

MAVEN

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
  <groupId>fr.afcepf.qualimetrie</groupId>
  <artifactId>banque-data</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>ejb</packaging>
  <dependencies>
    <dependency>
      <groupId>javaee</groupId>
      <artifactId>javaee-api</artifactId>
      <version>5</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
  <build>
  </build>
</project>
```

modelVersion :

liaison d'un projet à l'objet modèle utilisé

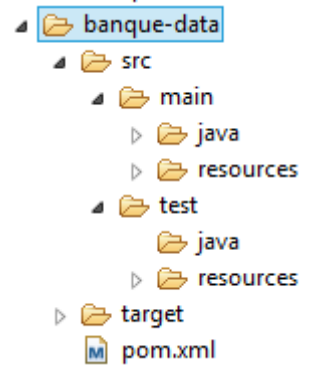
groupId : permet de rattacher le projet à l'unité organisationnelle

artifactId : nom unique de l'artefact produit par maven

version : la version de l'artefact

packaging : le type de packaging utilisé par le projet : jar, war, ear, ejb, jar, bundle ...

Structure d'un projet maven



Un descripteur maven

Distinction entre les sources réelles de l'application et les ressources de tests

Des répertoires de sorties pour le résultat de la compilation.

Dépendances

```
<dependencies>
  <dependency>
    <groupId>javaee</groupId>
    <artifactId>javaee-api</artifactId>
    <version>5</version>
    <scope>provided</scope>
    <type>jar</type>
  </dependency>
</dependencies>
```

les dépendances sont utilisées par maven au cours des phases du cycle de construction

scope : compile, provided, runtime, test, system, import

type : jar, war, ear, ejb, pom ... le type de packaging de la dépendance

Cycle de vie par défaut

1. validate	2. initialize	3. generate-sources	4. process-sources	5. generate-resources
6. process-resources	7. compile	8. process-classes	9. generate-test-sources	10. process-test-sources
11. generate-test-resources	12. process-test-resources	13. test-compile	14. process-test-classes	15. test
16. prepare-package	17. package	18. pre-integration-test	19. integration-test	20. post-integration-test
21. verify	22. install	23. deploy		

exemple :
mvn compile
mvn install

Commandes utiles

création d'un projet java:

```
mvn archetype:create -DgroupId=fr.afcepf.formation -DartifactId=banque-data
```

création d'un projet web:

```
mvn archetype:create -DgroupId=fr.afcepf.formation -DartifactId=banque-web
-DarchetypeArtifactId=maven-archetype-webapp
```

afficher la description d'un plugin:

```
mvn help:describe -Dplugin=source
```

génération d'un site de reporting:

```
mvn site:site
```

création d'un livrable:

```
mvn release:prepare -DtagBase=svn://repo/tags
mvn release:perform -Dgoals=install
```