



Maven



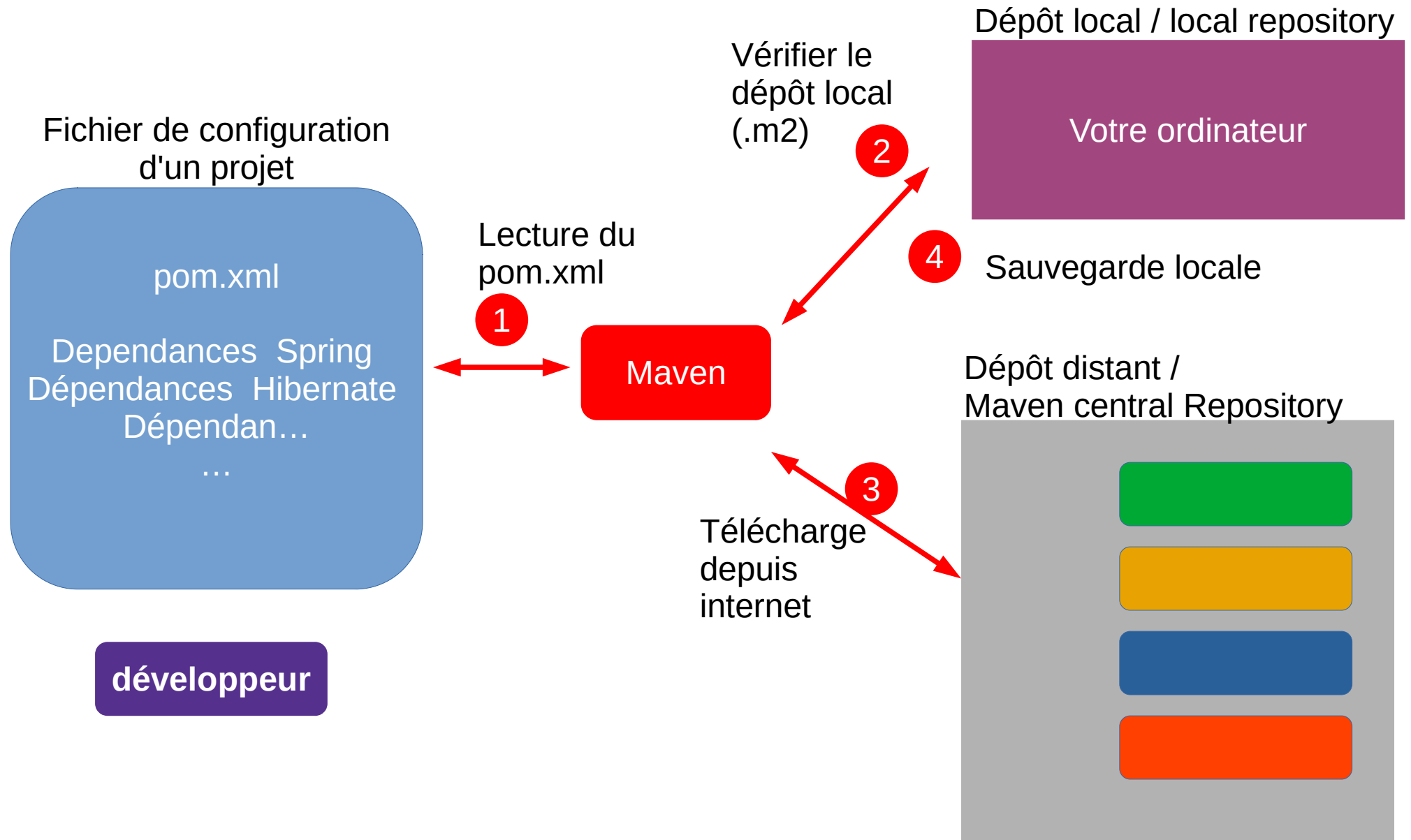
Maven c'est quoi

- Maven est un outil pour gérer votre application, très populaire concernant
 - La gestion les dépendances d'un projet
 - La gestion de la construction d'un projet (build)

Avant – après maven

- Jusqu'à présent vous téléchargez les fichiers .jar , vous les placez dans votre application, puis vous les référencez dans le classpath.
- Dans un projet maven , vous allez déclarer les fichiers .jar dans un fichier pom.xml qui se situe dans votre application (à la racine), et c'est maven qui récupèrera les dépendances sur internet. Le classpath sera configuré sans intervenir pour chaque dépendance.

