# Introduction à l'utilisation de Maven


Encadré par :

Pr. NAFIL Khalid

Réalisé par :

ZOUAHRI Jihane

JAMAL Aymane


Année Universitaire : 2023-2024

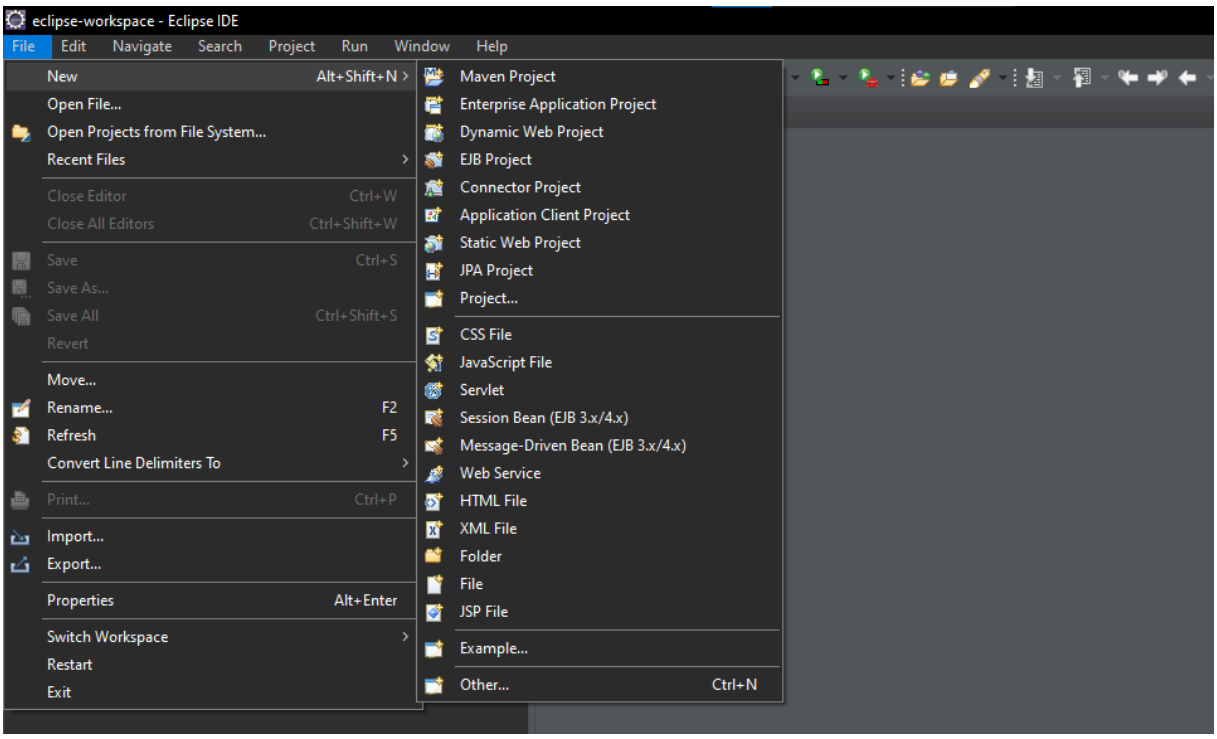
**Maven**, un outil puissant de gestion de projet et de construction pour Java, rationalise le processus de développement en offrant **une structure cohérente** et une **automatisation pour les projets**. Son intégration avec des Environnements de Développement Intégrés (IDE) populaires tels qu'Eclipse et IntelliJ IDEA **améliore l'efficacité des développeurs Java**. Dans cette introduction, nous allons explorer comment exploiter les capacités de Maven de manière transparente à la fois dans **Eclipse** et **IntelliJ IDEA.**

