# 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

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

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

# 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>${project.version}</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/
repositories/californium/</url>
  </repository>
 …
 </repositories>
```

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

# Plugins


```
<build>
      <plugins>
          <plugin>
              <groupId>org.apache.maven.plugins</groupId>
              <artifactId>maven-jar-plugin</artifactId>
              <version>2.2</version>
              <!-- nothing here -->
          </plugin>
```

# Plugins – 2

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-assembly-plugin</artifactId>
    <configuration>
        <appendAssemblyId>false</appendAssemblyId>
        <archive>
            <manifest>
                <addClasspath>true</addClasspath>
                <mainClass>org.eclipse.californium.examples.HelloWorldServer</mainClass>
                <addDefaultImplementationEntries>true</addDefaultImplementationEntries>
            </manifest>
        </archive>
        <descriptorRefs>
            <descriptorRef>jar-with-dependencies</descriptorRef>
        </descriptorRefs>
    </configuration>
    <executions>
        <execution>
            <id>make-assembly</id>
            <phase>package</phase>
            <goals>
                <goal>single</goal>
            </goals>
        </execution>
    </executions>
</plugin>
```

# Plugins – 3

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-dependency-plugin</artifactId>
    <executions>
        <execution>
            <id>copy-installed</id>
            <phase>install</phase>
            <goals>
                <goal>copy</goal>
            </goals>
            <configuration>
                <artifactItems>
                    <artifactItem>
                        <groupId>${project.groupId}</groupId>
                        <artifactId>${project.artifactId}</artifactId>
                        <version>${project.version}</version>
                        <type>${project.packaging}</type>
                    </artifactItem>
                </artifactItems>
                <outputDirectory>../run/</outputDirectory>
            </configuration>
        </execution>
    </executions>
</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.