

Prova Finale

Ingegneria del Software

Fare i JAR con Maven

23/05/2023

Scaglione San Pietro

RESPONSABILI

Alessandro Bertani

Tommaso Paladini

TUTOR

Silvia Marino

Patrick Niantcho

Slide basate su materiale di A. Nemirowsky e D. Cattaneo

Cosa serve fare

- La configurazione di Maven di default crea i JAR impacchettando le vostre classi e basta
 1. Non sono incluse le dipendenze
 - se usate librerie esterne come GSON il JAR non funzionerà
 2. Non viene configurata la classe col main
 - fare doppio click sul JAR non funziona
- Come creare un JAR eseguibile quindi?

Maven Shade Plugin

- Plugin per impacchettare **l'intero progetto** **comprese le sue dipendenze** in un unico JAR
 - Termine tecnico: "Uber Jar"
- Una volta inserito nel POM sostituisce la fase "package" di default

Maven Shade Plugin

1. Modificate il .pom aggiungendo l'integrazione che definisce il plugin
2. Per ogni JAR da creare aggiungete una *execution* del plugin
3. In ogni execution, impostate la main class (entry point) del JAR

Cosa aggiungere al .pom

Prima di tutto copiate e incollate la dichiarazione del plugin all'interno del POM così com'è

```
<project>
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-shade-plugin</artifactId>
      <version>3.2.4</version>
      <executions>
        <execution>
          <id>main-jar</id>
          <phase>package</phase>
          <goals>
            <goal>shade</goal>
          </goals>
          <configuration>
            <transformers>
              <transformer implementation=
                "org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">
                <mainClass>it.polimi.ingsw.psp99.MainClass</mainClass>
              </transformer>
            </transformers>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
...
</project>
```

Cosa aggiungere al .pom

```
<project>
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-shade-plugin</artifactId>
      <version>3.2.4</version>
      <executions>
        <execution>
          <id>main-jar</id>
          <phase>package</phase>
          <goals>
            <goal>shade</goal>
          </goals>
          <configuration>
            <transformers>
              <transformer implementation=
                "org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">
                <mainClass>it.polimi.ingsw.psp99.MainClass</mainClass>
              </transformer>
            </transformers>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
...
</project>
```

Per ogni JAR che volete produrre
(ad es. un JAR per la CLI, un JAR per
la GUI e un JAR per il Server)
aggiungete una nuova <execution>

Ogni execution dovrà avere un id
(scelto a piacere) e una mainClass
diversa, il resto resterà uguale

Materiale aggiuntivo

- Documentazione di Maven Shade
<http://maven.apache.org/plugins/maven-shade-plugin/>

Optional:

Ridurre la dimensione del POM

- Maven Shade permette di eliminare dal JAR le classi "inutili"
 - Esempio: Eliminare le classi GUI/CLI dal server
- Filtro delle classi in base al nome dei package
- Se volete provarci, leggete la documentazione:
<https://maven.apache.org/plugins/maven-shade-plugin/examples/includes-excludes.html>
- **Attenzione!! Se togliete troppe classi rischiate di produrre un JAR corrotto**