

# Formation Apache Maven

TP 02 - Plugins

# Table of Contents

Etape 1 - Créer un jar exécutable .....	1
Etape 2 - Créer un uber jar .....	2

# Etape 1 - Créer un jar exécutable

- Générer un package avec Maven :

```
mvn clean package
```

- Exécuter le JAR avec la commande :

```
java -jar target/premier-projet-maven.jar
```

Une erreur s'affiche, le JAR généré n'est pas exécutable. Pour qu'une archive JAR soit exécutable, il doit contenir un fichier *META-INF/MANIFEST.MF* dont un attribut *Main-Class* est renseigné.

- Configurer le plugin *maven-jar-plugin* (<https://maven.apache.org/plugins/maven-jar-plugin/usage.html>) :

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  ...

  <build>
    <finalName>${project.artifactId}</finalName>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>
        <version>2.4</version>
        <configuration>
          <archive>
            <manifest>
              <mainClass>dev.App</mainClass> ①
            </manifest>
          </archive>
        </configuration>
      </plugin>
    </plugins>
  </build>

  ...
</project>
```

① : chemin vers la classe principale du projet.

- Exécuter le JAR avec la commande

```
java -jar target/premier-projet-maven.jar
```

Le programme se lance mais il y a une erreur *NoClassDefFoundError* :

```
Exception in thread "main" java.lang.NoClassDefFoundError:  
com/github/lalyos/jfiglet/FigletFont  
    at dev.App.main(App.java:13)  
Caused by: java.lang.ClassNotFoundException: com.github.lalyos.jfiglet.FigletFont  
    at java.net.URLClassLoader.findClass(URLClassLoader.java:381)  
    at java.lang.ClassLoader.loadClass(ClassLoader.java:424)  
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:331)  
    at java.lang.ClassLoader.loadClass(ClassLoader.java:357)  
    ... 1 more
```

Comprenez-vous pourquoi ?

## Etape 2 - Créer un uber jar

- Utiliser le plugin *maven-shade-plugin* pour créer un *uber-jar* qui contient toutes les classes compilées du projet (y compris celles des dépendances) :

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  ...

  <build>
    <finalName>${project.artifactId}</finalName>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>

        ...

      </plugin>
      <!-- configuration du plugin shade -->
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-shade-plugin</artifactId>
        <version>3.0.0</version>
        <executions>
          <execution>
            <phase>package</phase>
            <goals>
              <goal>shade</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>

  ...
</project>

```

- Exécuter le JAR avec la commande

```
java -jar target/premier-projet-maven.jar
```

Vérifier que le programme fonctionne correctement.

- **PAUSE GIT** : Sauvegarder dans Github.