Comment ça marche ?



Dépendance en chaine

- Quand maven télécharge "*votre*" dépendance , il va également télécharger les dépendances dont "*votre*" dépendance dépend :-)
- Maven va réaliser ce travail automagiquement.
- Le classpath sera mis à jour également.

Maven et la structure du projet

- Maven considère qu'un projet possède une structure donnée, correspondant à une pratique répandue majoritairement. Cette structure est celle retenue par défaut (pour mettre à jour le classpath automatiquement par exemple).
- Cependant il est possible de configurer une structure de projet différente, selon vos souhaits.

Structure standard des repertoires

Repertoire	Description
src/main/java	Code source java
src/main/resources	Fichiers properties, et/ou de configuration
src/main/webapp	Pages JSP, fichiers web, et autres .js, .css, images...
src/test	Tests unitaires et properties de test
target	Répertoire de destination pour le code compilé. Il est créé automatiquement par maven

NB : A la racine de votre projet se trouve un fichier pom.xml



Maven vs IDE

- De plus les IDE du marché ont un support de maven natif

On peut executer des commandes maven depuis l'IDE

Lire, importer des projets maven.

- Les projets maven sont portables d'un IDE à un autre

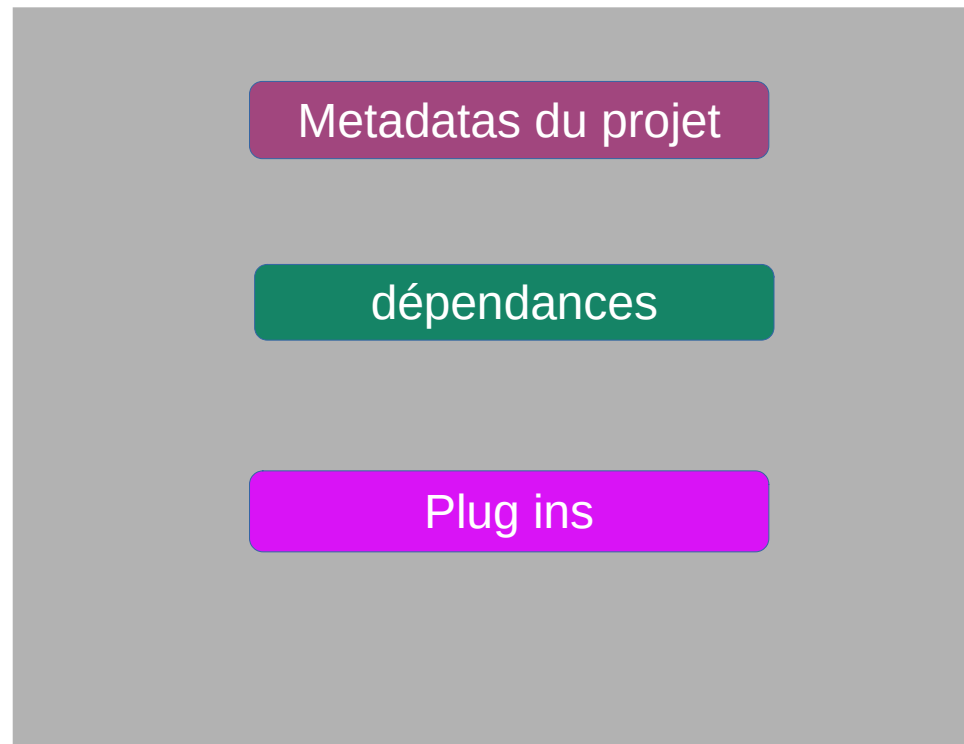


pom.xml

- Le fichier POM = project Object Model
- Il s'agit du fichier de configuration de votre projet dans lequel vous allez indiquer la liste des dépendances à installer dans ce projet, mais également d'autres informations.
- Il se situe obligatoirement à la racine de votre projet

