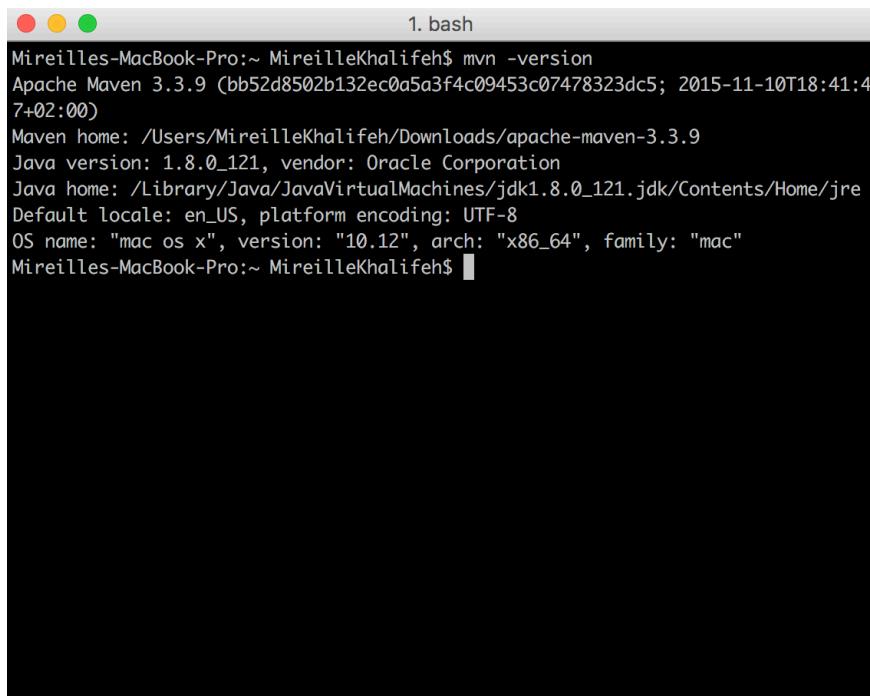


Maven demo



Installation:

Ce demo utilise le maven 3.3.9, et le java version: 1.8.0_121 et avec le NetBeans IDE 8.2, pour voir la version du maven et du java il suffit d'écrire la commande mvn -version:



A screenshot of a Mac OS X terminal window titled "1. bash". The window shows the command "mvn -version" being run and its output. The output includes the Maven version (3.3.9), Java home (User's Downloads folder), Java version (1.8.0_121), Java home (Library/Java/VirtualMachines folder), Default locale (en_US), platform encoding (UTF-8), and OS name (mac os x).

```
Mireilles-MacBook-Pro:~ MireilleKhalifeh$ mvn -version
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T18:41:4
7+02:00)
Maven home: /Users/MireilleKhalifeh/Downloads/apache-maven-3.3.9
Java version: 1.8.0_121, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.12", arch: "x86_64", family: "mac"
Mireilles-MacBook-Pro:~ MireilleKhalifeh$
```

Si le Java JDK n'est pas installé il faut l'installer, et Si maven n'est pas installé suivez les instructions suivantes sur le Mac ou ubuntu (j'utilise l'application iTerm au lieu du Terminal car il facilite le travail):

```
1. bash
Mireilles-MacBook-Pro:~ MireilleKhalifeh$ brew install maven
Updating Homebrew...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
awslogs          grunt          llvm@3.9        woboq_codebrowser
biogeme         guetzli        magic-wormhole  xsv
bluepill        jenkins-lts   streamlink      zxing-cpp
crowdin         juju-wait     teleport
ctop            latexila       tfenv
fzy             libcds         urh
==> Updated Formulae
abcde           libsquish
ace             libstfl
ack             libtensorflow
acpica          libvoikko
activemq        libwebsockets
admesh          libwps
alpine          llvm
android-ndk    logtalk
android-platform-tools lxc
ansible         lynis
apache-drill   lzo
apache-zeppelin macvim
app-engine-java mariadb
```

Puis on ouvre bash_profile on utilisant la commande vim ~/.bash_profile et on ajoute 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
```

Dans le premier export on remplace ce path par la location du maven sur notre ordinateur

```
1. vim
export JAVA_HOME=$(/usr/libexec/java_home)
export ANDROID_HOME=/Users/MireilleKhalifeh/Library/Android-SDK

