Maven

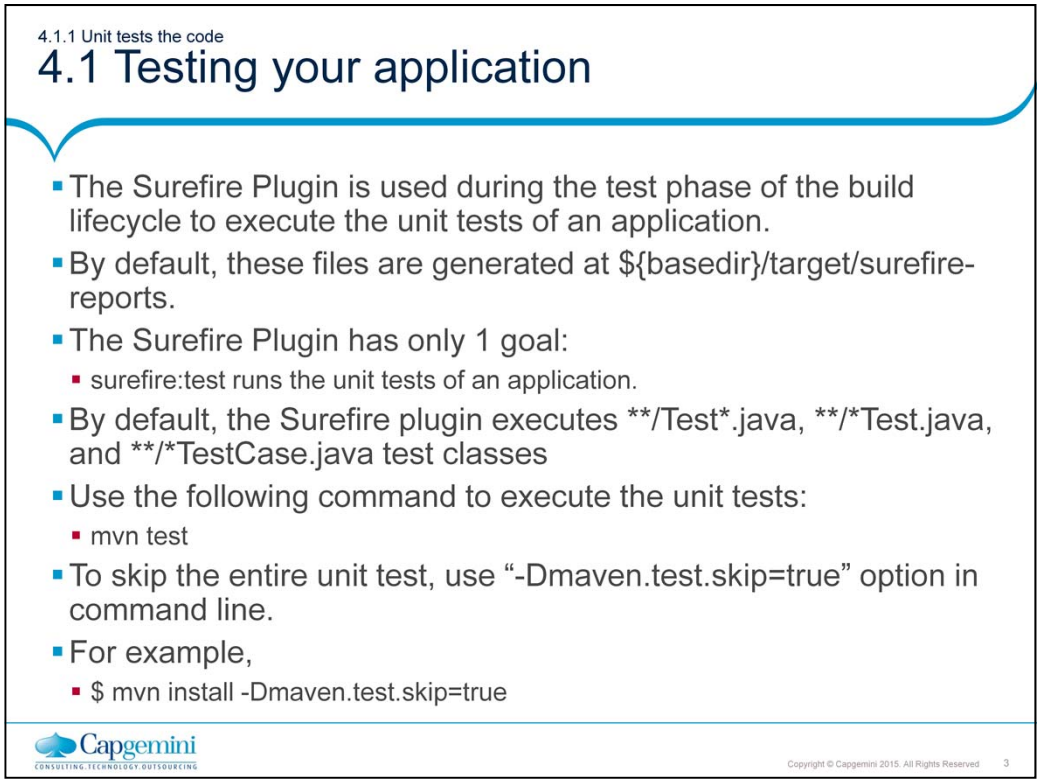
Creating Applications with Maven

Lesson Objectives

- In this lesson, you will learn about:
- Overview
- Testing your application
- Generating Java Projects using Maven in Eclipse Luna
  - Setting Maven in Eclipse Environment
  - Adding/Updating Dependencies
  - Importing Project
  - Creating Maven Module
- Creating Web application using Maven
  - Creating Maven web Module