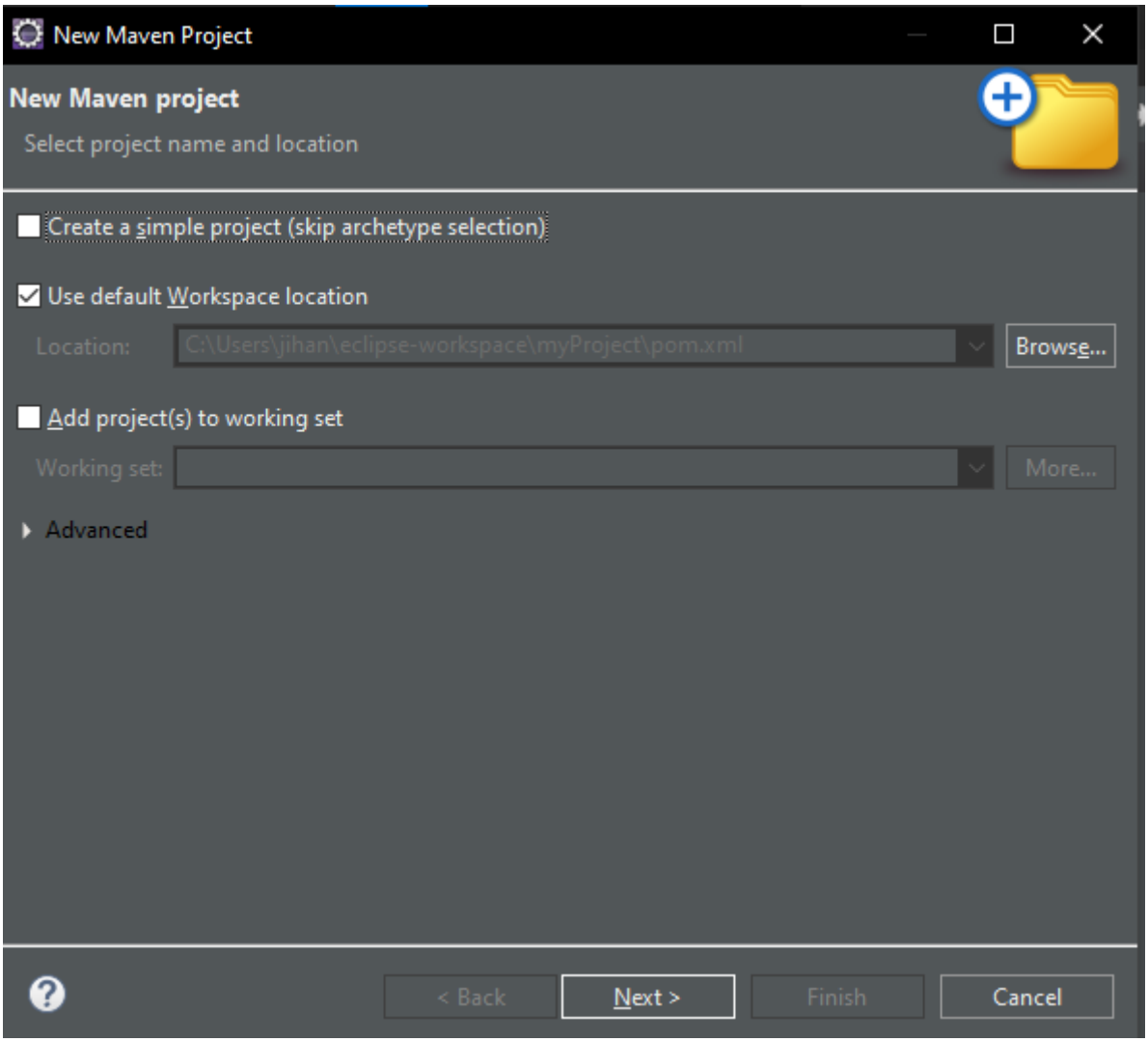
## 1) Maven dans Eclipse :