export PATH=$JAVA_HOME/jre/bin:$PATH
export PATH=$PATH:$ANDROID_HOME/tools
export PATH=$PATH:$ANDROID_HOME/platform-tools
export M2_HOME=/Users/MireilleKhalifeh/Downloads/apache-maven-3.3.9
export M2=$M2_HOME/bin
export PATH=$PATH:$M2s
~
~
```

on écrit :wq pour quitter et sauvegarder le bash_profile.

Puis \$source ~/.bash_profile.

Si on écrit mvn -version on doit avoir tout les details.

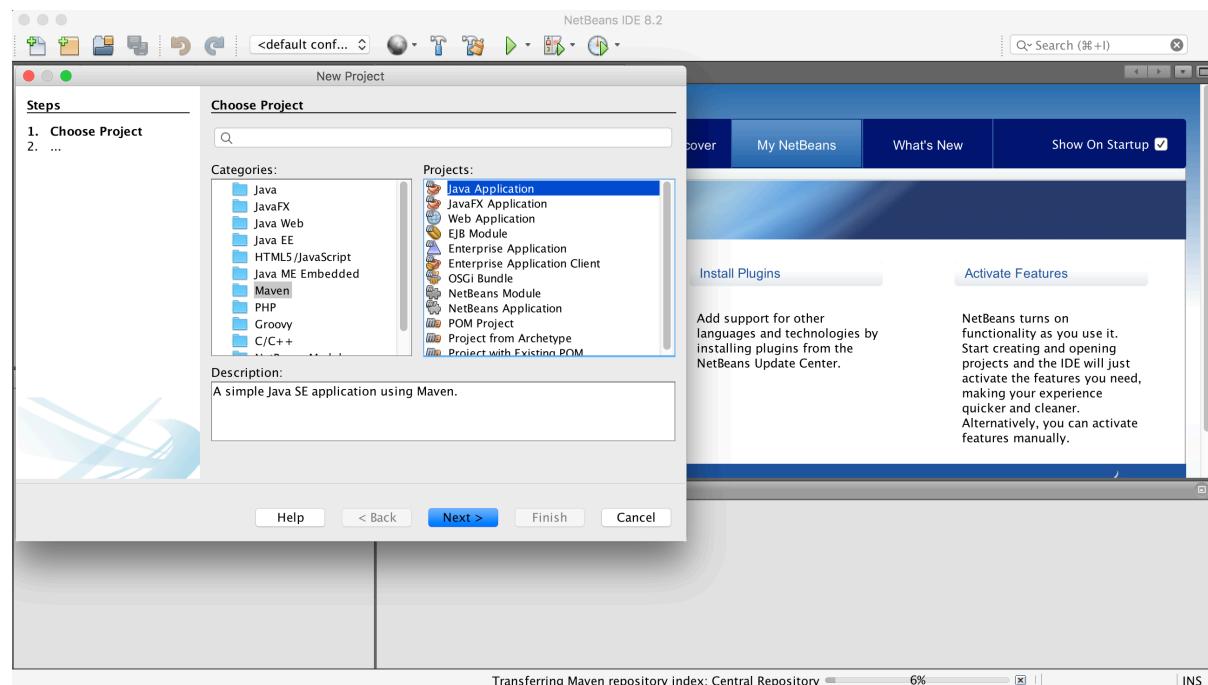
Pour l'installation sur windows et ubuntu voir ces liens:

<http://www.objis.com/formation-java/tutoriel-maven-installation-phases.html>.