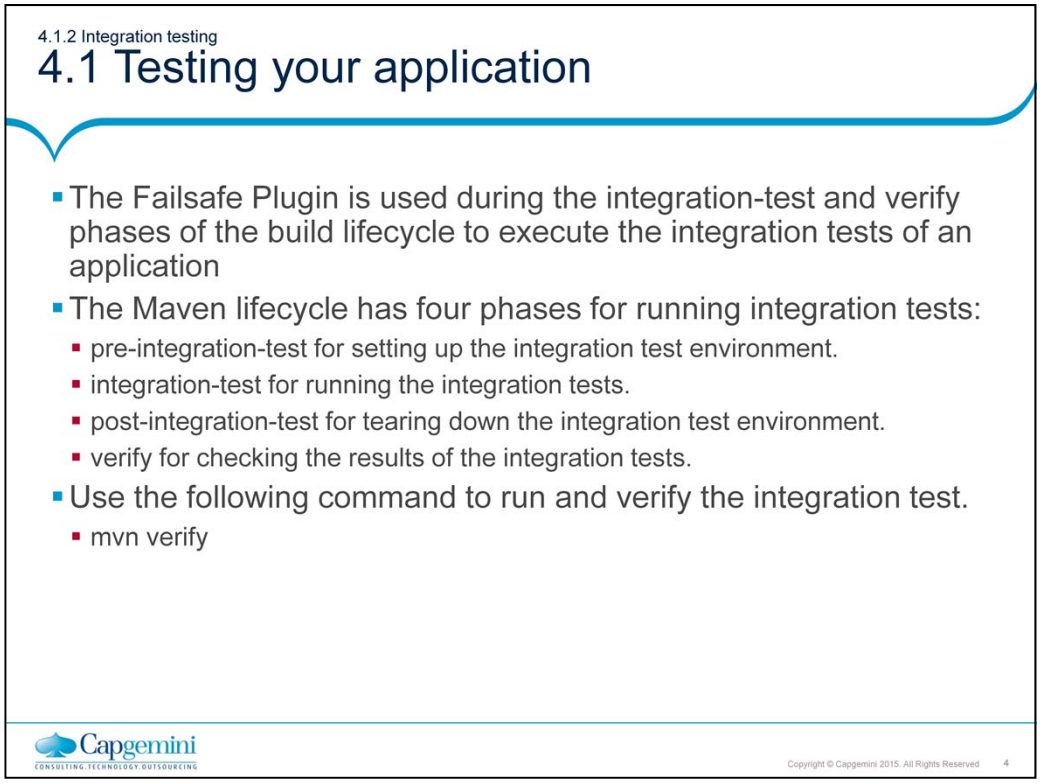
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2



It generates reports in 2 different file formats:

- Plain text files (*.txt)
- XML files (*.xml)



The Failsafe Plugin has only 2 goals:

failsafe:integration-test runs the integration tests of an application.

failsafe:verify verifies that the integration tests of an application passed.

The Failsafe plugin will look for **/IT*.java, **/*IT.java, and **/*ITCase.java.

4.1.3 Running unit test in Integration test phase

4.1 Testing your application

- Add these code snippet in pom.xml

```
.....
<executions>
        <execution>
                <id>unit-tests</id>
                <phase>test</phase>
                <goals>
                        <goal>test</goal>
                </goals>
                <configuration>
                <!-- Never skip running the tests when the test phase is invoked -->
                <skip>false</skip>
                        <includes>
                        <!-- Include unit tests within integration-test phase. -->

                        <include>**/*Tests.java</include>
                        </includes>
                        <excludes>
                        <!-- Exclude integration tests within (unit) test phase. -->

                        <exclude>**/*IntegrationTests.java</exclude>
                        </excludes>
                </configuration>
        </execution>
<executions>
......
```

