

Packaging Maven Project as Executable Jar

Objective:

Convert the maven project as an executable jar file. It is used to run all the tests by executing jar file.

Run TestNg Programmatically:

To run TestNg tests programmatically instead of testing.xml, create a class with main method and add the below code. We can execute this standalone java class file which will execute all the added TestNg classes.

```
public class ExecuteTestng {  
  
    public static void main(String[] args) {  
  
        XmlSuite suite = new XmlSuite();  
        // Assign suite name  
        suite.setName("Suite");  
  
        XmlTest test = new XmlTest(suite);  
        // Assign Test name  
        test.setName("Test");  
        List<XmlClass> classes = new ArrayList<XmlClass>();  
        // add all the testing classes  
        classes.add(new XmlClass("learning.CreateLead"));  
        test.setXmlClasses(classes);  
  
        List<XmlSuite> suites = new ArrayList<XmlSuite>();  
        // add suites to be executed  
        suites.add(suite);  
        TestNG tng = new TestNG();  
        tng.setXmlSuites(suites);  
        tng.run();  
  
    }  
  
}
```

Add Plugin to pom.xml:

To convert the maven project as an executable jar file, below plugin needs to be added to the pom.xml below </dependencies>.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-assembly-plugin</artifactId>
      <version>3.1.1</version>
      <configuration>
        <descriptorRefs>
          <descriptorRef>jar-with-
dependencies</descriptorRef>
        </descriptorRefs>
        <archive>
          <manifest>

            <mainClass>learning.ExecuteTestng</mainClass>
          </manifest>
        </archive>
      </configuration>
      <executions>
        <execution>
          <phase>package</phase>
          <goals>
            <goal>single</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

Update correct class file:

Correct class file which is having the TestNg execution code should be updated in the <mainClass> as below.

```
<manifest>

    <mainClass>learning.ExecuteTestng</mainClass>

</manifest>
```

Create a Variable to Set the Driver Path in POM.XML

If you skip this step then you need to execute the jar file from where the chromedriver.exe is located at, in order to overcome this, which is to run the executable file from wherever regardless of the chromedriver's location, you need to undergo this step.

```
<properties>
<driverpath>${driverpath}</driverpath>
</properties>
```

Properties file

Create a Properties File:

You've declared the property variable in pom.xml under <properties>, now you have to assign this variable value to a variable in the property file.

Create a file under src/main/resources with any name followed by the extension '.properties'.

Inside the file, assign '\${driverpath}' to a variable, example 'path'.

```
path=${driverpath}
```

Go to you pom.xml -> create | copy / paste this dependency under the <build>

```
<resources>
<resource>
<directory> src/main/resources</directory>
```

```
<filtering>true</filtering>
</resource>
</resources>
```

Pass the path value to Test Case

Finally, pass the path value / path variable to the test case with this code.

Go to your test case, paste this code above ChromeDriver initialization.

```
Properties props = new Properties();
props.load(CreateLead.class.getClassLoader().getResourceAsStream("project.p
roperties"));
String path = props.getProperty("path");
System.setProperty("webdriver.chrome.driver", path+"\\chromedriver76.exe);
```

Packaging Maven Project as Executable JAR

☐ Right click on your project -> Run As -> Maven build.. -> Enter Goals as 'package -Ddriverpath=*'your chrome driver's path'*

Ex: package -Ddriverpath=D:\softwares\chromedriver.exe

☐ Run it.

☐ Now go to your eclipse workspace in your system directory -> go to your project folder -> target folder -> run the jar file that ends with '-jar-with-dependencies.jar'.

☐ Launch your command prompt from this path.

☐ Now run this command '**java -jar filename.jar**'.