

# TP1 – Utilisation de Maven

## Exercice 1 : Maven de A à Z

### Partie1 : Installation Maven

1. Téléchargez et dézippez l'archive courante de MAVEN, par exemple apache-maven-3.6.3-bin.zip.
2. Analysez le contenu du répertoire bin.  
QUESTION : à quoi sert le fichier m2.conf ?
3. Analysez le contenu du répertoire conf.  
QUESTION : à quoi sert la balise 'localRepository' présente dans le fichier settings.xml
4. Ajouter la variable d'environnement utilisateur M3\_HOME avec pour valeur le chemin du répertoire apache-maven-3.2.2
5. Ajouter la variable d'environnement M3 avec comme valeur %M3\_HOME%bin
6. Créez ou mettez à jour la variable d'environnement PATH avec %M3%
7. Assurez-vous que la variable d'environnement JAVA\_HOME pointe bien sur le répertoire d'installation du JDK, et que %JAVA\_HOME%bin est bien dans la variable PATH.
8. Lancez une nouvelle commande MS-DOS et exécutez : mvn -version pour avoir la version de Maven installées.

### Partie2 : la notion de 'Phase' MAVEN

Maven associe à tout projet MAVEN des phases. A chacune de ces phases est associé/déclenché par défaut des tâches mises en œuvre par des plugins.

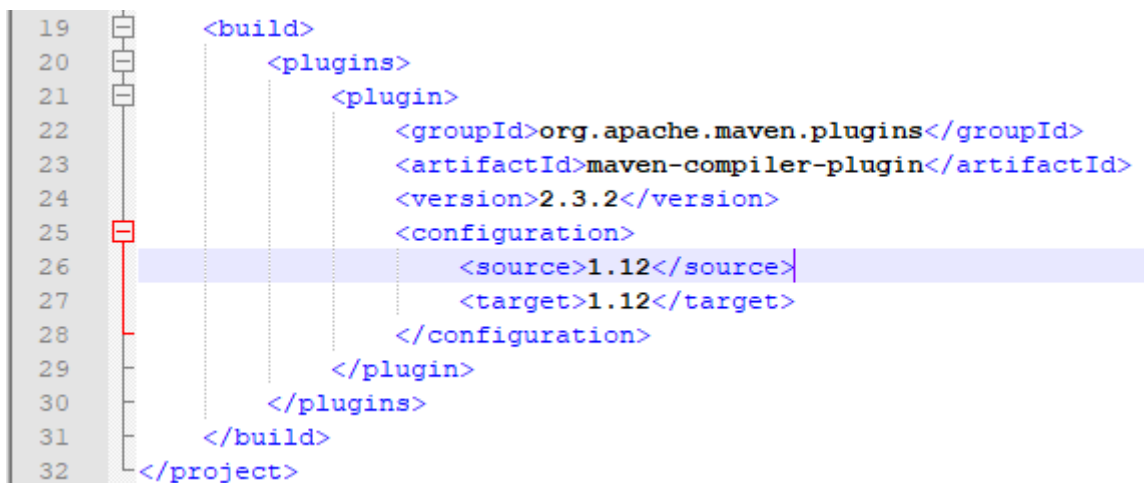
1. Lancez une commande maven 'vide', c'est à dire tapez uniquement 'mvn' en ligne de commande. Maven signale une erreur et précise la liste des phases disponible. Quelle est la liste des phases de maven 3 ?
2. Créez le répertoire : C:\coursmaven\TPs\TP1
3. Dans le répertoire créé, exécutez la commande suivante :

```
mvn archetype:generate -B -DarchetypeArtifactId=maven-archetype-quickstart -DgroupId=fr.efrei.mavenapps -DartifactId=MyApp -Dversion=1.0-SNAPSHOT -DpackageName=fr.efrei.mavenapps
```

- Que se passe-t-il ?

QUESTION : quels types de fichiers sont téléchargés ? et où sont t'ils déposés ?