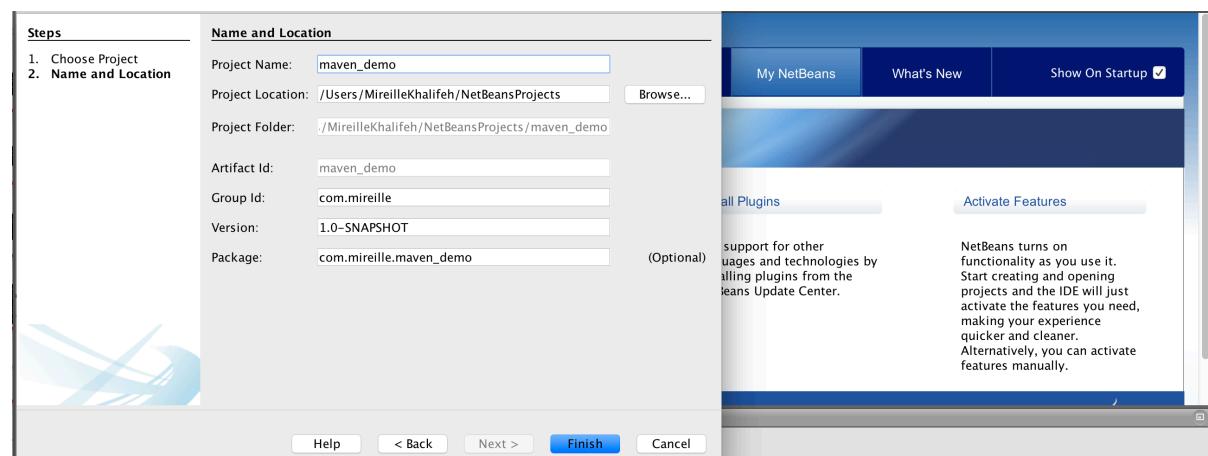
<https://doc.ubuntu-fr.org/maven>

Projet Maven:

Commencer par créer un nouveau projet, j'ai choisi un simple application java pour cette démonstration:

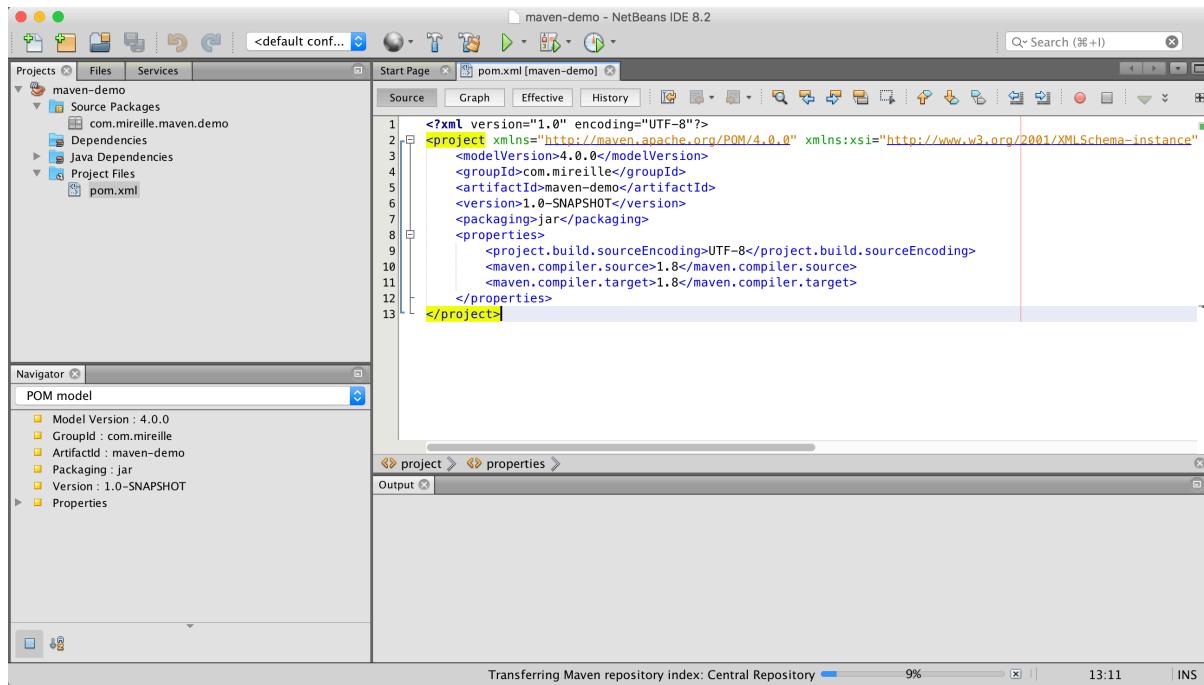


Choisissez le nom du projet qui sera le même l'artifact id et le groupe id.



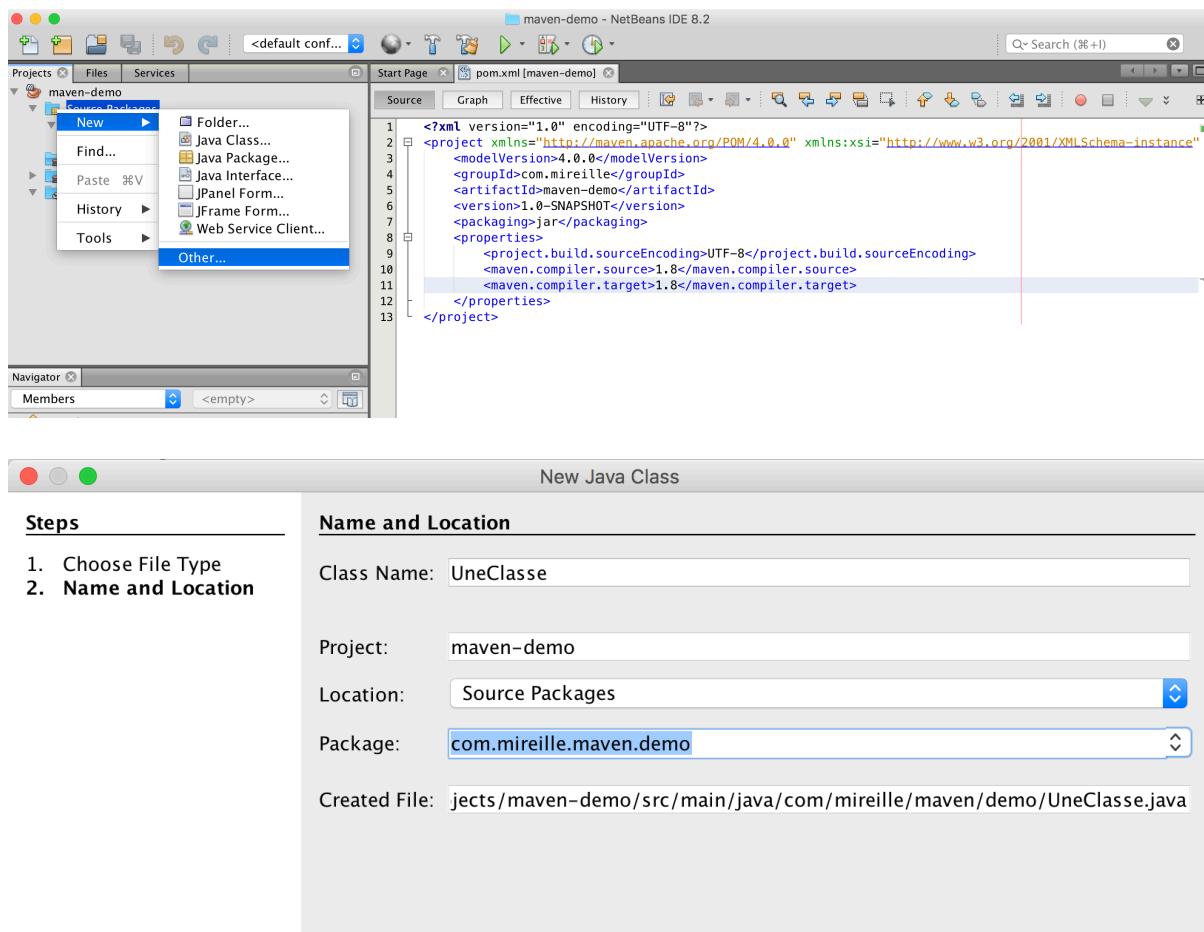
La version SNAPSHOT indique que ce n'est pas la version finale du projet.

Le projet étant créé, on ouvre pom.xml et voilà ce qu'on trouve:

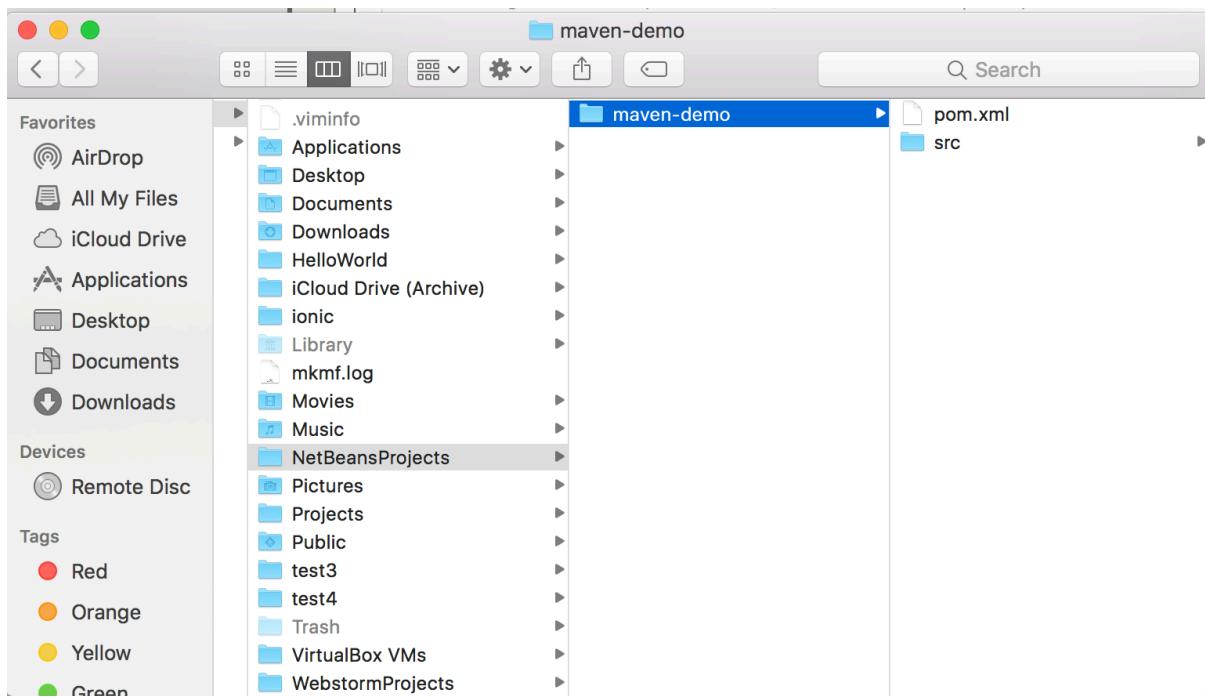