**Eclipse**, un IDE Java largement utilisé, est livré avec le plugin "m2e", simplifiant l'intégration de Maven.

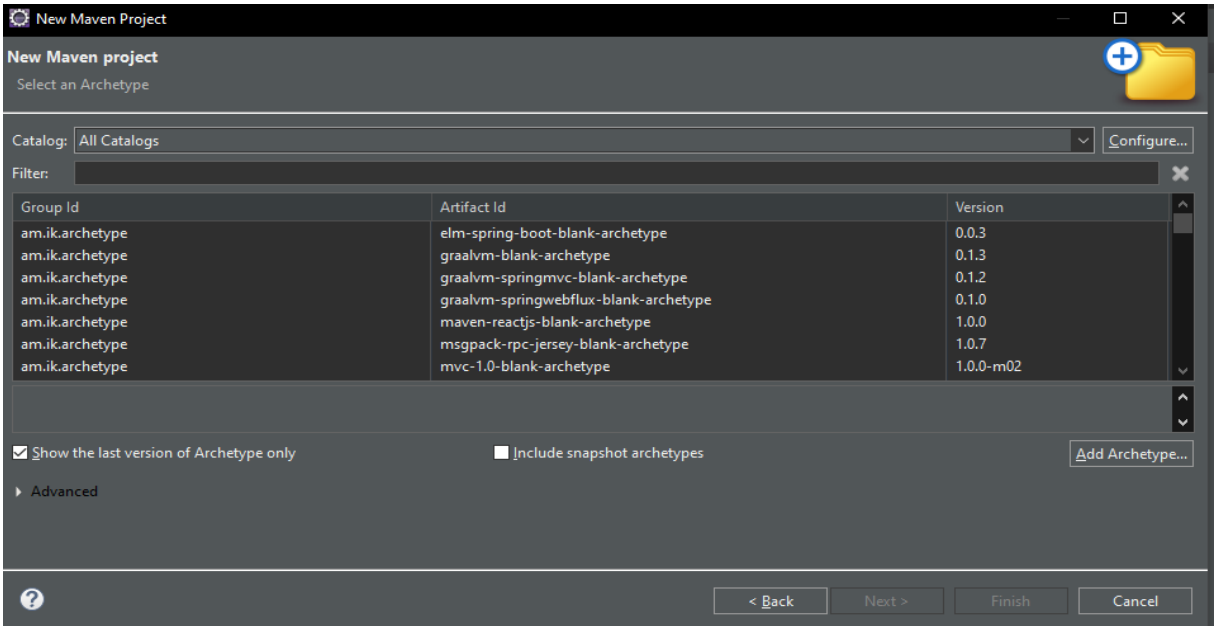
➔ **Creation de projet Maven dans eclipse :**

- Ouvrez Eclipse.
- Lancez Eclipse sur votre machine.
- Allez dans le menu "File" > "New" > "Maven Project."





