# Maven Assignment

Assuming, you have GIT, tomcat, maven setup in your development environment; do the following:

1. At command prompt, what will be command to generate archetypes? Run this command so that you can generate the list of archetypes.
2. At command prompt, what will be command to create a new maven project with archetype whose description is following:
   1. ”A basic starter template using springboot, jpa data, thymeleaf and MVC”
   2. Run this command with the number for correct archetype so that new default project is created with springboot, jpa, thymeleaf and MVC.
3. Once the project is created with the required template:
   1. What will be the command to compile the project?
   2. What will be the command to package the built project?
   3. What will be command to execute the jar which is created by above command?
      1. After executing the jar, you should be able to test the REST API in any browser window.
   4. What will be command to clean all the compiled class files and jar files in target folder?
   5. Change POM.xml to create war executable in place of jar executable in the above project. Run the command to create .war executable.
4. Install tomcat in local development environment.
   1. Create manager user in tomcat by creating the config entry in tomcat-users.xml
   2. Deploy the war created in step 3.e in Apache Tomcat and access the REST API endpoint exposed by the deployed war.
      1. Hints:
         1. You may need to add following dependency in the project

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

<scope>provided</scope>

</dependency>

* + - 1. And, you may need to add following code to Spring Boot Main Class.

@SpringBootApplication

public class Application extends SpringBootServletInitializer{

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

}

* + - 1. SpringBootServletInitializer is imported from

org.springframework.boot.web.servlet.support.SpringBootServletInitializer

1. Jetty is the light weight servlet container. Add jetty dependency to the POM.xml
   1. You can get the Jetty dependency from <https://mvnrepository.com/artifact/org.eclipse.jetty/jetty-maven-plugin>
   2. Execute the project using jetty

# GIT Assignment

* + - 1. Create a new maven project with default archetype.
      2. Compile the project and generate jar/war file for the project using maven at command prompt.
      3. Initialize the newly created project as a GIT repository
      4. Commit the repository to the internal GIT server. Ask your mentor for internal GIT URL to be used in assignment.
      5. Currently all files are committed to remote repository. Configure the local repository such that target folder is not committed from local repository to remote repository.
      6. Delete the “target” folder from the remote repository which was committed in step 4.
      7. Create a new branch “welcomeapi” in the local repository.
         1. Make changes to the source code i.e. add a new url (/welcome) to the rest controller
         2. Now, push the change from local repository to remote repository in the branch “welcomeapi”
         3. You must not push these changes to main branch in remote repository.
         4. Locally, merge the changes done welcomeapi branch to main branch.
         5. Push the changes from local repository to remote repository.
         6. Once the changes are merged and pushed to the remote copy of repository on GIT server, delete the branch welcomeapi.

# Junit Assignment

Consider the following Calculation class created by developers:

public class Calculation {

public static int add(int a, int b) {

return a + b;

}

public static int sub(int a, int b) {

return a - b;

}

Write a Junit Test case to test the above class for the business logic