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

 5

4.2.1 Starting Eclipse at JDK

4.2 Setting Maven in Eclipse Environment

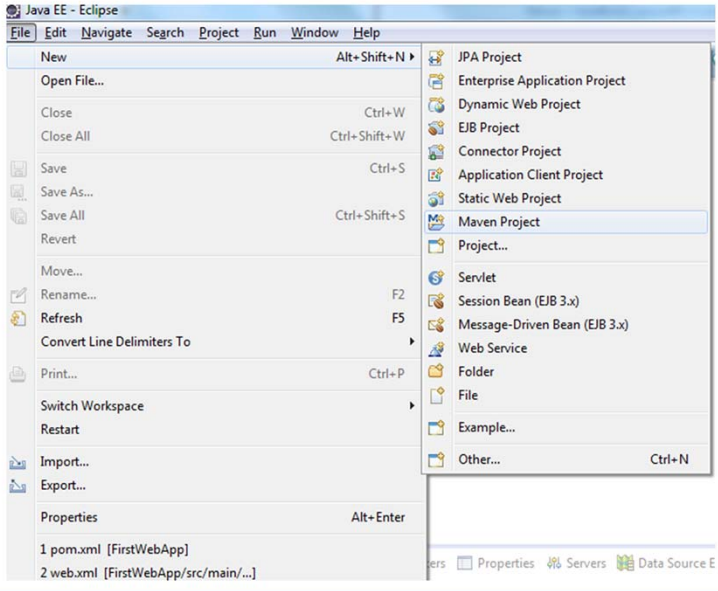
- Maven requires Eclipse using a JDK, instead of a JRE.
- To check with what Java version (JRE or JDK) Eclipse is running, do the following:
  - Open the menu item "Help > About Eclipse". (On the Mac, it's in the Eclipse-menu, not the Help-menu)
  - Click on "Installation Details".
  - Switch to the tab "Configuration"
  - Search for a line that starts with "-vm". The line following it shows which Java binary is used.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 6