```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://www.w3.org/2001/XMLSchema-instance"
  modelVersion="4.0.0">
  <groupId>com.mireille</groupId>
  <artifactId>maven-demo</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
</project>
```

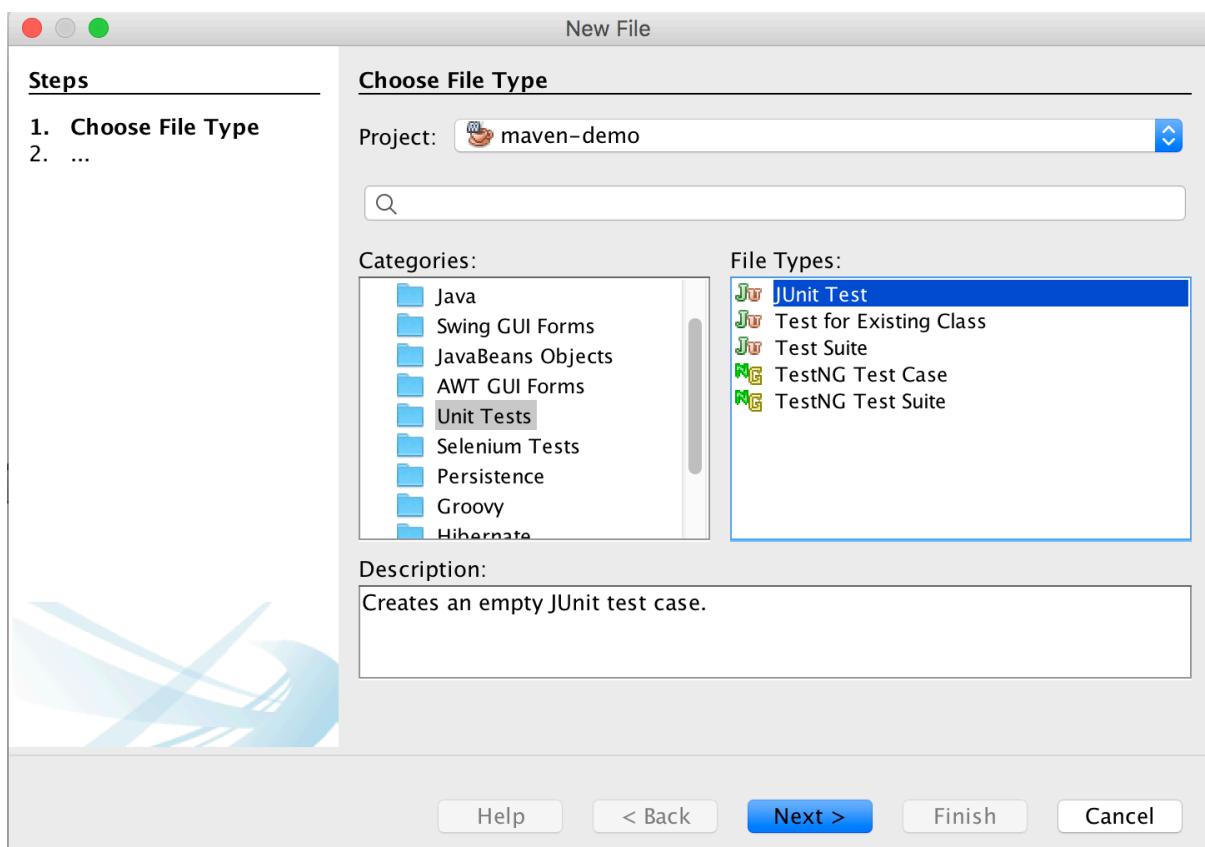
On ajoute une classe java:

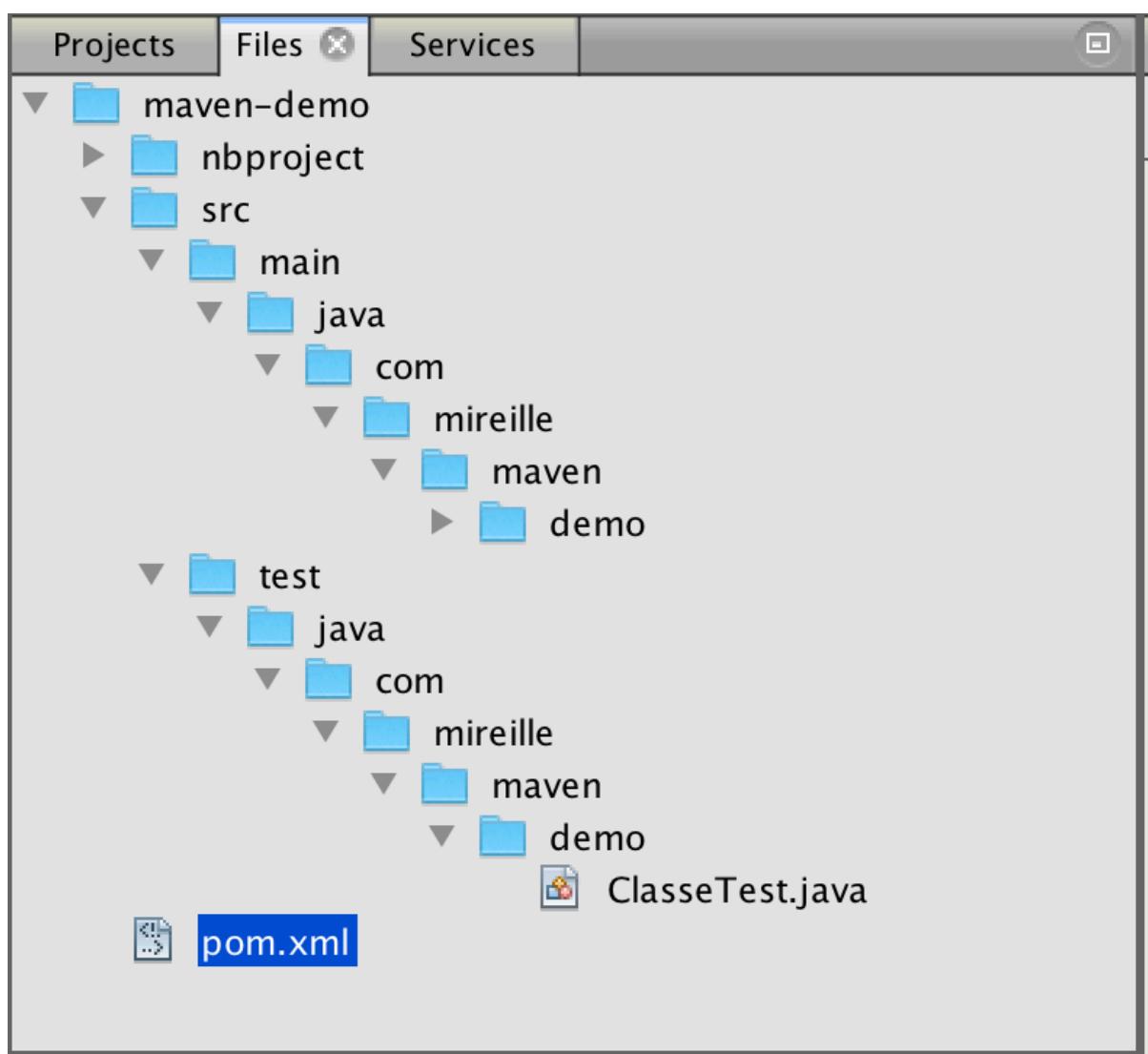
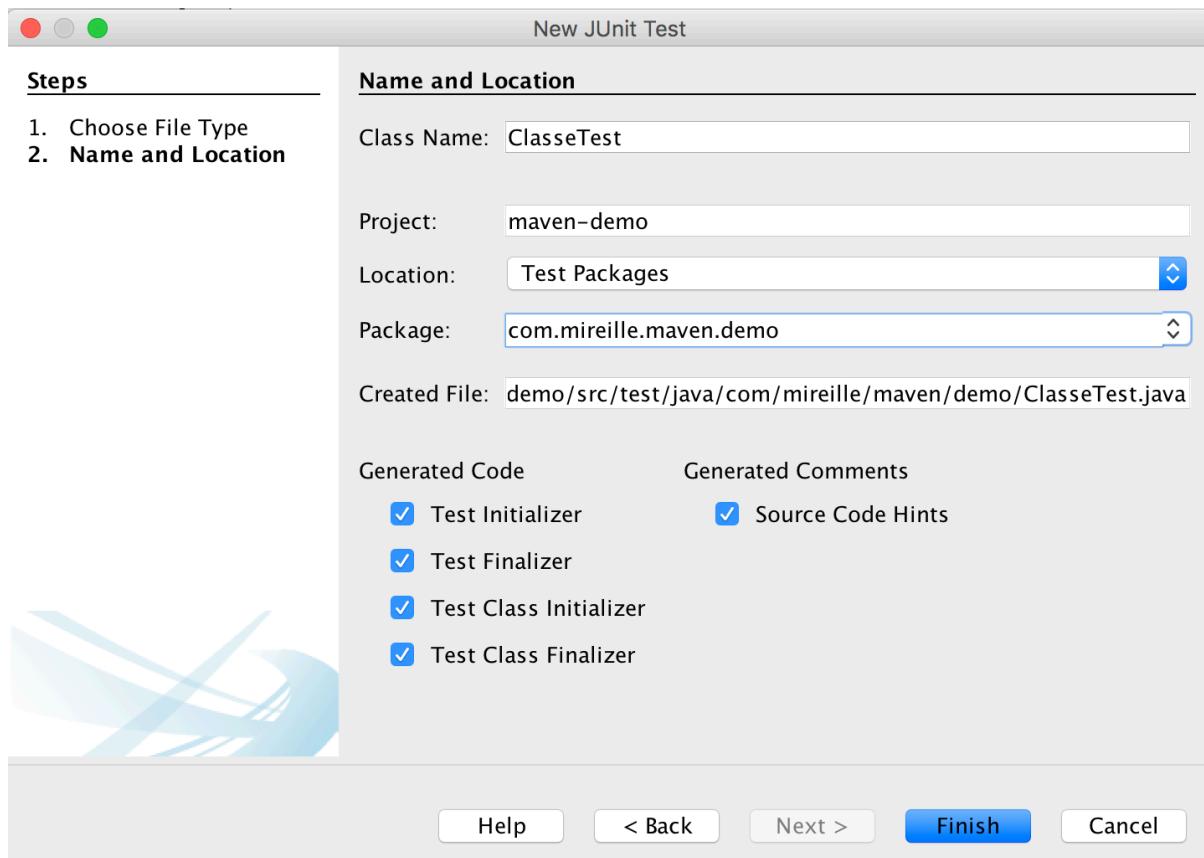


On peut voir dans le finder la structure de notre projet, on a le pom et un folder src.



Puis on ajoute un test, le jUnit test:





On remarque que dans notre projet on a maintenant nbprojet, src, test et pom.xml. Si on ouvre le pom.xml on trouve qu'il y a des dependencies, ce sont le junit et org.hamcrest le dernier est automatiquement ajouté.

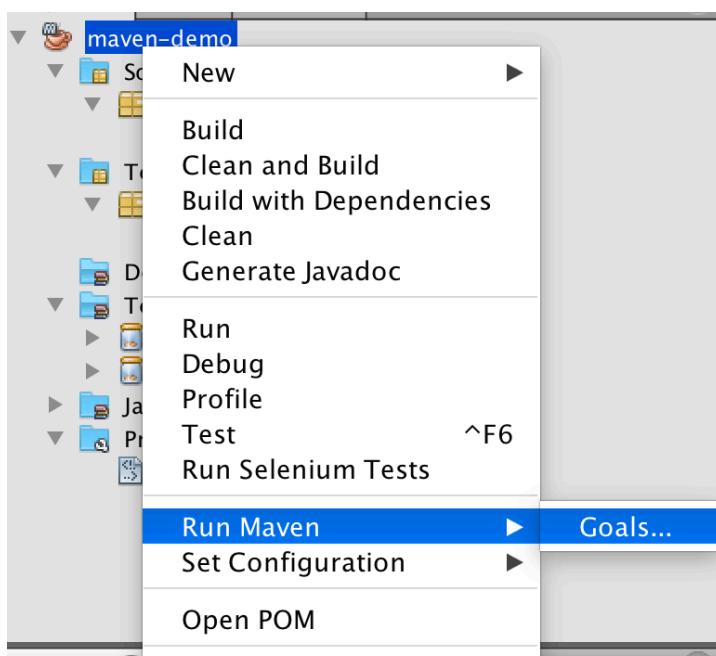
```