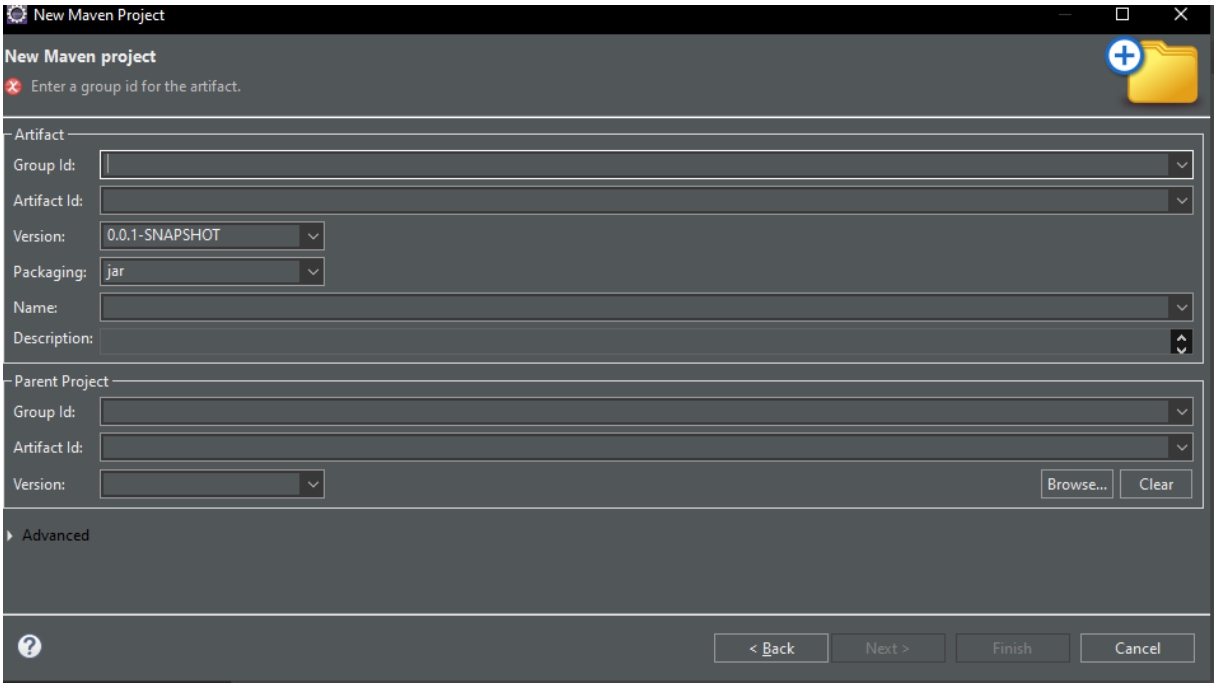

- **Suivez l'Assistant pour configurer votre projet, en spécifiant l'ID du groupe, l'ID de l'artefact, et d'autres détails**.



**Un artefact Maven** est un modèle de projet prédéfini qui est utilisé comme point de départ lors de la création d'un nouveau projet Maven. Un archetype définit la structure de base du projet, y compris l'organisation des répertoires, les fichiers de configuration, les dépendances initiales, et d'autres paramètres de projet.

Si vous ne souhaitez pas utiliser de modèle (archetype) prédéfini et que vous préférez créer un projet simple sans modèle particulier, vous pouvez généralement choisir l'option "Create a simple project (skip archetype selection)" sur l'étape précédente.

- **Suivez l'assistant pour définir les paramètres de votre projet, tels que l'ID du groupe, l'ID de l'artefact, etc.**




### ❖ Group ID (ID du Groupe) :

Il identifie votre organisation ou groupe de projets. Utilisez une convention de nommage inversée de domaine, par exemple, com.example.

### ❖ Artifact ID (ID de l'Artefact) :

C'est le nom unique de votre projet. Il s'agit généralement du nom du produit ou de l'application. Par exemple, my-app.

### ❖ Version :

La version actuelle de votre projet. Vous pouvez utiliser des numéros de version standard tels que 1.0-SNAPSHOT. Un SNAPSHOT indique une version en développement.

### ❖ Packaging :

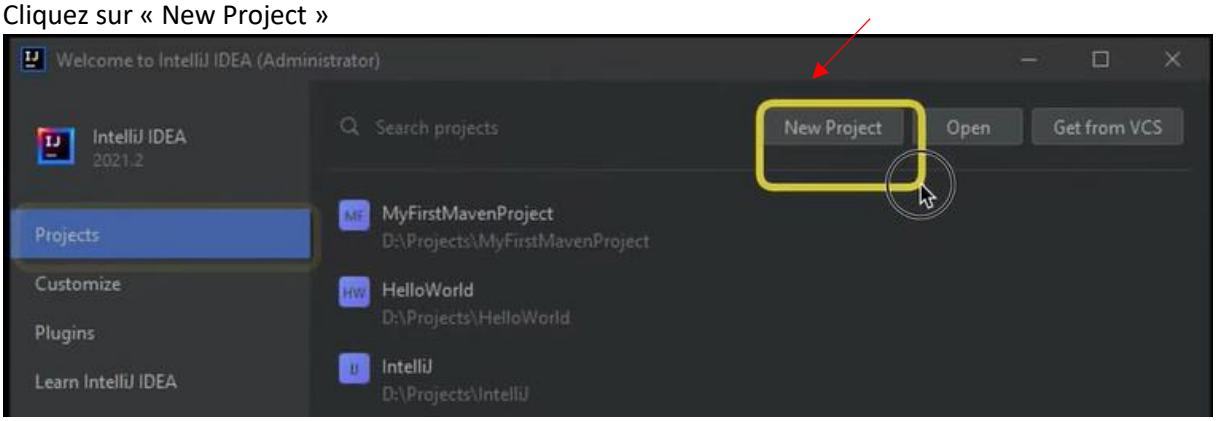
Le type d'emballage pour votre projet. Par défaut, il est généralement configuré en tant que jar pour les projets Java standard.