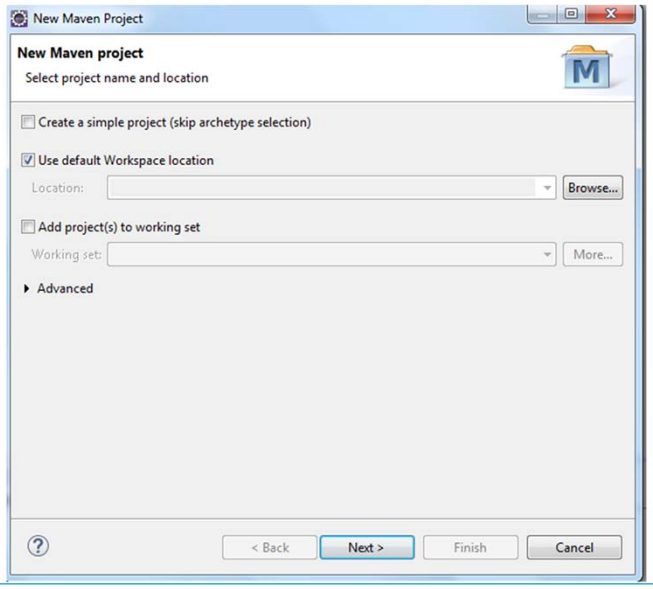
4.3.1 Maven Project in Eclipse

4.3 Creating Maven Project in Eclipse Luna



4.3.2 Maven Project in Eclipse

4.3 Creating Maven Project in Eclipse Luna

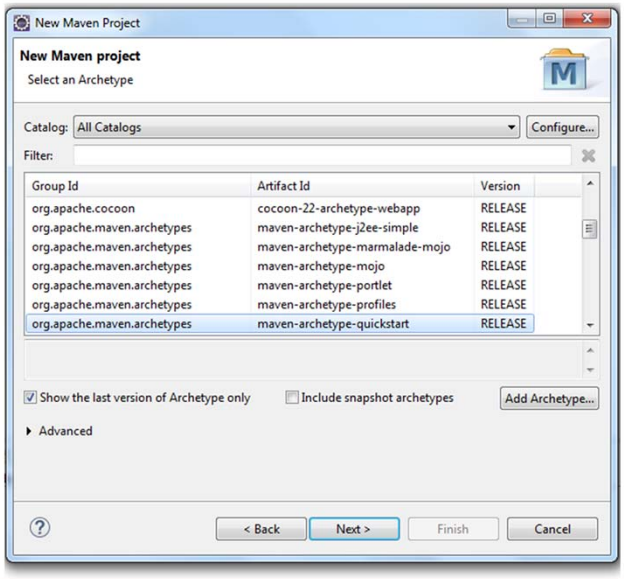


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 8

4.3.3 Maven Project in Eclipse

4.3 Creating Maven Project in Eclipse Luna

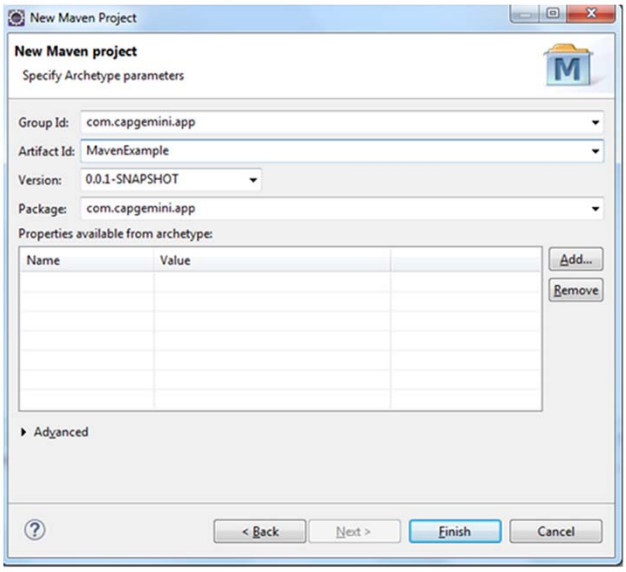


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 9

4.3.4 Maven Project in Eclipse

4.3 Creating Maven Project in Eclipse Luna



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

4.3.5 Maven Project in Eclipse

4.3 Creating Maven Project in Eclipse Luna

- Update the jUnit version to 4.11 in pom.xml

*MavenExample/pom.xml

```
1  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://w
2    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.a
3    <modelVersion>4.0.0</modelVersion>
4
5    <groupId>com.capgemini.app</groupId>
6    <artifactId>MavenExample</artifactId>
