

TD6 : Git et Maven

Objectif du TD

À la fin de ce TD, vous devriez :

1. Comprendre les fonctions de bases de Maven et la structure d'un projet Maven ;
2. Connaître les commandes de base de Git.

Ce TD va se dérouler de manière interactive avec l'enseignant. Votre participation est attendue pour intégrer les concepts évoqués.

I) Prise en main de Maven

Maven est un outil de gestion de projet et d'automatisation de la construction principalement utilisé pour les projets Java. Il est conçu pour simplifier le processus de construction (compilation, tests, déploiement) d'un projet. Maven permet également de gérer les dépendances, c'est-à-dire les bibliothèques et autres ressources dont un projet a besoin pour fonctionner.

Maven gère différentes étapes du cycle de vie du projet, il permet notamment de configurer la compilation et les tests du projet. Le fichier `pom.xml` contient les informations de configuration du projet, ainsi que le nom du projet, la version, les dépendances utilisées, etc.

Exercice 1 : Compréhension

1. Téléchargez le zip TD6 sur Moodle et extrayez son contenu. Il contient un projet Maven. Ce projet est un mini-jeu où le joueur doit deviner un nombre choisi par le jeu. Observez et décrivez la structure du projet. *Il s'agit de la structure par défaut des répertoires dans un projet Maven.*
2. Quelles informations contient le fichier `pom.xml` ?

Exercice 2 : Utilisation

1. Si ce n'est pas déjà fait, installez Maven sur votre machine.
 - <https://maven.apache.org/install.html> pour Windows,
 - `brew install maven` sous MacOS,
 - `apt install maven` pour une distribution Linux.
2. La commande `mvn clean compile` permet de (re)compiler le projet. Compilez le projet, pour cela placez-vous à la racine du projet.
3. La commande `mvn exec:java -Dexec.mainClass="com.devinette.JeuDeviner"` permet de l'exécuter en précisant que le main à exécuter est celui de la classe `JeuDeviner`. Exécutez le projet.

La partie `<properties>` du `pom.xml` contient des valeurs spécifiques qui peuvent être utilisés dans le projet, comme les versions des dépendances à utiliser ou des variables.

4. Ajoutez la ligne suivante dans la section `<properties>` :
`<exec.mainClass>com.devinette.JeuDeviner</exec.mainClass>`
et exécutez le projet avec la commande `mvn exec:java`.

Exercice 3 : Création

La commande `mvn archetype:generate` permet de générer un projet à partir d'un archétype (modèle prédéfini pouvant être personnalisé). Pour cela, les informations à propos du projet sont à donner :

- le nom du groupe
- le nom du projet
- l'archétype que l'on souhaite utiliser.

Le nom du groupe correspond à un identifiant unique représentant le groupe auquel appartient le projet (entreprise, pôle, ...). Il a la forme d'un nom de domaine inversé, par exemple : `fr.UPC.GL.projects`.

La commande suivante permet de créer le squelette du projet `jeu-devinette` avec l'archétype le plus simple et en utilisant les valeurs par défaut (le mode interactif permettant de changer ces valeurs étant désactivé).

```
mvn archetype:generate -DgroupId=com.devinette -DartifactId=jeu-devinette  
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

1. Créez un nouveau projet maven pour un jeu de pong. Le projet s'appellera 'pong' et sera contenu dans un package 'games'.
2. Quelle dépendance a été automatiquement intégrée ? Quand cette dépendance sera-t-elle utilisée ?

Informations complémentaires

Voici les commandes de bases pour exécuter Maven :

- `mvn compile` pour compiler le projet
- `mvn test` pour lancer les tests
- `mvn package` pour créer un package
- `mvn install` pour installer le projet dans le dépôt local (le rendre accessible par d'autres projets en local)
- `mvn deploy` pour déployer le projet dans un dépôt distant
- `mvn exec:java` pour exécuter le projet

Une dépendance peut avoir une certaine portée (scope) : `compile`, `test`, `provided`, `runtime`.

II) UML et Git

Exercice 4 :

1. Proposez un diagramme d'activités décrivant les étapes pour modifier un répertoire Git distant. Chaque activité correspondra à une ligne de commande git. Doivent apparaître a minima :
 - `git status`
 - `git add`
 - `git rm`
 - `git commit`
 - `git push`