Structure du pom.xml

pom.xml



Exemple de fichier pom.xml

```
<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.ecolepratique</groupId>
  <artifactId>masuperapp</artifactId>
  <version>1.0.FINAL</version>

  <name>masuperapp</name>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <!-- ajouter des plugins selon vos besoins -->

</project>
```

Les coordonnées du projet

- L'objectif, c'est que l'ensemble des coordonnées permettent d'identifier de manière unique votre projet

```
<groupId>com.ecolepratique</groupId>  
<artifactId>masuperapp</artifactId>  
<version>1.0.FINAL</version>
```

Nom	Description
groupId	Nom de votre entreprise, groupe, ou organisation. Par convention , on retourne le nom de domaine : com.ecolepratique
artifactId	Nom de votre projet : masuperapp
version	Un numéro de "sortie" (release) : 1.0, 2.4 ... Si le projet est toujours en cours de développement alors on indique une version de type : 1.0-SNAPSHOT

Ajoutons 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 https://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
  <modelVersion>4.0.0</modelVersion>
```

```
  <groupId>com.ecolepratique</groupId>
```

```
  <artifactId>masuperapp</artifactId>
```

```
  <version>1.0.FINAL</version>
```

```
  <name>masuperapp</name>
```

```
  <dependencies>
```

```
    <dependency>
```

```
      <groupId>org.springframework</groupId>
```

```
      <artifactId>spring-context</artifactId>
```

```
      <version>5.0.0.RELEASE</version>
```

```
    </dependency>
```

```
    <dependency>
```

```