7    <version>0.0.1-SNAPSHOT</version>
8    <packaging>jar</packaging>
9
10   <name>MavenExample</name>
11   <url>http://maven.apache.org</url>
12
13   <properties>
14     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15   </properties>
16
17   <dependencies>
18     <dependency>
19       <groupId>junit</groupId>
20       <artifactId>junit</artifactId>
21       <version>4.11</version>
22       <scope>test</scope>
23     </dependency>
24   </dependencies>
25 </project>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

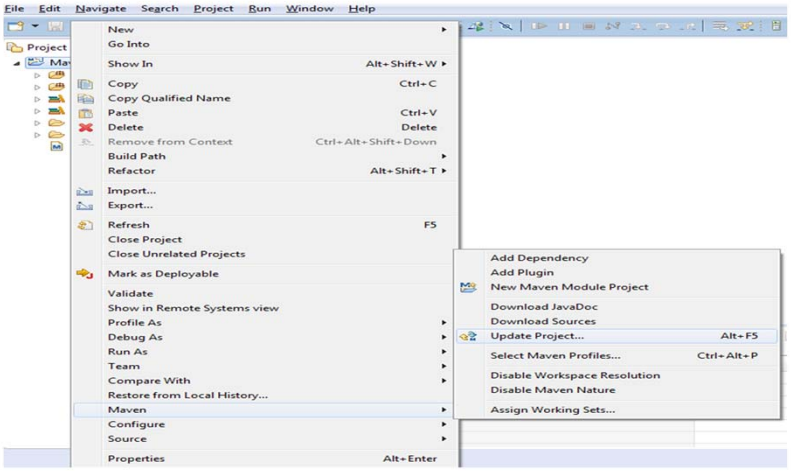
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11

4.3.6 Maven Project in Eclipse

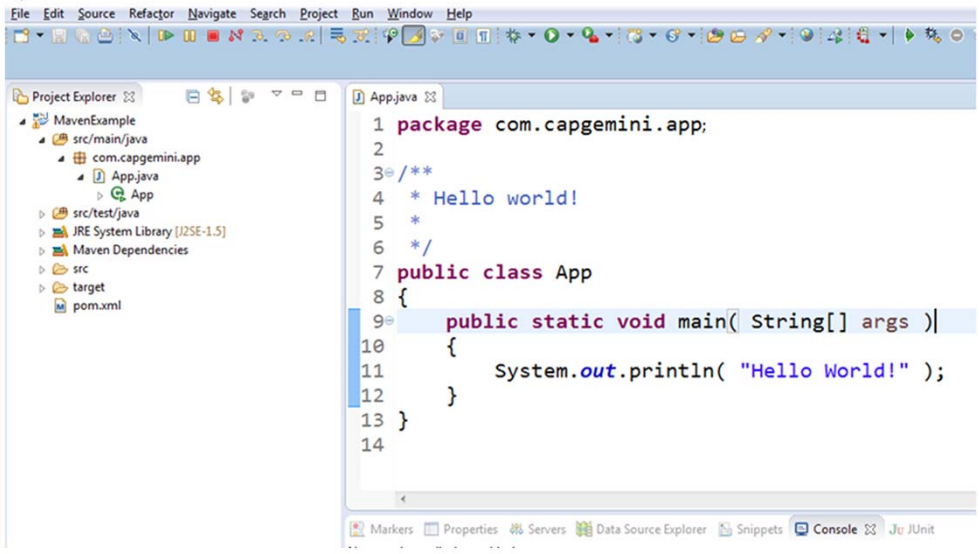
4.3 Creating Maven Project in Eclipse Luna

- Update the project as shown:



4.3.7 Maven Project in Eclipse

4.3 Creating Maven Project in Eclipse Luna

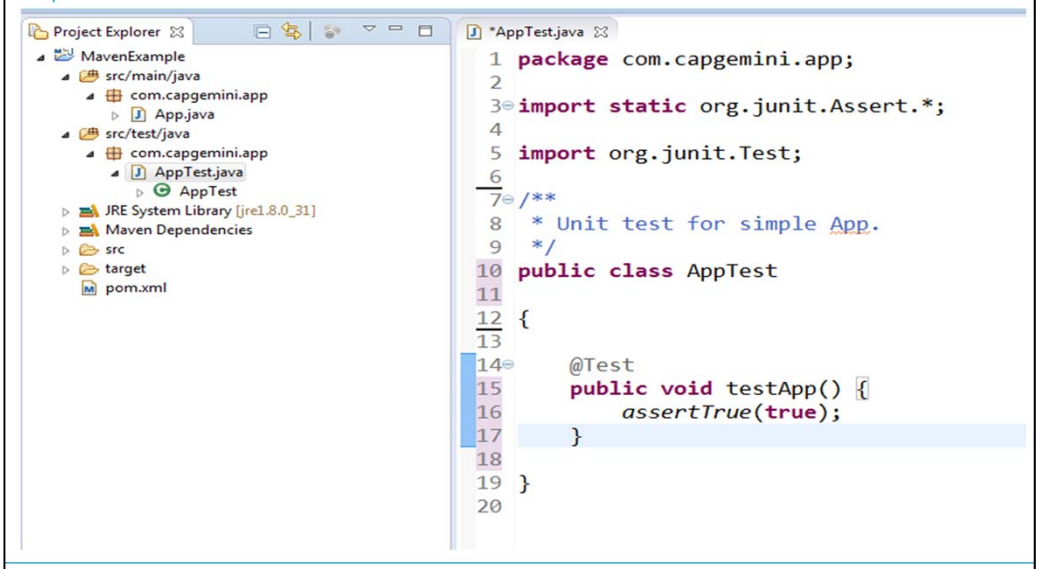


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 13

4.3.8 Maven Project in Eclipse

