

Maven java and web project and pushing into GITHUB

Creation of Maven Java Project Procedure Steps:

Step1: Open Eclipse IDE for Enterprise JAVA and WEB developers.

Step2: Click on File->New->Maven Project

Step3: Type quickstart in Filter and wait for artifacts to load.

Step4: Select org.apache.maven.archetypes maven-archetype-quickstart 1.4 and click on next

Step5: Mention TeamID: as SE, ArtifactID: MavenJava and click on next

Step6: Now in Console we can see all the artifacts are grouped and console is waiting as Y: Now click Y and finally SE.MavenJava gets ready by combining all the artifacts.

Step7: We can see our MavenJava project ready with its own pom.xml file in the Project Explorer tool bar menu present at the left side.

Step8: Now click on src/main/java->SE.MavenJava->App.java where we can find a code sample for HELLO WORLD.

Step9: Now right click on App.java->Run as->Maven Clean

Step10: Now right click on App.java->Run as->Maven Install

Step11: Now right click on App.java->Run as->Maven Test

Step12: Now right click on App.java->Run as->Maven Build

Step13: In Goals, Type “clean install test”, click on apply next click on Run, for building the MavenJava project.

Step14: Every phase is checked and BUILD SUCCESS message can be seen in the console.

Step15: Now right click on App.java->Run as->Java Application, click on OK.

Step16: Hello world is shown as output for MavenJava Project.

Creation of Maven Web Project Procedure Steps:

Step1: Open Eclipse IDE for Enterprise JAVA and WEB developers.

Step2: Click on File->New->Maven Project

Step3: Type webapp in Filter and wait for artifacts to load.

Step4: Select org.apache.maven.archetypes maven-archetype-webapp 1.4 and click on next

Step5: Mention TeamID: as SE, ArtifactID: MavenWeb and click on next

Step6: Now in Console we can see all the artifacts are grouped and console is waiting as Y: Now click Y and finally SE.MavenWeb gets ready by combining all the artifacts.

Step7: We can see our MavenWeb project ready with its own pom.xml file in the Project Explorer tool bar menu present at the left side.

Step8: Now click on MavenWeb->src->main->Webapp->index.jsp where we can find a code sample for HELLO WORLD.

Step9: Now open browser and open mvnrepository.com

Step10: Search for JAVA servlet API and click on the latest version.

Step11: Copy the dependency code of the servlet. And paste it in MavenWeb's pom.xml under target folder.

Step12: Now click on window in the menu bar, click on showview->servers.

Step13: Add servers by clicking on "No servers available".

Step14: Select Tomcat v9.0 server/ Tomcat v11.0 server in the server type Click on next, select Apache Tomcat v9.0(optional), Click Apply and close.

Step15: Now we can find tomcat server attached to the project in server menu of the console. Double click on it where the tomcat configuration page appears.

Step16: select "Use Tomcat Installation"(which is 2nd option) in Server locations, and modify the port numbers to "0" for admin port and "8085" for Https/1.1 and close the tab.

Step17: Double click "servers" folder present at the left side in project explorer, click on Tomcat v9.0 server at localhost-config->tomcat-users.xml, scroll to the end and paste the following exactly above</tomcat-users>tag and close the tab.

```
role rolename="admin-gui,manager-gui ,manager-script,manager-jmx,manager-status"/><user  
password="1234" roles="manager-gui, admin-gui ,manager-script" username="admin"/>
```

Step18: Now right click on index.jsp ->Run as->Maven Clean

Step19: Now right click on index.jsp ->Run as->Maven Install

Step20: Now right click on index.jsp ->Run as->Maven Test

Step21: Now right click on index.jsp ->Run as->Maven Build

Step22: In Goals, Type "clean install test", click on apply next click on Run, for building the MavenWeb project.

Step23: Every phase is checked and BUILD SUCCESS message can be seen in the console.

Step24: Now right click on index.jsp ->Run as->Run on server, click on OK.

Step25: Select choose an existing server and click on finish.

Step26: Hello world Webpage is shown as output for MavenWeb Project.

Steps to Push the Maven Projects to GIT:

Java Project Push Procedure Steps:

Step1: Create two repositories for MavenJava and MavenWeb projects in your GitHub account.

Step2: In eclipse right click on MavenJava folder present in project explorer and select show in local terminal option->GitBash.

Step3: execute: git init->git branch –M main.

Step4: Now add your repository to the MavenJava using, git remote add origin<git java repository url>command.

Step5: execute: git add . ->git commit –m “Maven java push” ->git push –u origin main

Step6: Now refresh your GitHub account and we can see our MavenJava files are pushed to MavenJava repository in your GitHub account.

Web Project Push Procedure Steps:

Step1: In eclipse right click on MavenWeb folder present in project explorer and select show in local terminal option->GitBash.

Step2: execute: git init->git branch –M main.

Step3: Now add your repository to the MavenWeb using, git remote add origin<git Web repository url>command.

Step4: Push your web project same as Java to GitHub.

Step5: Now refresh your GitHub account and we can see our MavenWeb files are pushed to MavenWeb repository in your GitHub account.

Conclusion: we have learnt creation of Java and Web project through Maven and pushing them to GitHub.