En suivant ces étapes, vous fournissez les informations essentielles nécessaires pour configurer votre projet Maven. Ces détails seront enregistrés dans **le fichier pom.xml** qui servira de configuration centrale pour votre projet Maven.

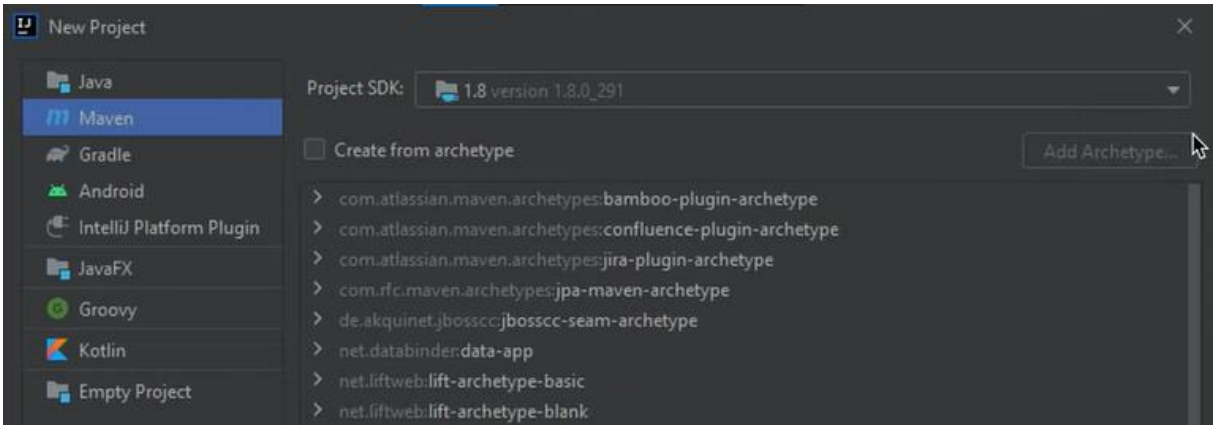
## 2) Maven dans IntelliJ IDEA :

**IntelliJ IDEA**, un autre IDE Java populaire, s'intègre parfaitement à Maven, offrant une expérience de développement fluide et productive

