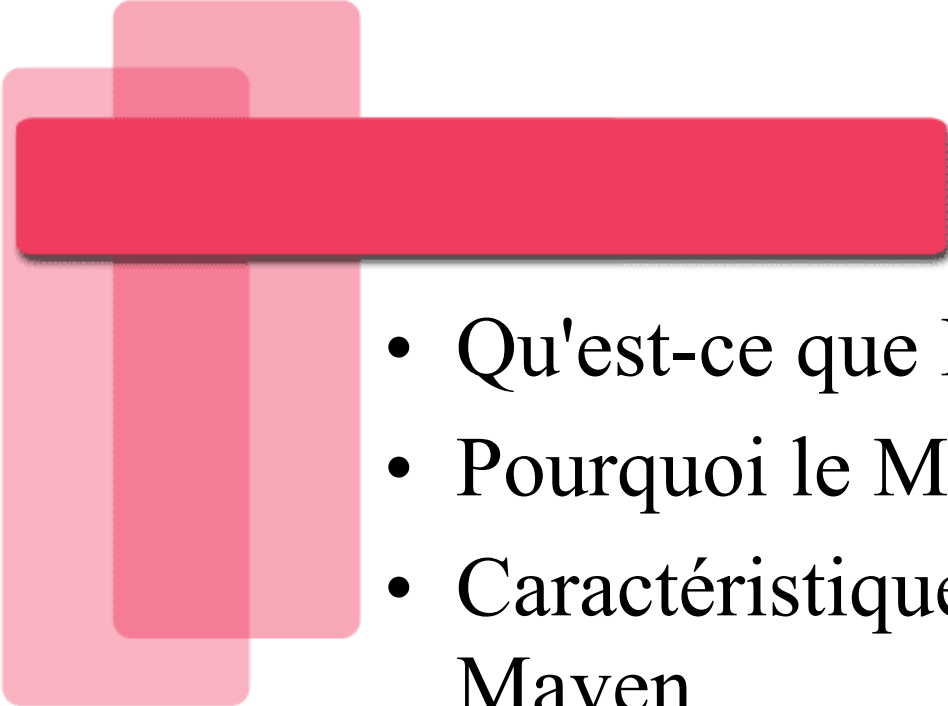




# Maven

par Mireille Khalifeh



- 
- A decorative graphic on the left side of the slide consists of three overlapping pink squares of varying shades and a solid red horizontal bar positioned above the list.
- Qu'est-ce que Maven?
  - Pourquoi le Maven?
  - Caractéristiques Principales De Maven
  - Installation de Maven
  - POM.xml
  - Références

# Qu'est-ce que Maven?

- Maven est un outil open source de la communauté Apache entièrement écrit en Java.
- Outil pour la gestion et l'automatisation de production des projets logiciels Java.
- Basé sur le format XML Project Object Model (POM).
- Maven se présente comme un exécutable en ligne de commande mais il est également intégré nativement dans les IDE les plus courants dans le monde Java : Eclipse, IntelliJ IDEA, NetBeans.


A decorative graphic on the left side of the slide consisting of three overlapping pink squares of different shades. The top square is a medium pink, the middle one is a darker pink, and the bottom one is a lighter pink.

## Pourquoi le Maven?

- Comment partager mon projet avec d'autres développeurs quand ils n'ont pas exactement la même configuration de poste que la mienne et qu'ils n'utilisent pas le même IDE que moi ?
- Comment compiler et tester mon projet en dehors d'un IDE ?
- Comment automatiser certaines tâches répétitives et limiter ainsi les erreurs ou les oublis ?