4.3 Creating Maven Project in Eclipse Luna

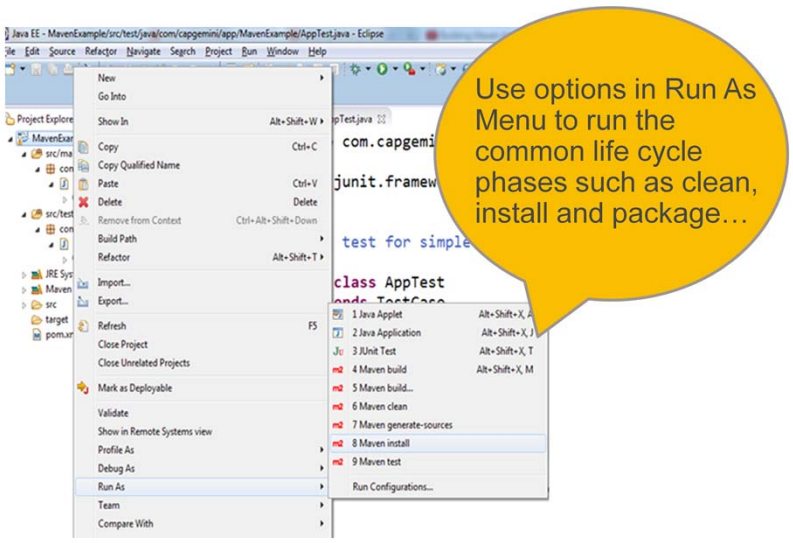


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 14

4.3.9 Maven Project in Eclipse

4.3 Creating Maven Project in Eclipse Luna



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 15

4.3.10 Adding and Updating Dependencies

4.3 Creating Maven Project in Eclipse Luna

- Eclipse offers two options for adding dependencies to a project.
  - By manually editing the POM file to type in the XML to add the dependency.
    - Open the pom.xml using XML Editor
    - Edit the pom by adding /updating dependency and save the file.
  - Searching for a dependency using groupId
    - Open the pom.xml using Maven POM Editor
    - Click on Dependency tab to add the dependency.
    - Search the dependency by entering groupId or artifactID.
    - Eclipse queries the dependency in repository indexes and even shows a version of the artifact that is currently in local Maven repository