<?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">
  <groupId>com.mireille</groupId>
  <artifactId>maven-demo</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.hamcrest</groupId>
      <artifactId>hamcrest-core</artifactId>
      <version>1.3</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
</project>

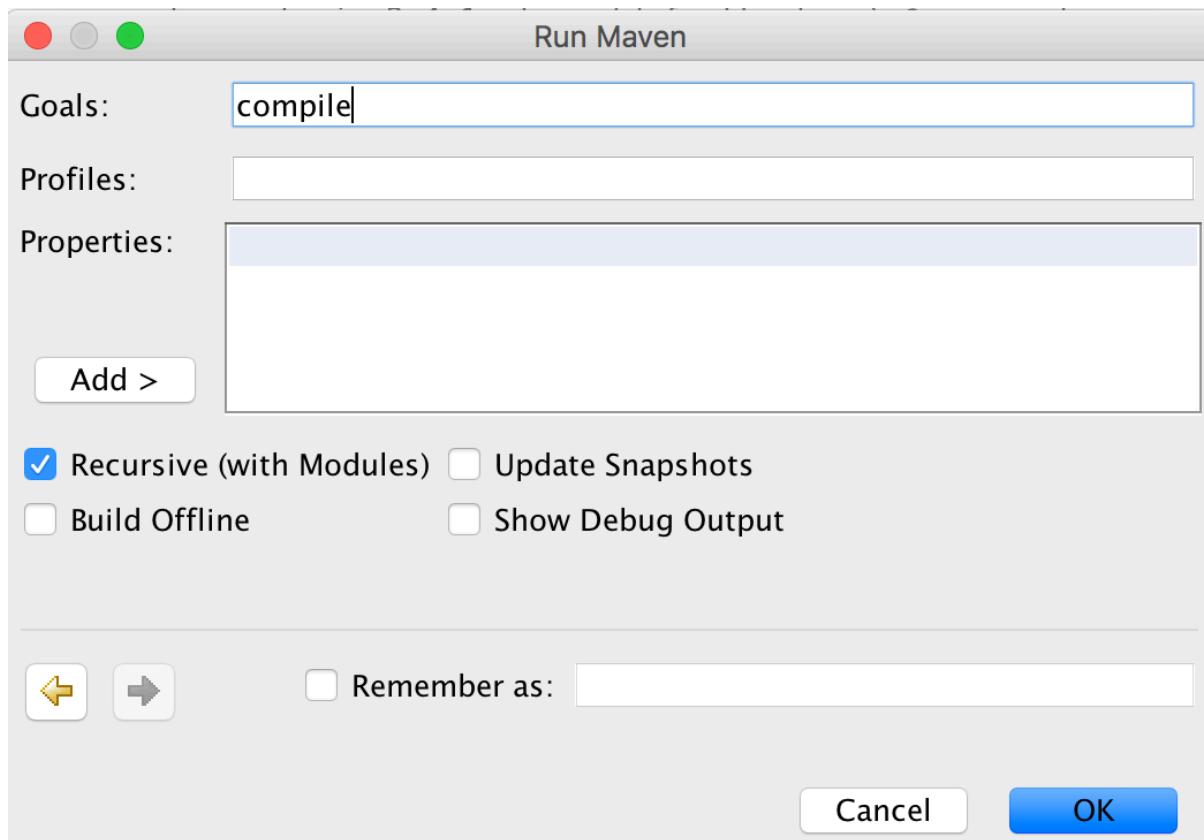
```

Si je supprime les dependencies et je sauvegarde mon projet, on remarque que le “Test Dependencies” est vide.

Pour compiler, executer des tests et créer des jars, Right Click sur mon projet -> Run Maven -> Goals. (ca peut être costume au lieu de Run Maven)



Compile:



