

Maven

Giacomo Tanganelli

PhD student @ University of Pisa

g.tanganelli@iet.unipi.it

Apache Maven - Concept

- Software project management tool
- Based on project object model (POM)
- POM is the fundamental unit (XML file)
 - information
 - configuration
 - dependencies
- de facto standard for large Java projects
- Java library exported in Maven repositories

POM - Minimal

<project>

<modelVersion>4.0.0</modelVersion>

<groupId>it.unipi.iot</groupId>

<artifactId>iot-example</artifactId>

<version>1.0</version>

<packaging>jar</packaging>

</project>

Project inheritance

<code><project></code>	<code> --/iot-example</code>
<code> <modelVersion>4.0.0</modelVersion></code>	<code> --/server</code>
<code> <parent></code>	<code> -- pom.xml</code>
<code><groupId>it.unipi.iot</groupId></code>	<code> </code>
<code><artifactId>iot-example</artifactId></code>	<code> --pom.xml</code>
<code><version>1.0</version></code>	
<code></parent></code>	
<code><groupId>it.unipi.iot</groupId></code>	
<code><artifactId>server</artifactId></code>	
<code><version>1.0</version></code>	
<code><packaging>jar</packaging></code>	
<code></project></code>	

Project aggregation

```
<project>  
  <modelVersion>4.0.0</modelVersion>  
  <groupId>it.unipi.iot</groupId>  
  <artifactId>iot-example</artifactId>  
  <version>1.0</version>  
  <packaging>pom</packaging>  
  <modules>  
    <module>server</module>  
  </modules>  
</project>
```

```
| --/iot-example  
|   --/server  
|     -- pom.xml  
| --pom.xml
```

Dependencies

```
<dependencies>
...
<dependency>
  <groupId>org.eclipse.californium</groupId>
  <artifactId>californium-core</artifactId>
  <version>1.1.0-SNAPSHOT</version>
</dependency>
...
</dependencies>
```

- Your project can rely on other project
- Specify each project as a dependency
- See it as “import jar into library path”
- Import from where?

Repositories



```
</repositories>
...
<repository>
  <id>repo.eclipse.org</id>
  <name>Californium
Repository</name>
  <url>https://repo.eclipse.org/content/rep
ositories/californium/</url>
</repository>
...
</repositories>
```

- Each pom.xml inherits the standard repository
- Some projects, i.e. Californium, have their repository

Plugins



```
<build>  
  <pluginManagement>  
    <plugins>  
      ...  
    </plugins>  
  </pluginManagement>  
</build>
```


Plugins



```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-jar-plugin</artifactId>
  <version>3.0.2</version>
  <configuration>
    <archive>
      <manifest>
        <addClasspath>true</addClasspath>
        <mainClass>$PACKAGE.$MAINCLASS</mainClass>
      </manifest>
      <addDefaultImplementationEntries>true</addDefaultImplementationEntries>
    </archive>
    <descriptorRefs>
      <descriptorRef>jar-with-dependencies</descriptorRef>
    </descriptorRefs>
  </configuration>
</plugin>
```

Compile and Install

- In order to compile modules in the root of the project run:
 - `$> mvn clean install`
- Usually jar files will be deployed inside a target folder in each subproject:
 - Californium configuration, for convenience, copies generated jars inside the run folder in the root folder.

Example

- Take a look to `cf-helloworld-server/pom.xml`
- Take a look to `californium/pom.xml`
- Try to create a new Maven project ready to use the californium library.