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16

4.3 Creating Maven Projects in Eclipse Luna

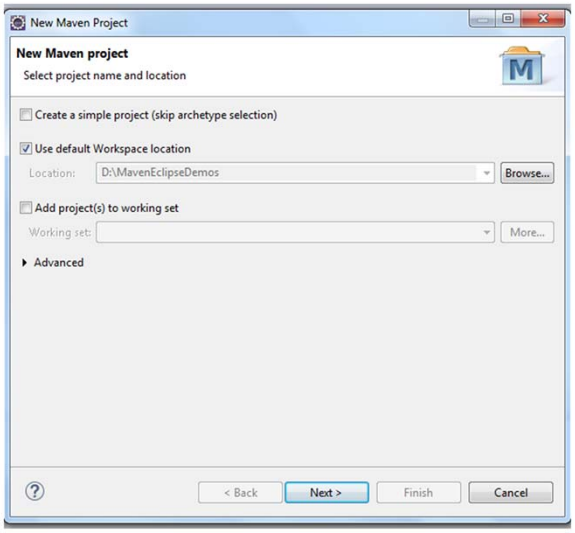
- Creating and executing the “Maven Example!” program

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 17

4.4.1 Adding and Updating Dependencies

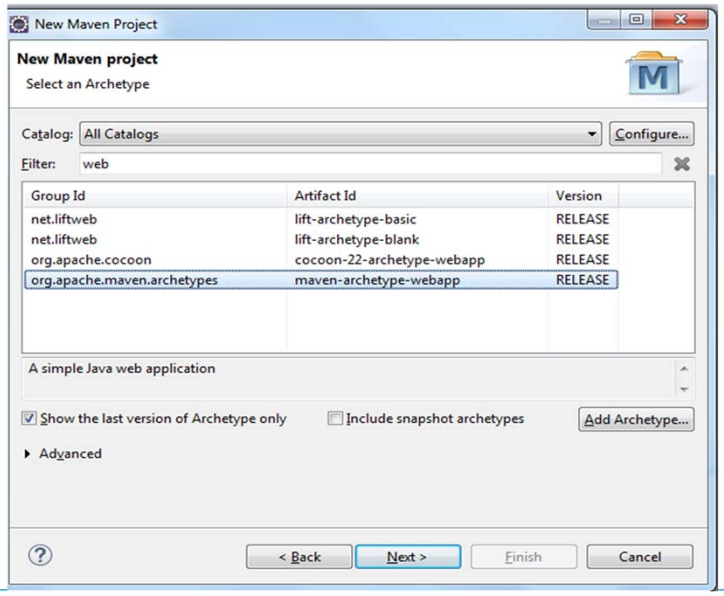
4.4 Creating Web application using Maven



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

4.4.2 Adding and Updating Dependencies

4.4 Creating Web application using Maven

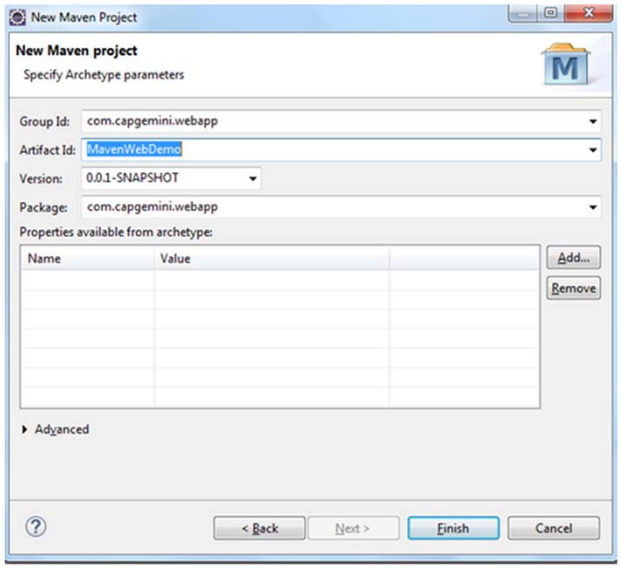


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 19

4.4.3 Adding and Updating Dependencies

4.4 Creating Web application using Maven

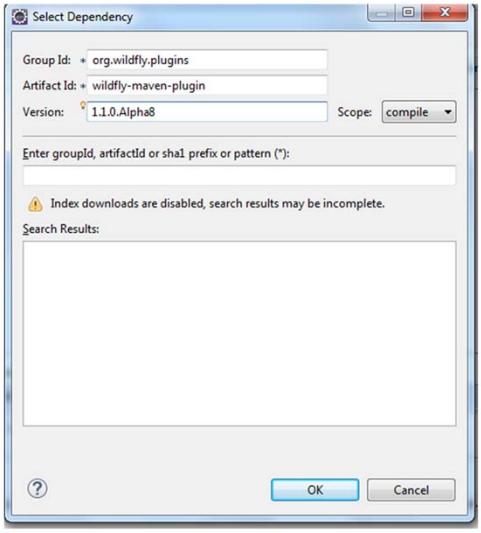


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 20

4.4.4 Adding and Updating Dependencies

4.4 Creating Web application using Maven



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 21

4.4.5 Adding and Updating Dependencies

4.4 Creating Web application using Maven