```
Output - Build (maven-demo) ×
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory /Users/MireilleKhalifeh/NetBeansProjects/maven-demo/src/test/resource
--- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ maven-demo ---
Compiling 1 source file to /Users/MireilleKhalifeh/NetBeansProjects/maven-demo/target/test-classes

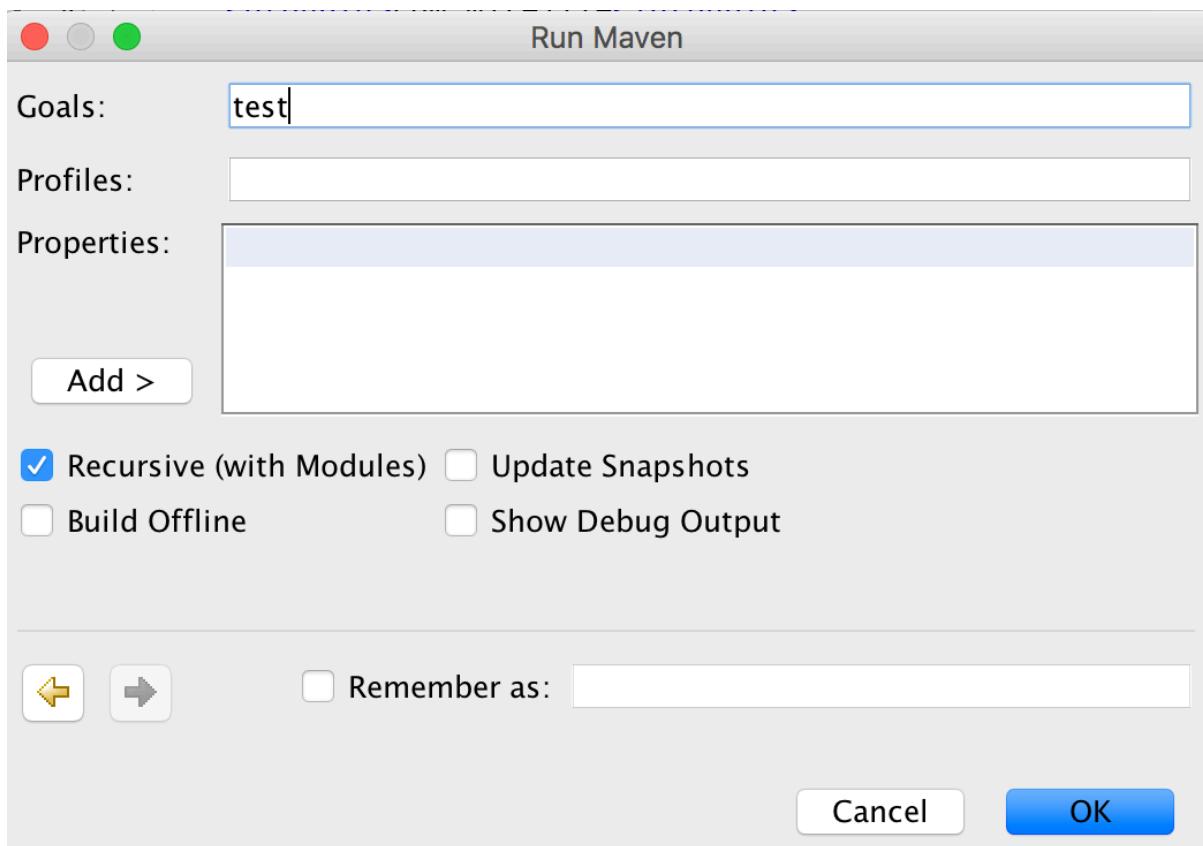
--- maven-surefire-plugin:2.10:test (default-test) @ maven-demo ---
Surefire report directory: /Users/MireilleKhalifeh/NetBeansProjects/maven-demo/target/surefire-reports
Downloading: http://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.10/surefire-
Downloaded: http://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.10/surefire-
Downloaded: http://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.10/surefire-
Downloaded: http://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.10/surefire-
-----
T E S T S
-----
Results :

Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

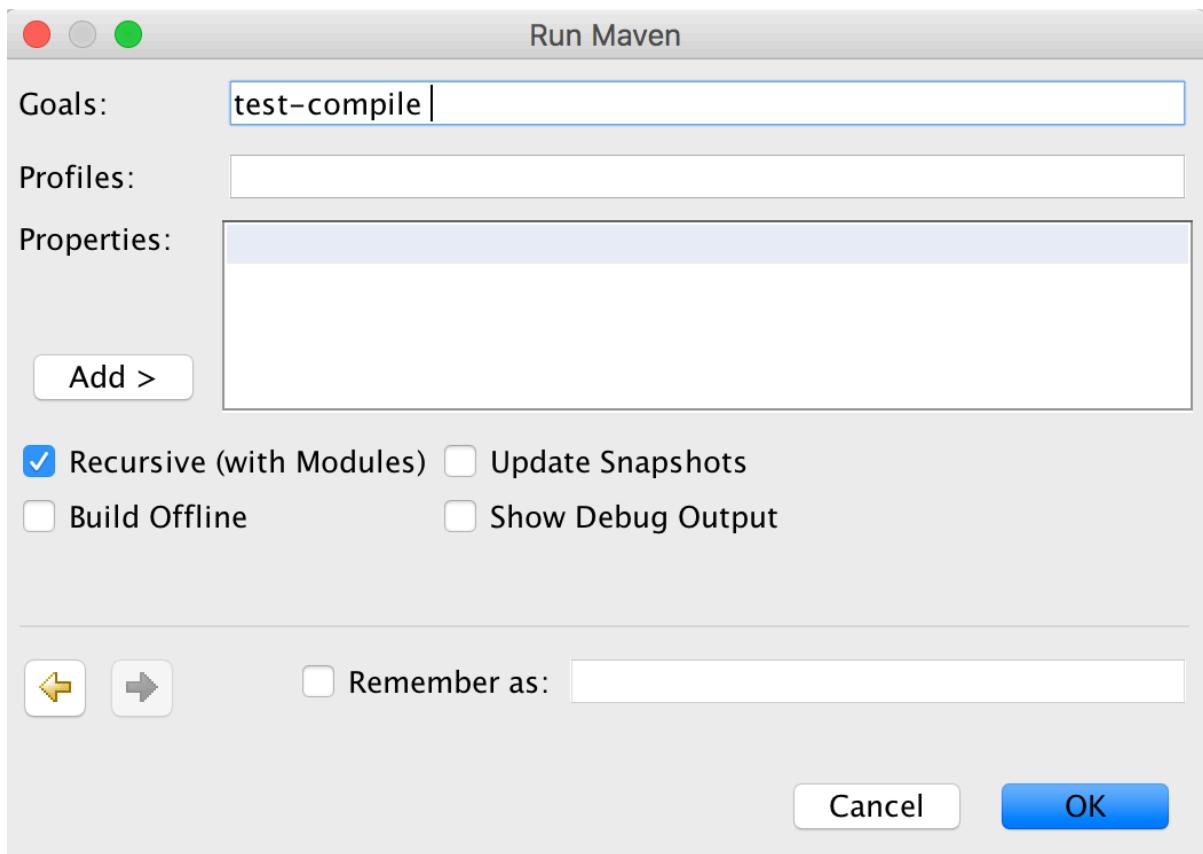
-----
B U I L D   S U C C E S S
-----
Total time: 8.830s
Finished at: Mon Mar 27 08:37:37 EEST 2017
Final Memory: 17M/309M
```

Transferring Maven repository index: Central Repository 15% 6:21/1:1 | INS

Executer un test:



Compiler un test:



```
Output - Build (maven-demo) x
cd /Users/MireilleKhalifeh/NetBeansProjects/maven-demo; JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.
Scanning for projects...

-----[BUILDING]-----
Building maven-demo 1.0-SNAPSHOT

-----[MAVEN RESOURCES]-----
[maven-resources-plugin:2.5:resources (default-resources) @ maven-demo]
[debug] execute contextualize
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory /Users/MireilleKhalifeh/NetBeansProjects/maven-demo/src/main/resource

-----[MAVEN COMPILER]-----
[maven-compiler-plugin:2.3.2:compile (default-compile) @ maven-demo]
Nothing to compile - all classes are up to date

-----[MAVEN RESOURCES TEST]-----
[maven-resources-plugin:2.5:testResources (default-testResources) @ maven-demo]
[debug] execute contextualize
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory /Users/MireilleKhalifeh/NetBeansProjects/maven-demo/src/test/resource

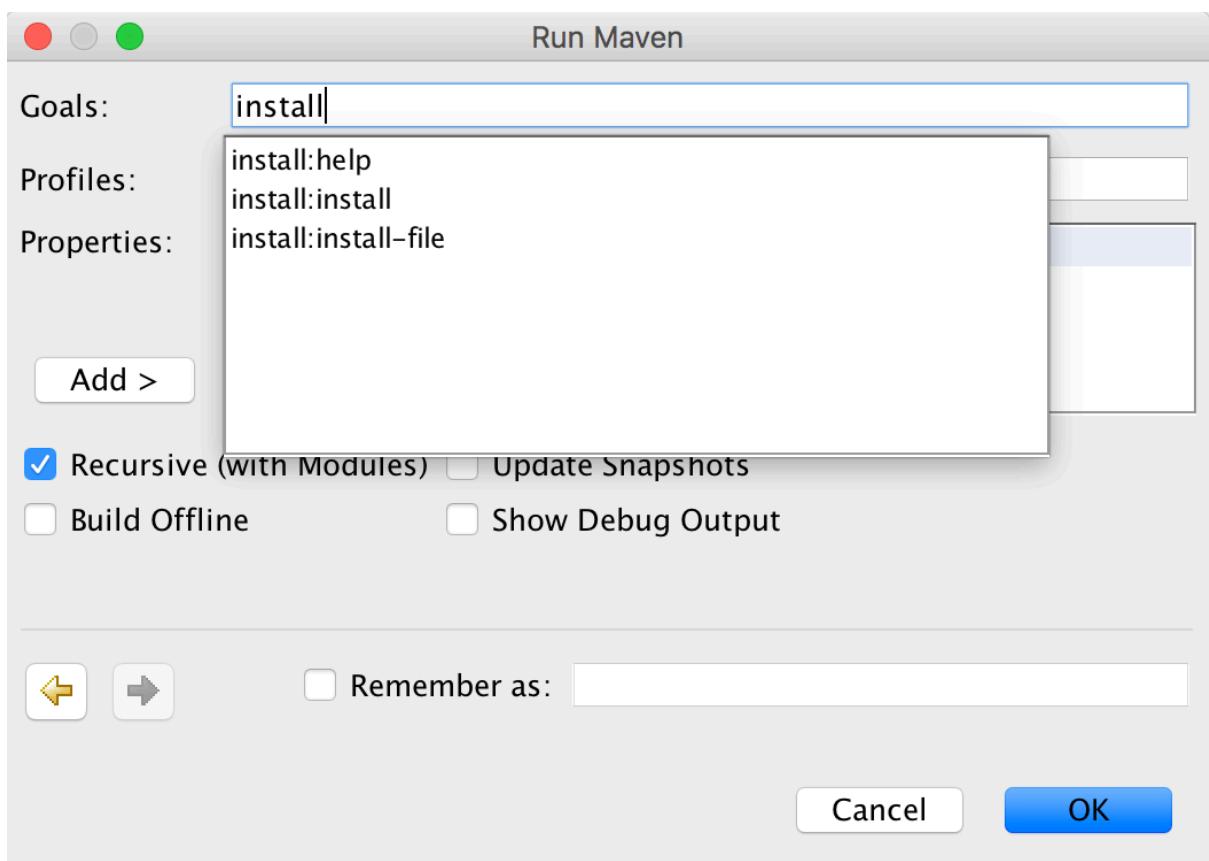
-----[MAVEN COMPILER TEST]-----
[maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ maven-demo]
Nothing to compile - all classes are up to date

