

Université Mohammed Premier Oujda

Ecole National des Sciences Appliquées

Année Universitaire 2019-2020 TP : Maven2

Un outil de build et gestionnaire de projets JAVA

I. Partie 1 : Installation de l'outil Maven2 :

a. Vérifier votre installation de Java

Dans une console de commande exécuter la commande `java -version` pour s'assurer que votre poste contient une version JDK 1.5 ou supérieur :

```
C:\Users\Administrateur>java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
```

b. Téléchargement de Maven

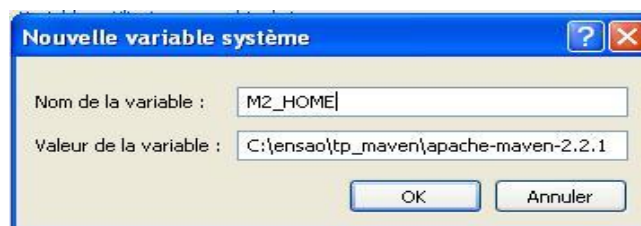
Télécharger Maven depuis le site internet du projet Apache Maven avec l'url suivante <http://maven.apache.org/download.html> . Lorsque vous téléchargez Maven, faites attention à choisir la version stable de Apache Maven disponible sur le site de Maven (version **2.2.1** : **apache-maven-2.2.1-bin.zip**).

c. Installation de Maven

Choisissez l'emplacement où vous voulez installer Maven (utiliser par exemple : **c:\ensao\tp_maven** pour suivre les étapes du TP) puis décompressez l'archive **apache-maven-2.2.1-bin.zip** à cet endroit.

Vérifier que le répertoire « **C:\ensao\tp_maven\apache-maven-2.2.1** » contient les dossiers « bin », « boot », « conf », ...

Créer la variable d'environnement **M2_HOME** qui pointe sur le chemin « **C:\ensao\tp_maven\apache-maven-2.2.1** » :



Mettre à jour la variable d'environnement **PATH** en ajoutant le chemin « **;%M2_HOME%\bin;** » à la fin de la liste des chemins déjà renseignés :



d. Vérification de l'installation Maven

Une fois Maven installé, vous pouvez vérifier sa version en exécutant la commande suivante **mvn -v** en ligne de commande. Si Maven a été correctement installé, vous devriez voir en sortie quelque chose ressemblant à cela :

```
C:\Users\Administrateur>mvn -v
Apache Maven 3.6.0 (97c98ec64a1fdfee7767ce5fffb20918da4f719f3; 2018-10-24T19:41:47+01:00)
Maven home: C:\apache-maven-3.6.0\bin\..
Java version: 1.8.0_171, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_171\jre
Default locale: fr_FR, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

Exécuter la commande **mvn -h** pour afficher le help/la liste des commandes Maven.

II. Partie 2 : Création d'un projet Maven « Simple » :

a. Création du projet :

Créer un répertoire de travail dans votre poste (utiliser par exemple : **c:\projet1Maven** pour suivre les étapes du TP) ;

Via une console de commande se positionner sur ce répertoire via la commande « **cd c:\projet1Maven** » ;

Exécuter la commande « **mvn archetype:generate -**

DgroupId=org.sonatype.mavenbook.simple -DartifactId=simple -

DpackageName=org.sonatype.mavenbook -Dversion=1.0-SNAPSHOT » :

1. Commentez les arguments de la ligne de cette commande. A quoi servent ces arguments ?
2. Où se trouve-t-il le projet généré par cette commande ?
3. Parcourez le dossier qui a été créé. Que contient-il ?
4. Qu'est-ce que le fichier **pom.xml** ?
5. Où est la classe contenant la fonction **main** ?
6. Consulter le fichier **c:\projet1Maven\simple\pom.xml** puis décrire son contenu.

b. Construction du projet « Simple »:

Une fois le projet créé grâce au plugin Maven Archetype suivant les instructions de la section précédente, il faut construire et packager l'application. Afin d'y parvenir :

1. Compilez le projet avec la commande **mvn compile**.
2. A quoi sert le dossier **test** ? Lancez les tests avec la commande **mvn test**.
3. Créez un JAR avec la commande **mvn package**. Où se trouve-t-il ?
4. Re-Parcourir l'arborescence du **répertoire c:\projet1Maven\simple** puis expliquez le rôle de son contenu (répertoires et fichiers). Qu'observez-vous ?
5. Vous venez juste de créer, compiler, tester, packager et installer le projet Maven le plus simple possible. Pour vous prouver que ce programme fonctionne, exécutez-le depuis la ligne de commande : **java -cp target/simple-1.0-SNAPSHOT.jar org.sonatype.mavenbook.simple.App** ou via Maven: **mvn exec:java -Dexec.mainClass="org.sonatype.mavenbook.simple.App"**
6. Consulter la classe **App.java** puis remplacer la chaîne « **Hello World!** » par « **TP Maven : Bonjour tout le monde** », sauvegarder le puis exécuter la commande **mvn install**
7. Re-exécuter le projet. Que remarquez-vous ?
8. Consulter la classe de test générée par Maven puis :
Ajouter la ligne **assertTrue(false)** ; dans la méthode de test ;
Relancer le build du projet (**mvn clean install**)
Que remarquez-vous ?
9. Relancer le build du projet sans les tests unitaires via (**mvn clean install -Dmaven.test.skip=true**) ou (**mvn clean install -DskipTests=true**)

III. Partie 2 : Création d'un projet Maven pour eclipse :

a. Installation du plugin Maven dans Eclipse:

Si Maven n'est pas installé dans votre eclipse, revendez dans la page <http://eclipse.org/m2e/download/> pour l'installer.

Générez un projet Eclipse avec la commande : **mvn eclipse:eclipse -Dwtpversion=2.0**

Ouvrez Eclipse et importez le projet généré (via importer un projet existant dans l'espace de travail).

Consulter la classe de test générée par Maven puis:

Remplacer la ligne `assertTrue(false)` ; par `assertTrue(true)` ; dans la méthode de test ;
Relancer le build du projet (run as clean puis run as install)
Exécuter la classe Main du projet
ReConsulter la classe de test générée par Maven puis :
Remplacer la ligne `assertTrue(true)` ; par `assertTrue(false)` ; dans la méthode de test ;
Relancer le build du projet (run as clean puis run as install)
Que remarquez-vous ?
Relancer le build du projet sans les tests unitaires