```
10   <dependencies>
11     <dependency>
12       <groupId>junit</groupId>
13       <artifactId>junit</artifactId>
14       <version> 4.11 </version>
15       <scope>test</scope>
16     </dependency>
17     <dependency>
18         <groupId>org.wildfly.plugins</groupId>
19         <artifactId>wildfly-maven-plugin</artifactId>
20         <version>1.1.0.Alpha8</version>
21     </dependency>
22   </dependencies>
23   <build>
24     <finalName>MavenWebDemo</finalName>
25   </build>
26 </project>
27
```

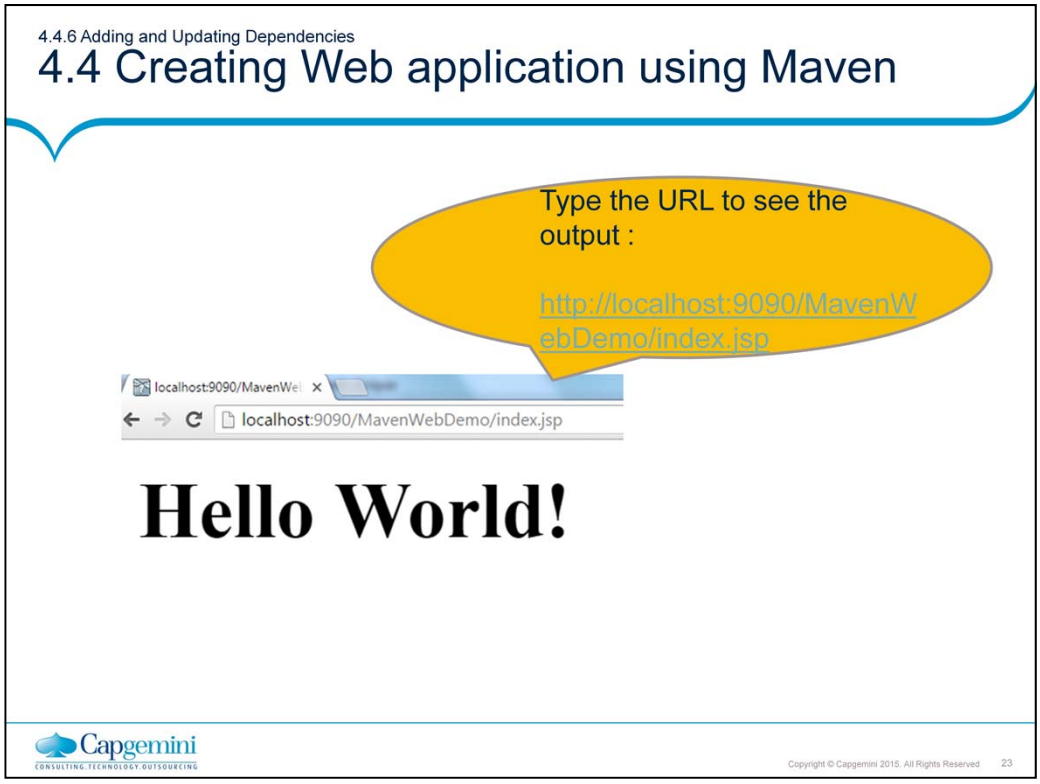
Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 22

4.4.6 Adding and Updating Dependencies

4.4 Creating Web application using Maven



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 23

4.4.7 Adding and Updating Dependencies

4.4 Creating Web application using Maven

Creating and executing the “Maven Example!” program

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 24

4.4.8 Integrating Maven with Eclipse

Lab: 2

- 2.1: Configure environment to run Maven project
- 2.2: Creating a sample Maven project

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 25

Summary

- In this lesson, you have learn about:
  - Overview
  - Testing your application
- Generating Java Projects using Maven
  - Using archetype
  - Creating Maven Module
- Generating Java web application Projects using Maven

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 26

Review Question

- Question 1: mvn deploy command executes all the build phases in the default lifecycle
  - True
  - False
- Question 2:Configuration of dependencies, is specified in which of the following files?
  - pom.xml
  - setting.xml
  - .m2 folder
  - test files

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 27

Review Question

- Question 3: _____repository for containing released artifacts and _____ repository for storing deployed snapshots.

- Question 4: Which of the following statements is true?
  - groupId and artifactId must be same in a project.
  - groupId and artifactId may be same in a project.
  - groupId and artifactId must not be same in a project.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 28