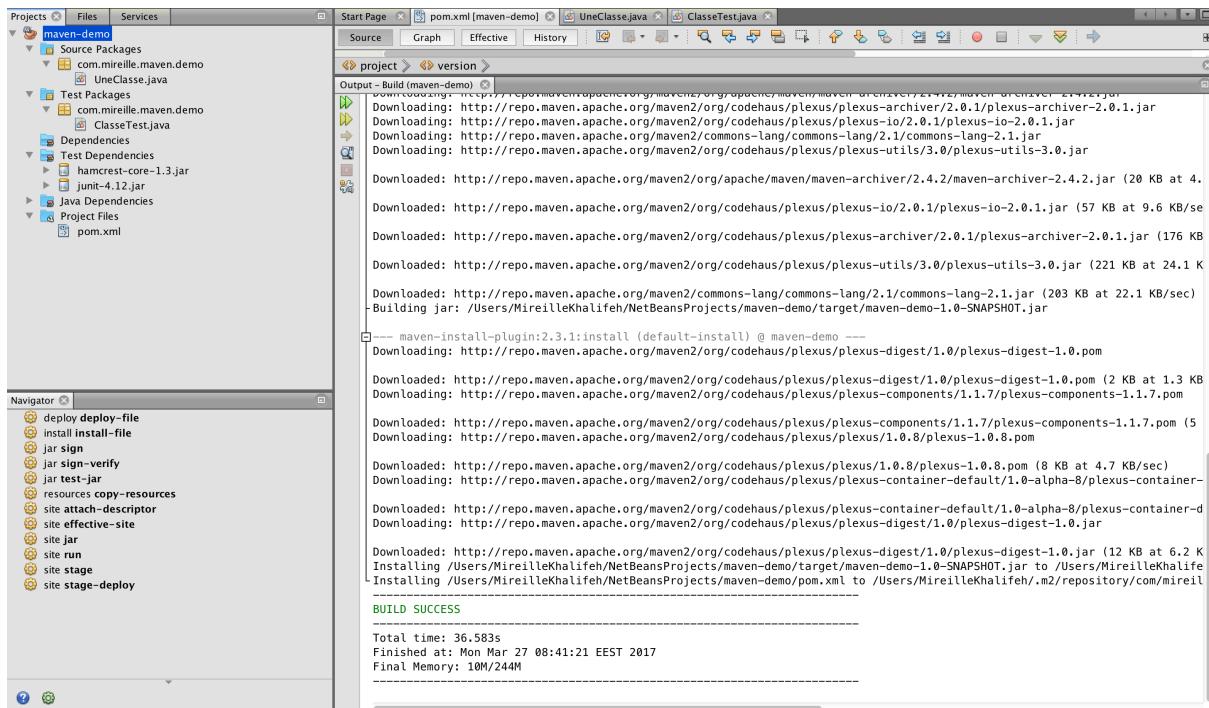
-----[BUILD SUCCESS]-----
BUTLD SUCCESS

Total time: 0.522s
Finished at: Mon Mar 27 08:39:26 EEST 2017
Final Memory: 6M/245M

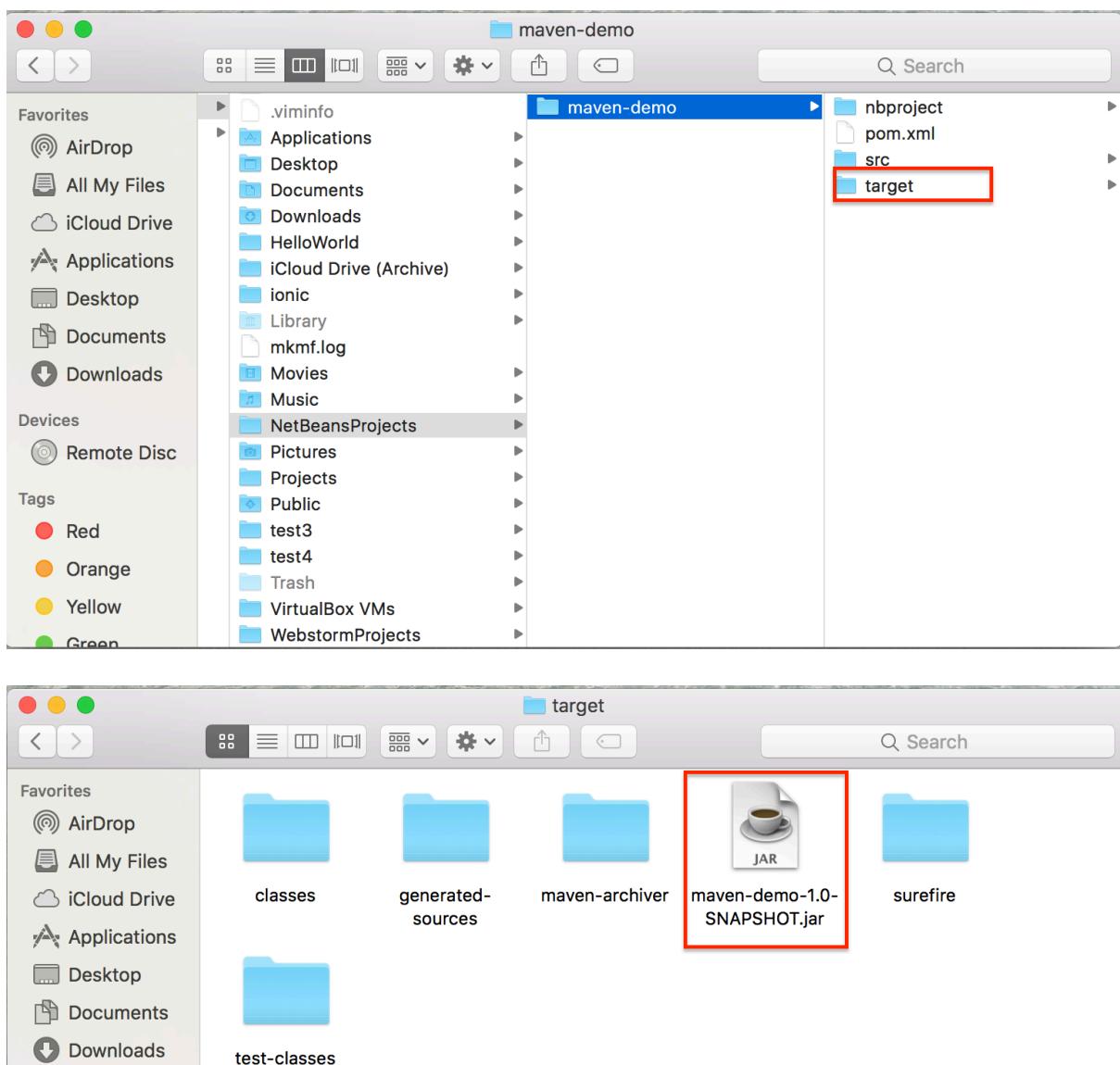
-----[TRANSPORT]-----
Transferring Maven repository index: Central Repository [Progress Bar] 20% | 1 | 6:21/1:1 | INS
```

Pour créer un jar :





Le jar est maintenant créé, on remarque que le répertoire target est maintenant ajouté:



Pour supprimer ce jar on peut utiliser la commande clean, elle supprime tout le target.

