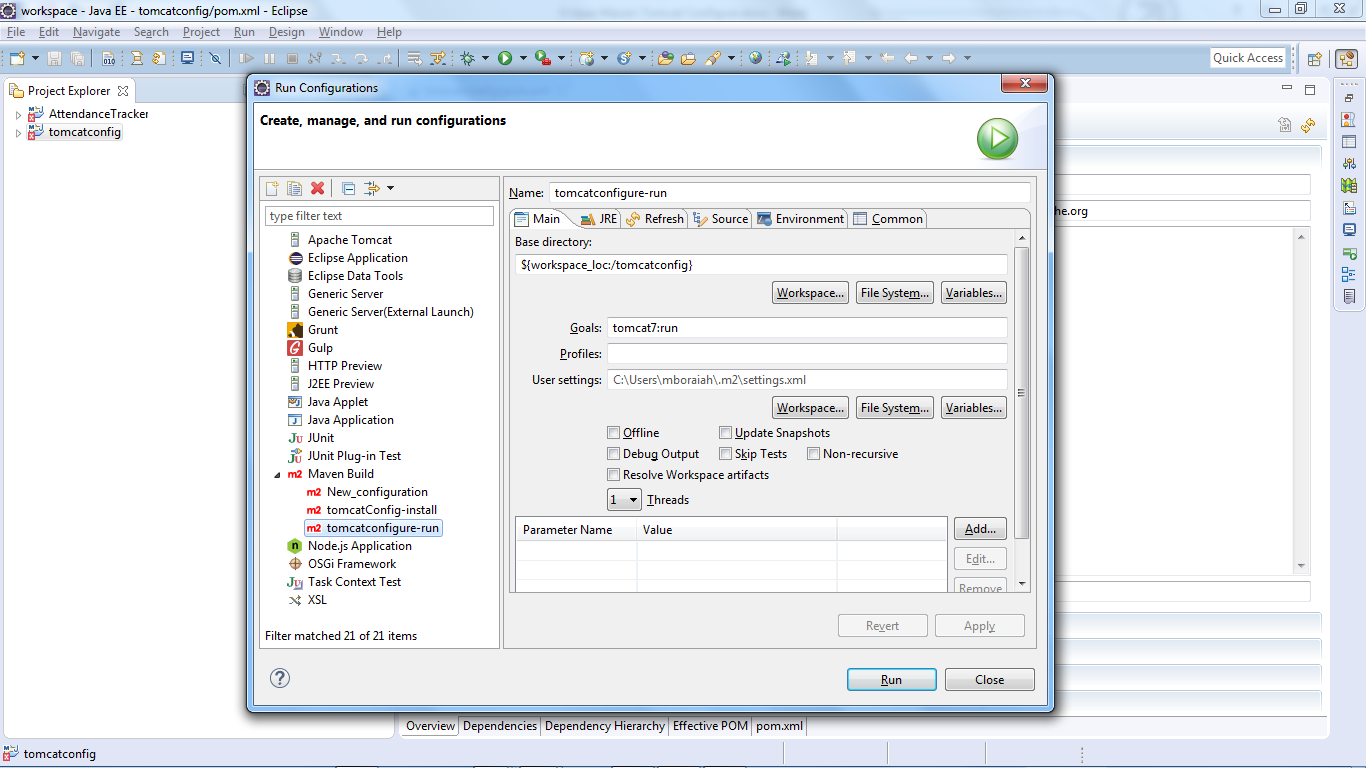
Goals : need to add ***clean install***

User settings :  above file save into c:/Users/<<user\_name>>/.m2/

Then click apply and then run.

====🡺 Run the project using tomcat

Select ***Run menu*** 🡪 ***Run Configuration.*** Then you are able to see the Properties window as show below.



In that select maven Build 🡪 right click and ***New***,

then provide

Name : as run configuration

Base directory : click WorkSpace [above of Goals text box] and select project name.

Goals : need to add ***tomcat7:run***

User settings :  above file save into c:/Users/<<user\_name>>/.m2/

Then click **apply** button and then **run** button.

THE END