      <groupId>org.hibernate</groupId>
```

```
      <artifactId>hibernate-core</artifactId>
```

```
      <version>5.2.11.Final</version>
```

```
    </dependency>
```

```
  </dependencies>
```

```
  <!-- ajouter des plugins selon vos besoins -->
```

```
</project>
```



Comment trouver les GAV de vos libraires ?

- 1ere option : consulter la page de chaque projet concerné (spring.io, hibernate.org, ...etc).
- 2ème option : visitez le site search.maven.org (plus facile)
- 3ème option : mvnrepository.com



Créer un projet Maven : archetypes

- On utilise les archetype pour créer un nouveau projet Maven. L'archetype correspond à un modèle de projet Maven.
- Pour un projet web correspondra un archetype de projet, pour un projet java on utilisera un autre modèle de projet ...etc. (archetype:=starter project)

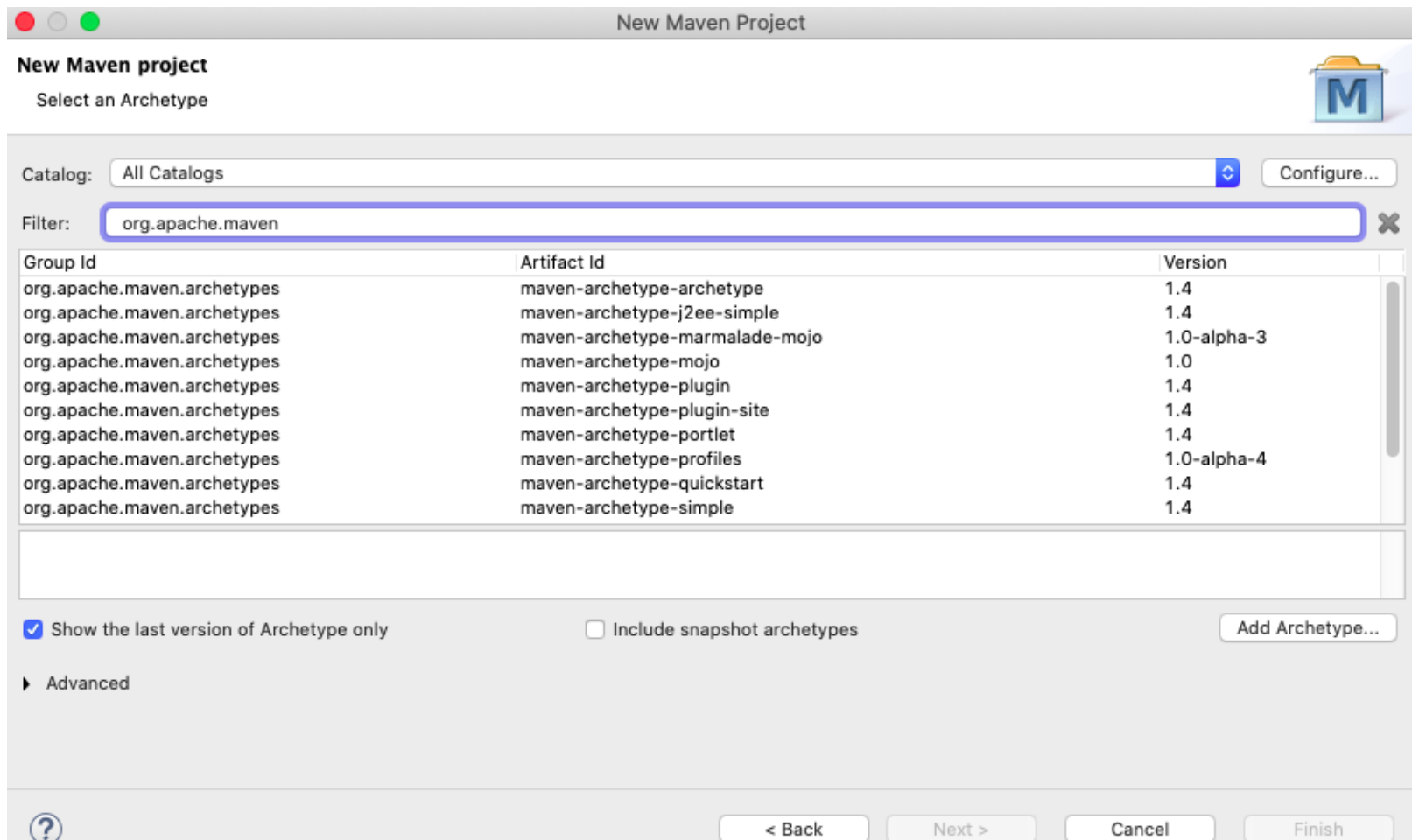
Les archetypes courants :

Archetype Artifact ID	Description
maven-archetype-quickstart	Un archetype pour générer un projet stand alone
maven-archetype-webapp	Un archetype pour générer un projet web
... autres ...	Cf : http://maven.apache.org/archetypes

On va pouvoir créer un projet maven et utiliser un archetype depuis la ligne de commande mais aussi en mode graphique via les IDE.

Archetypes et Eclipse

- New > Maven project > suivant



Quickstart Archetype

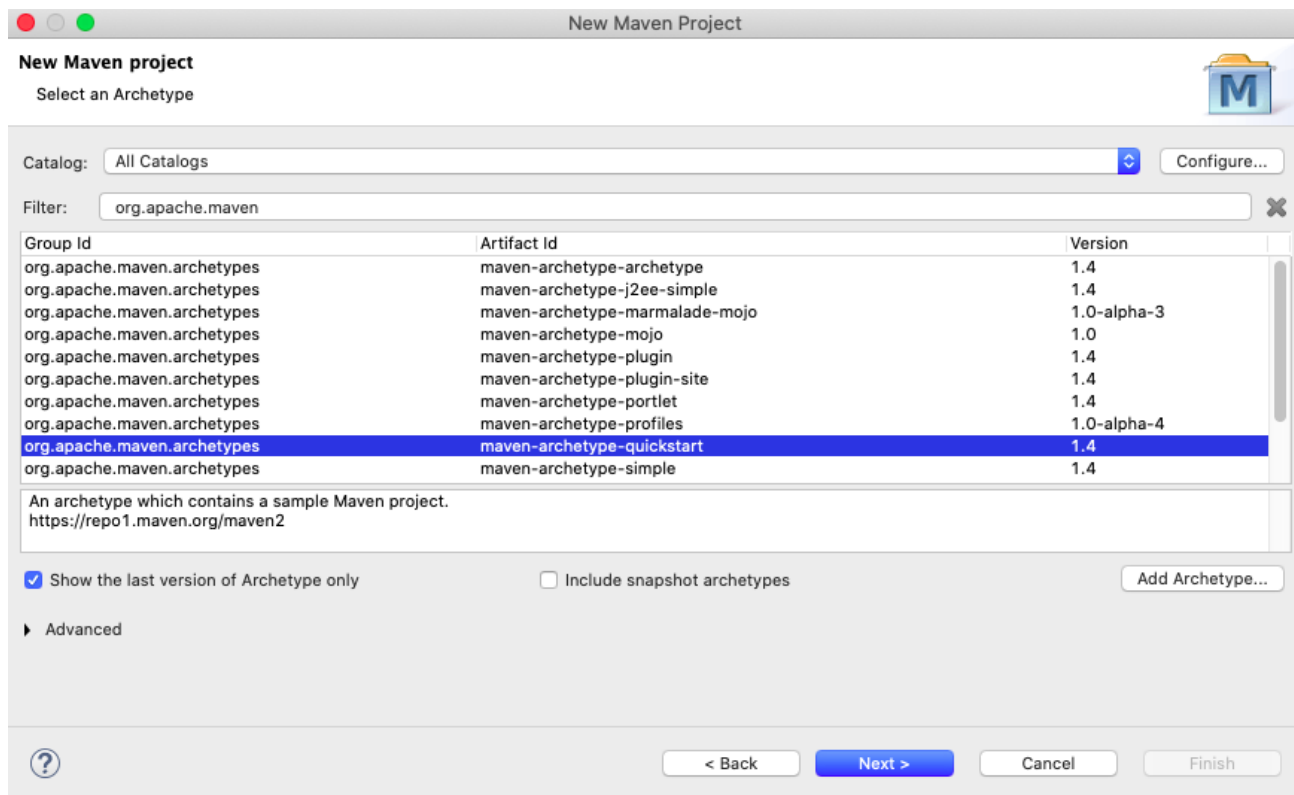
projet

pom.xml

src/main/java/App.java

src/test/java/AppTest.java

- Il contient un projet Maven initial, c'est à dire vide, avec juste la structure du projet.



Quickstart Archetype (suite)