4. Analysez le répertoire .m2 créé par maven dans votre répertoire utilisateurs système
5. Analysez le contenu du sous-répertoire mavenapps
6. Editez le fichier App.java (par exemple avec éditeur Notepad++ ou Visual Studio Code) et observez le.
7. De même, éditez et observez le fichier src/test/java/TestApp.java
8. Analysez le contenu du fichier pom.xml
  - A quoi sert la balise 'scope' de la dépendance Junit ?
  - Citez d'autres balises que l'on peut retrouver dans ce fichier pom.xml
9. A partir du répertoire 'myApp', lancez la commande : **mvn compile**
  - Que se passe-t-il ?
10. Télécharger puis ajouter la section suivante pour configurer le compilateur avec la bonne version



```
19 <build>
20   <plugins>
21     <plugin>
22       <groupId>org.apache.maven.plugins</groupId>
23       <artifactId>maven-compiler-plugin</artifactId>
24       <version>2.3.2</version>
25       <configuration>
26         <source>1.12</source>
27         <target>1.12</target>
28       </configuration>
29     </plugin>
30   </plugins>
31 </build>
32 </project>
```

11. Relancez la commande mvn compile.
12. Analysez le contenu du répertoire myApp/target généré
13. Lancez la commande : **mvn test**
  - Que se passe-t-il ?
  - Analysez le contenu du répertoire myApp/target généré
  - Analysez le contenu du répertoire myApp/target/surefire-reports généré
14. Lancez la commande : **mvn package**

- Que se passe-t-il ?
- Analysez le contenu du répertoire myApp/target généré

15. Quelle relation avec mvn compile ?

16. Lancez la commande :

```
java -cp target/myApp-1.0-SNAPSHOT.jar fr.efrei.mavenapps.App
```

- Que se passe-t-il ?

17. Lancez la commande : **mvn install**

- Que se passe-t-il ?

18. Qu'est-ce qu'un repository local ?

19. Quel est le repository local par défaut ? Comment le changer ?

- Montrez que le jar de l'application est désormais dans le repository local. A quel endroit ?
- Confirmez la création du jar myApp-1.0-SNAPSHOT.jar dans le répertoire %USER\_HOME%/.m2/repository/fr/efrei/demomaven

20. Lancez la commande : **mvn clean**

- Que se passe-t-il ?

21. Lancez la commande : **mvn clean install**

- Que se passe-t-il ?

Pour voir le site du projet, déclarer le plugin maven-site-plugin dans la section 'build' , puis préciser les types de rapports souhaités

### Partie 3 : Génération site web projet

1. Ajouter la déclaration suivante (correspondant au code suivant) dans le pom.xml, juste au-dessus de la section 'dependencies' :

```

17 <!-- Plugin de génération du site du projet -->
18 <build>
19 <plugins>
20 <plugin>
21 <groupId>org.apache.maven.plugins</groupId>
22 <artifactId>maven-site-plugin</artifactId>
23 <version>3.0-beta-3</version>
24 <configuration>
25 <reportPlugins> <!-- Ici 1 seul rapport , celui des informations standards du POM-->
26 <plugin>
27 <groupId>org.apache.maven.plugins</groupId>
28 <artifactId>maven-project-info-reports-plugin</artifactId>
29 <version>2.0.1</version>
30 </plugin>
31 </reportPlugins>
32 </configuration>
33 </plugin>
34 </plugins>
35 </build>

```

2. Lancez la commande ‘mvn site’

- Que se passe-t-il ?

3. Supprimer la ligne concernant la version puis relancez la commande : mvn site.

- Constatez la création du répertoire ‘target/site’
- Ouvrir le fichier index.html de ce répertoire avec votre navigateur
- Cliquez sur le lien ‘Dependencies’

4. Quel lien avec notre fichier pom.xml ?

5. Expliquez les autres liens du menu (gauche) du site :

- about
- Continuous integration
- Issue tracking
- Mailing list
- Plugin Management
- Project License
- Project Summary
- Project Team
- Source Repository

QUESTION : comment faire pour que ce rapport soit en français ?

## Partie 4 : Personnalisation du site web projet

1. Créez un répertoire src/site

2. Dans ce répertoire, créez un fichier site.xml

- Analysez le contenu du fichier
3. Lancez mvn site à nouveau
    - Analysez le contenu de targetsite
    - Ouvrez le fichier index.html
  4. Expliquez le lien entre cette interface et le fichier site.xml
  5. Ajoutez une nouvelle section au menu de gauche.

**Exercice 2 :**

1. Création d'un projet Java sans Maven en ligne de commande
2. Création d'un projet Java avec Maven en ligne de commande
3. Création d'un projet Java Maven avec un IDE

Pour chaque cas cité ci-dessus :

- Créez une classe Calcul qui contient deux méthodes Somme qui permet de retourner la somme de deux nombres double et Produit qui permet de retourner le produit de deux nombres double.
- Créez une classe pour le test unitaire.
- Compilez toutes les classes.
- Exécutez les tests unitaires
- Installez le jar du projet dans le repository local de Maven.

