MAVEN FUNDAMENTALS

LAB ASSIGNMENTS

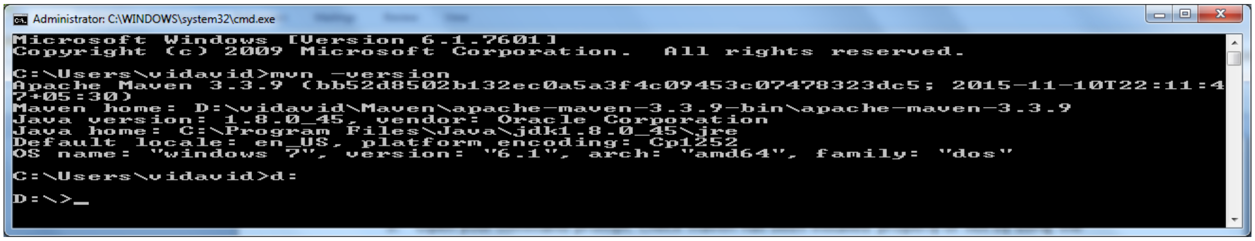
Lab 1:

Install maven in your machine and try to create one simple maven project through **command prompt.** Check the complete project layout which should contain src/main/java and src/test/java. And also check where maven includes the pom.xml file or not.

Steps:

1. Open the URL “ http://maven.apache.org/download.cgi”
2. Download latest maven version, you can get one zip folder.
3. Unzip it in your local drive(such as D:)
4. Set your M2_HOME and PATH variable as
    a. M2_HOME = D:\vidavid\Maven\apache-maven-3.3.9-bin\apache-maven-3.3.9
    b. PATH= D:\vidavid\Maven\apache-maven-3.3.9-bin\apache-maven-3.3.9\bin
5. Open your command prompt. Check maven has been installed properly or not by using the following command.
    a. mvn –version

   (if maven has been installed properly you will get the command like the below)



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\vidavid>mvn -version
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T22:11:4
7+05:30)
Maven home: D:\vidavid\Maven\apache-maven-3.3.9-bin\apache-maven-3.3.9
Java version: 1.8.0_45, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.8.0_45\jre
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 7", version: "6.1", arch: "amd64", family: "dos"

C:\Users\vidavid>d:

D:\>_
```

6. Change d: drive and create one folder my-first-maven-app
7. Type the below command:
    a. mvn archetype:generate

   (Now maven plug-ins will be downloaded)

8. Once maven plug-ins are downloaded, under
    a. Choose a number or apply filter (format: [groupId:] artifactId, case senditive contains) : 510 :
    b. Here enter 510 or 810 (it depends on your maven version what you use)
       (Note: This is to create simple non web java project)
9. Do the followings to complete the project creation:
    Choose a number 6: 6
   Define value for property 'groupId': : **com.capgemini**
   Define value for property 'artifactId': :**maven-demo**
   Define value for property 'verion' : 1.0-SNAPSHOT:: **1.0-SNAPSHOT**

Define value for property 'package' : : **com.capgemini.demos**

y:**y**

10. Now open your folder, maven must be creating a completed project structure for you.

Conclusion:

From the above example, we learnt

- How to install maven
- How to create simple maven project which has complete project layout.

Lab 2:

Modify maven configuration file pom.xml, such as specify default compiler for your java project as jdk 1.8 and also include the below jars in **eclipse IDE.**

Jars:

- junit-4.12.jar
- mockito-all-1.10.19.jar.

Steps:

11. Open GitHub URL (https://github.com/Training2017/MCI)
12. Click **ProductApp.zip folder,** next page click download link. The zip file will be downloaded.
13. Unzip the folder
14. Go to Eclipse IDE, choose import ->General -> Existing Projects into Workspace
15. Under select root directory click browse button.
16. Go to the unzip folder location and choose the folder in the name of ProductApp.
17. Now the Product Application project will be imported in your local machine.
18. Open pom.xml file under ProductApp.
19. Add the configurations in the highlighted place.

```
pom.xml
<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/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.capgemini.product</groupId>
  <artifactId>ProductApp</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
      <!-- include the dependency here -->

  </dependencies>

  <build>
     <plugins>
             <!-- Add your plugin to specify default jdk as 1.8 -->

       </plugins>
       </build>
</project>
```

Conclusion:

From the above example, we learnt to use pom.xml file for our project configurations such as including **dependencies** and **build plug-in** configurations.