New Maven Project

Specify Archetype parameters

Group Id: quickstartArchetype

Artifact Id: projetStandAloneDemo

Version: 0.0.1-SNAPSHOT

Package: com.tld

Properties available from archetype:

Name	Value

Advanced

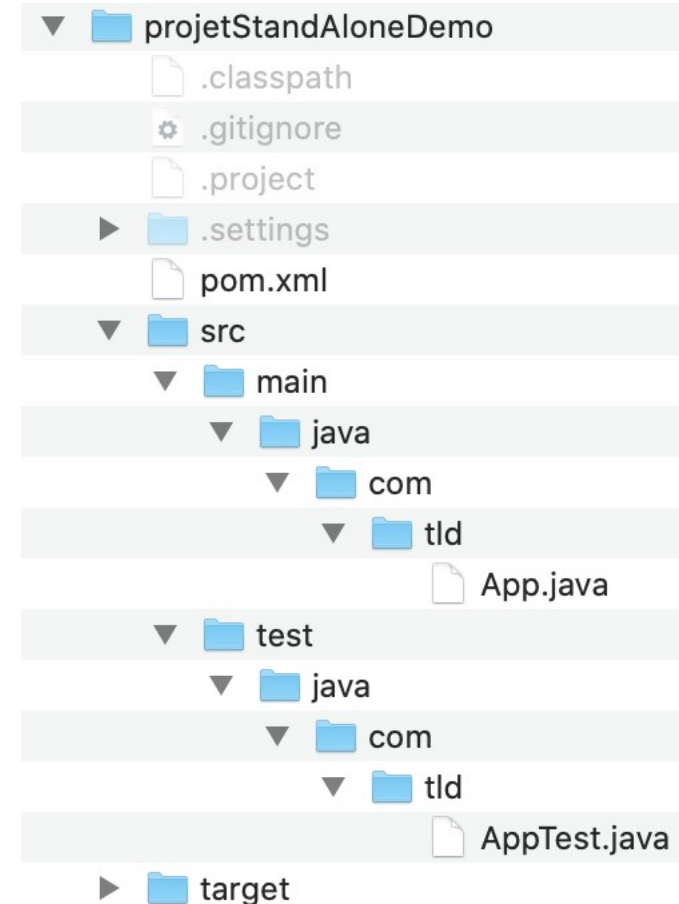
< Back Next > Cancel Finish

projet

pom.xml

src/main/java/App.java

src/test/java/AppTest.java



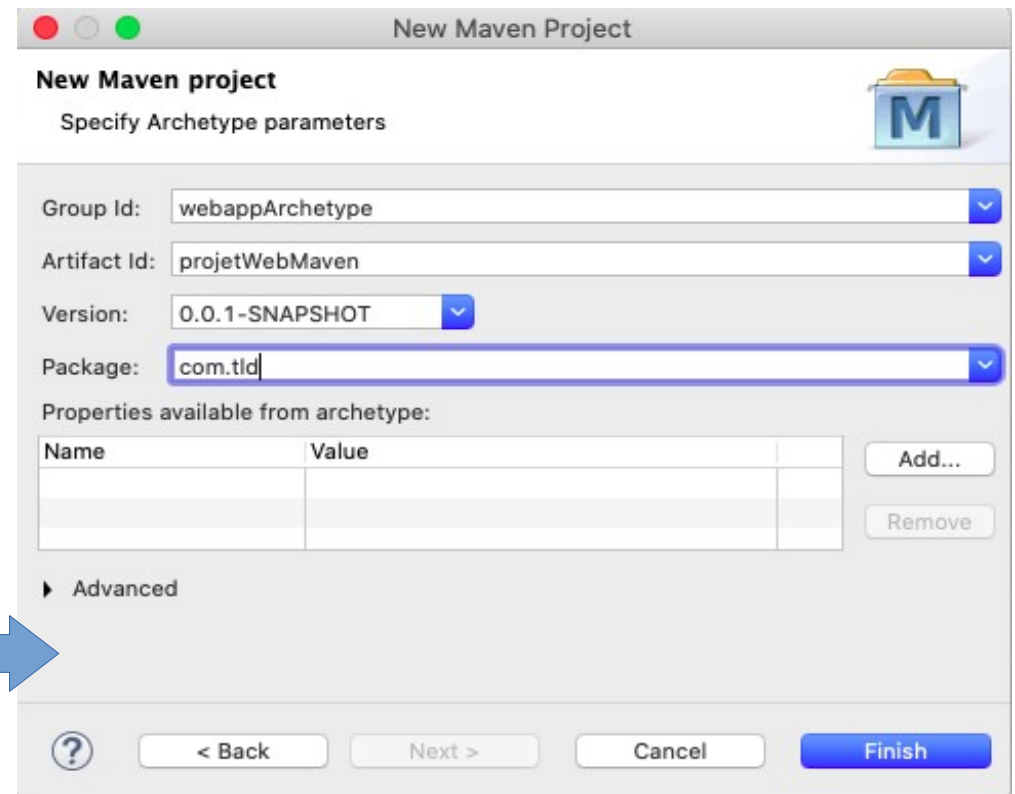
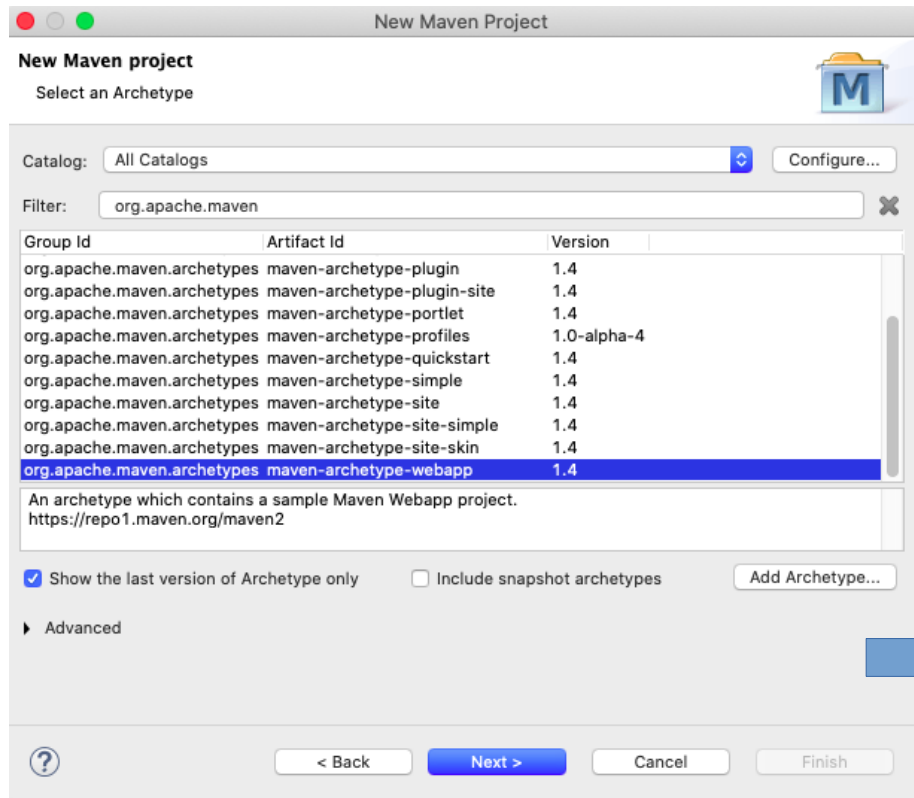
Webapp Archetype

projet

pom.xml

src/main/webapp/index.jsp

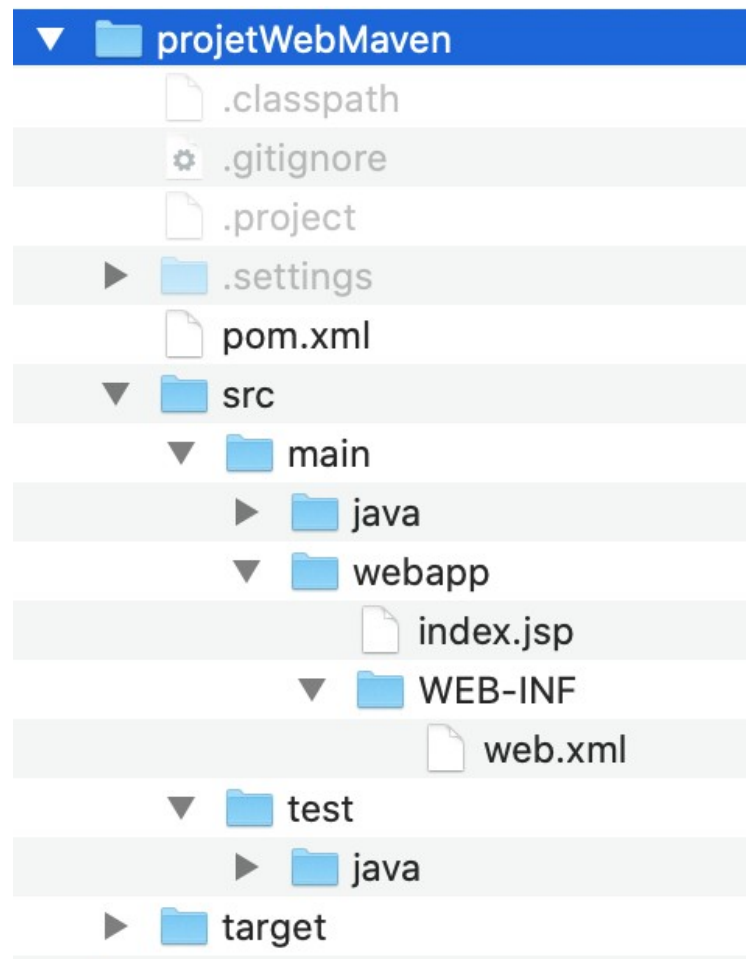
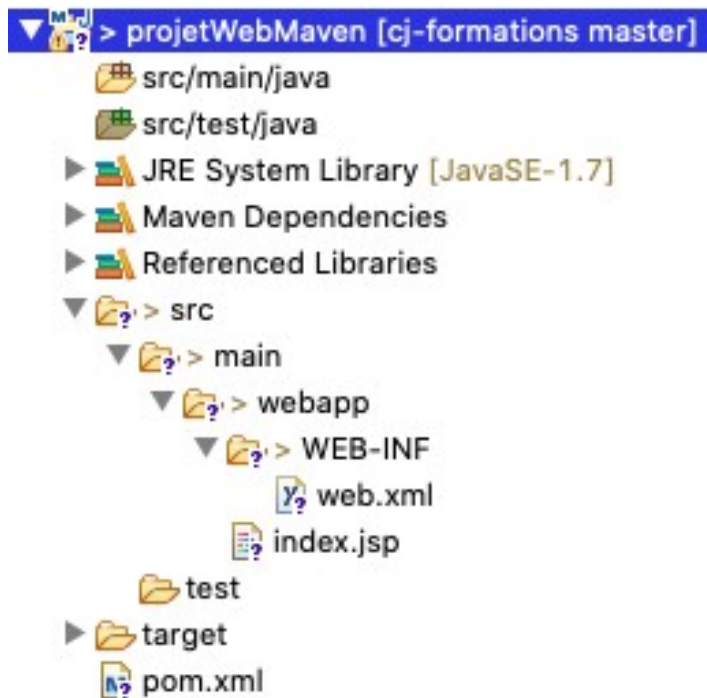
src/main/webapp/WEB-INF/web.xml



Webapp archetype (suite)

projet

- pom.xml
- src/main/webapp/index.jsp
- src/main/webapp/WEB-INF/web.xml





TD Créer une application maven