## Caractéristiques principales de Maven

- Découpe le cycle de construction du projet en phases pré-définies.
  - Gérer automatiquement les dépendances logicielles.
  - Garantir une homogénéité entre les projets
- 

## Build Tools

- Un niveau au-dessus de ant.
- Make  $\rightarrow$  ant  $\rightarrow$  Maven
- (Assembly  $\rightarrow$  C  $\rightarrow$  C++)

Make	makefile	target
Ant	build.xml	target
Maven	project.xml maven.xml	goals



jalopy	site	junit	etc...
test	cactus	artifact	xdoc
ear	gump	xdoclet	javadoc
java	jar	war	ejb

maven core, jelly, werkz
ant

forehead classloader util launcher
------------------------------------

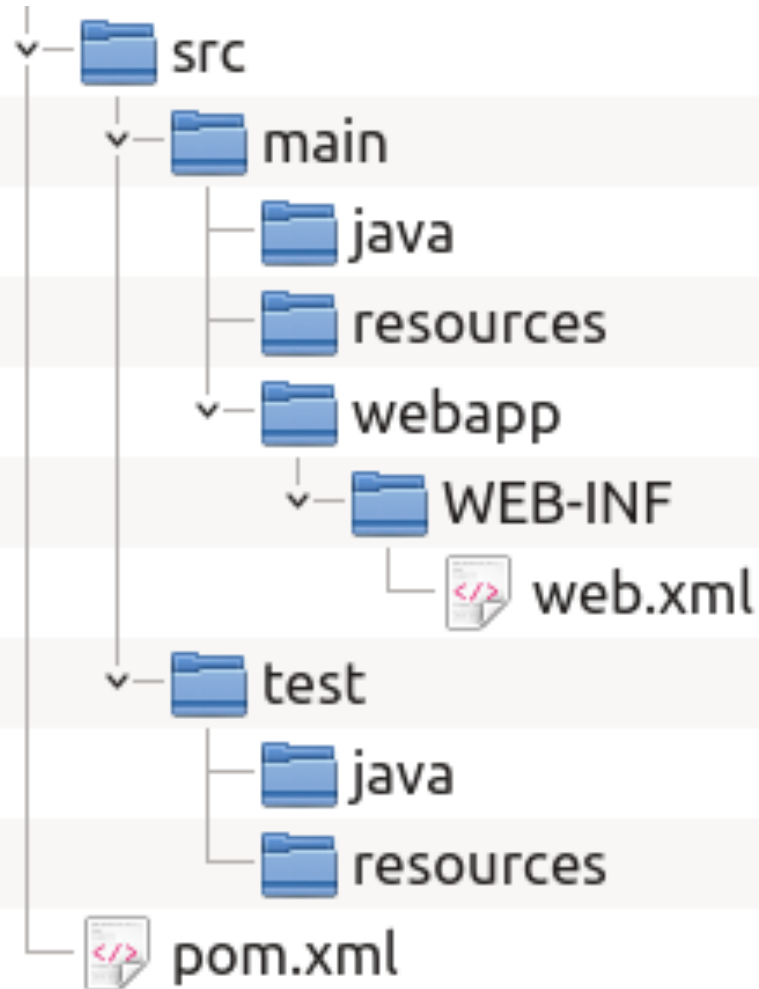
Maven

# Installation de Maven

- `mvn -version`
- `brew install maven`
- `vim ~/.bash_profile`
- On copie ces lignes:  
`export M2_HOME=/Users/username/somefolder/  
apache-maven-3.3.9`  
`export M2=$M2_HOME/bin`  
`export JAVA_HOME=/Library/Java/Home/`  
`export PATH=$PATH:$M2`
- `mvn -version`



# Arborescence des fichiers





pom.xml

```
<!--  
  La version du format du fichier pom.  
  Actuellement la dernière version est la 4.0.0.  
-->  
<modelVersion>4.0.0</modelVersion>
```



pom.xml

<!--

Le group ID de l'application. Le group ID s'apparente à un package Java mais pour un projet. Il évite une collision de nom dans le cas de deux projets ayant le même nom puisqu'ils peuvent avoir des group ID différents. Ainsi si deux projets s'appelle maven-demo et qu'ils ont des group ID différents, ils sont considérés comme étant des projets différents.

-->

<groupId>com.mireille</groupId>



pom.xml

```
<!--  
  Le nom du projet  
-->  
<artifactId>maven-demo</artifactId>
```

pom.xml

```
<!--
```

La version de notre projet. Maven gère le versionnage afin de permettre le suivi des évolutions d'un projet.

