

# TP5 : Maven

---

## Objectif du TP

L'objectif de ce TP est de découvrir l'outil Maven pour le développement de projets logiciels (dans l'environnement Eclipse ou non).

## SOUS WINDOWS (SANS ECLIPSE)

1. Télécharger Maven [ici](#) et dézipper le fichier obtenu sous `c:\` par exemple.
2. Analysez le contenu des répertoires de Maven et dites à quoi servent le fichier `M2.conf` et la balise `localRepository` dans le fichier `settings.xml` ?
3. Ajouter la variable d'environnement système `M2_HOME` avec pour valeur le chemin du répertoire d'accès à Maven. Faites de même avec la variable `M2` qui fait référence au répertoire `bin` de Maven. Modifier la variable système `PATH` en ajoutant `;%M2%` à la fin.
4. Lancez une invite de commande windows et vérifiez que Maven est correctement installé avec l'instruction `mvn --version`.
5. Exécutez `mvn`. Quel est le résultat ? S'il y a une erreur, résoudre le problème.
6. Créez un nouveau répertoire de TP ( `c:\TP_Maven` par exemple). Lancez l'instruction `mvn archetype:create -DgroupId=com.objis.mavenapps -DartifactId=monappli`. Que se passe-t-il ? Explorez les fichiers téléchargés dans votre répertoire de TP puis dans le répertoire `.m2` dans `Documents and Settings`.
7. En analysant le fichier `pom.xml`, précisez quel est la dépendance du projet `monappli`.
8. Placez-vous dans le répertoire `monappli` et exécutez l'instruction `mvn compile` pour compiler le projet téléchargé. Que remarquez-vous ?
9. Lancez les tests unitaires du projet `monappli` avec une commande maven ! Analysez les fichiers générés ?
10. Lancez l'instruction `mvn package`. Analysez le répertoire `target`.
11. Exécutez l'instruction `java -cp target/monappli-1.0-SNAPSHOT.jar com.objis.mavenapps.App`. Que se passe-t-il ?
12. Analysez bien les sorties écran lors de l'instruction `mvn install`. Qu'est ce qui a été ajouté par maven.
13. Utilisez l'instruction `mvn clean`. Que se passe-t-il ? et avec `mvn clean install` ?
14. Devinez que fait l'instruction `mvn site` ? Affichez le fichier `index.html` dans votre navigateur par défaut.
15. Malheureusement, le site est en anglais. À l'aide de l'[aide en ligne](#) de maven, modifiez votre `pom.xml` pour gérer l'internationalisation. Il est possible qu'une erreur apparaisse, supprimez la version dans ce que vous venez d'ajouter dans le fichier `pom.xml`.
16. Créer un nouveau répertoire de projet maven et utilisez l'instruction `mvn archetype:generate`. Maven va vous demander le numéro de projet ; en sachant qu'il va en afficher 800 !

## PLUGIN ECLIPSE

17. Importez le projet `monappli` dans Eclipse et testez l'ensemble des commandes vues précédemment.

18. Créez un nouveau projet maven dans Eclipse.



### Pour aller plus loin

19. Installer un calculateur de métriques logicielles dans Eclipse en suivant le tutoriel [suivant](#). Vous appliquerez le calcul des métriques sur un de vos projets de développement par exemple et vous modifierez votre code pour améliorer au moins un problème.
20. Un autre outil d'analyse de code appelé SONAR est plus utilisé. Renseignez vous.
21. Pour faire de l'intégration continue, étudiez l'outil Jenkins.

### Sources pour ce TP

Utiliser Maven : <http://matthieu-lux.developpez.com/tutoriels/java/maven/>

un exemple concret simple : <http://dcabasson.developpez.com/articles/java/maven/introduction-maven2/>

<http://www.objjs.com/formation-java/tutoriel-formation-maven-2.html>

<http://www.jmdoudoux.fr/java/dej/chap-maven.htm#maven>

Tutoriel Metrics (Plugin Eclipse) : <http://baptiste-wicht.developpez.com/tutoriels/eclipse/metrics/>