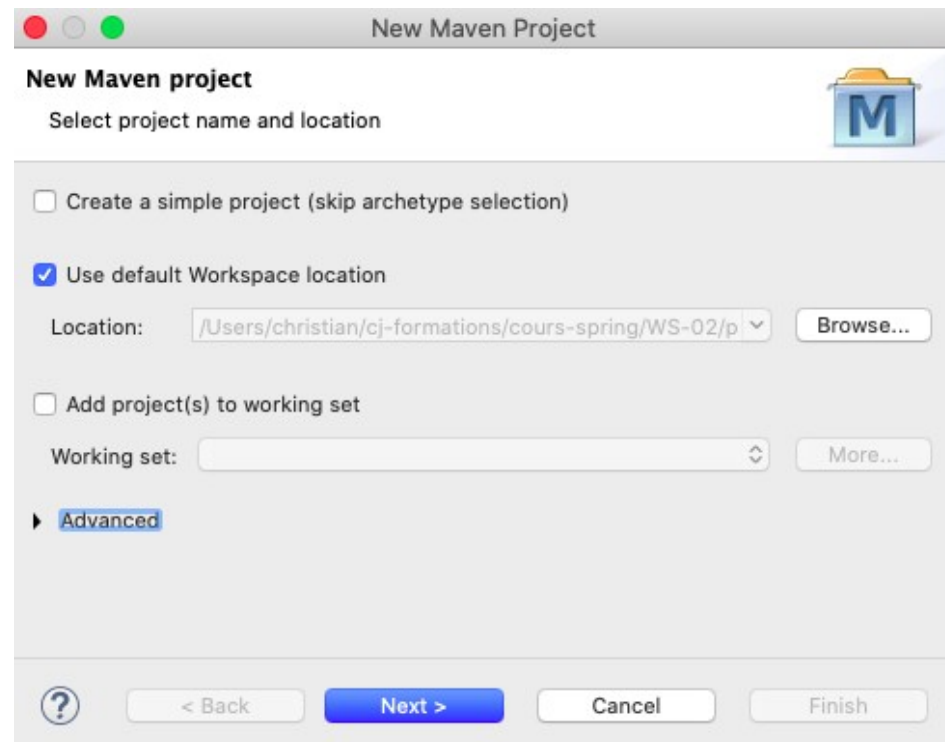
- Créer un simple Project avec Eclipse
- Exécuter notre application
- Mettre à jour notre version de java utilisée

Eclipse = m2eclipse plugin

- Inutile de télécharger et d'installer maven séparément
- Eclipse utilise un plugin interne contenant maven : m2eclipse plugin
- Eclipse > Help> Install new software> [What is already installed ?](#)
- On va utiliser maven à l'intérieur d' eclipse :
 - créer un projet
 - Ajouter des dépendances

Créer un projet maven avec Eclipse

- Eclipse> menu File>New > other
- Maven> Maven Project> Next
- Garder les réglages par défaut



>Next

Créer projet maven avec Eclipse (suite)

- Sélectionner l'archetype (filtrer avec : quickstart)
- >Next
- >saisir les coordonnées GAV de votre projet
- >Finish