Ici, le suffixe "-SNAPSHOT" indique à Maven que le projet est en cours de développement pour cette version.

```
-->
```

```
<version>0.0.1-SNAPSHOT</version>
```



pom.xml

```
<!--
```

Le type de packaging, c'est-à-dire le type de projet.  
Ici, on indique à Maven que le projet doit être packagé  
sous la forme d'un JAR.

```
-->
```

```
<packaging>jar</packaging>
```

pom.xml

```
<!--
```

Les propriétés de notre projet. On peut définir des propriétés spécifiques au projet ou des propriétés standard à Maven pour paramétrer la construction du projet.

```
-->
```

```
<properties>
```

```
...
```

```
</properties>
```

## pom.xml

```
<properties>
```

```
  <!-- Propriété standard définissant la version minimale de Java  
supportée par les fichiers sources (ici 1.8 pour Java 8).-->
```

```
  <maven.compiler.source>1.8</maven.compiler.source>
```

```
  <!-- Propriété standard définissant la version Java des fichiers compilés  
du projet (ici 1.8 pour Java 8).-->
```

```
  <maven.compiler.target>1.8</maven.compiler.target>
```

```
  <!-- Le format d'encodage des fichiers source du projet. Attention,  
l'encodage par défaut n'est pas le même sous Windows et sous les  
systèmes *NIX. Il est donc plus sage de toujours positionner cette  
propriété dans le fichier pom.xml. -->
```

```
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
```

```
</properties>
```

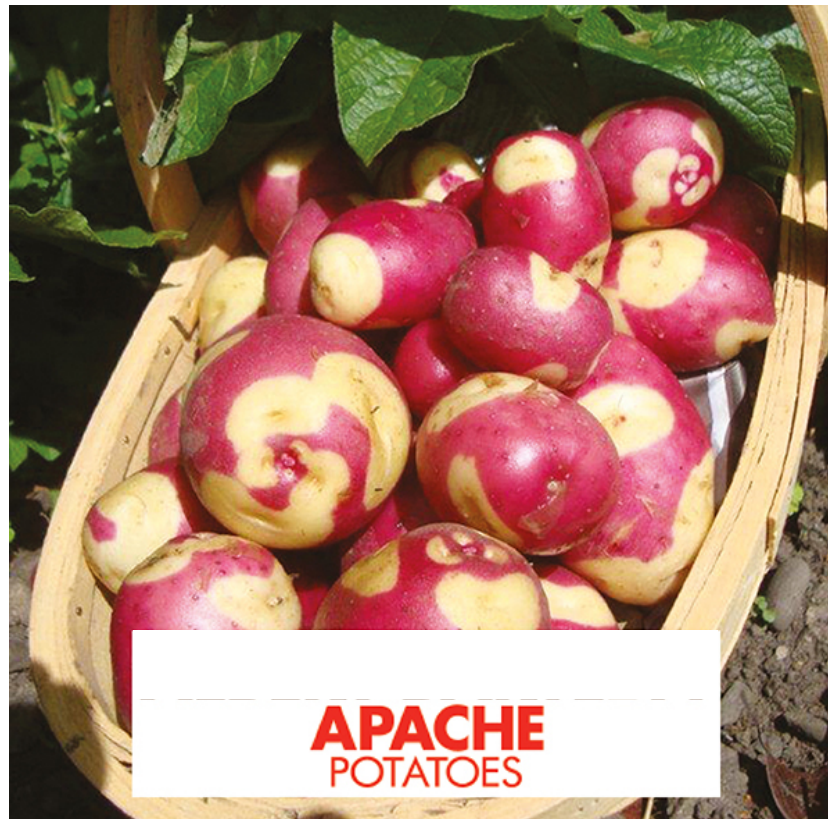


# Références

- <http://spoonless.github.io/>
- Pour l'installation
- [Maven Installation on mac](#)
- Pour avoir une référence complete:
- <http://books.sonatype.com/mvnref-book/pdf/mvnref-pdf.pdf>
- Pour le demo:
- [How to create a Java project with Maven in Netbeans IDE](#)

Sorry for the long presentation

Here's an APACHE potato



# Questions