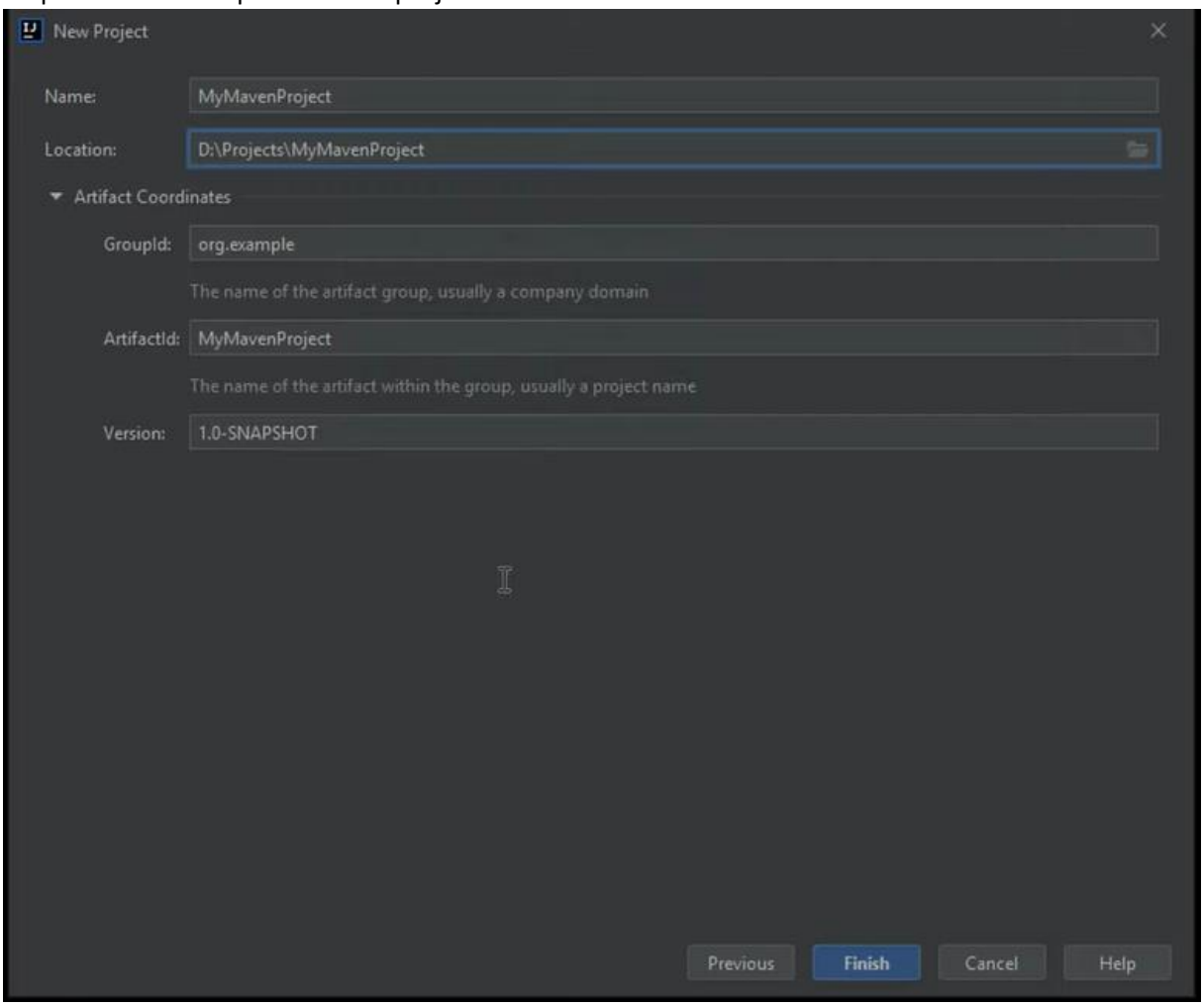
- Lancez IntelliJ IDEA sur votre ordinateur.
- Cliquez sur « New Project »




- Choisissez "Maven" parmi les options proposées.
- Sélectionnez un artefact si vous souhaitez utiliser un modèle (archetype) prédéfini, sinon, cochez "Create from archetype" pour créer un projet simple sans modèle.



- Suivez l'assistant pour spécifier l'ID du groupe, l'ID de l'artefact, la version, etc.
- Cliquez sur "Finish" pour créer le projet.



## 3) Structure de projet :

### a) Dossier Principal (src/main) :

C'est le dossier principal où on place le code source Java.
La structure à l'intérieur de ce dossier comprend généralement des sous-dossiers tels que :

- **src/main/java :** Contient les fichiers de code source Java (fichiers .java).
- **src/main/resources :** Contient des ressources non Java faisant partie de l'application, comme des fichiers de configuration, des fichiers XML et d'autres ressources qui ne sont pas du code source Java.

### b) Dossier de Test (src/test) :

Ce dossier est dédié aux tests du code.
On utilise ce dossier pour écrire des tests unitaires et effectuer d'autres types de tests. La structure à l'intérieur de ce dossier comprend des sous-dossiers tels que :

- **src/test/java** : Contient les fichiers de code source pour les classes de test .
- **src/test/resources :** On y stocke les fichiers de ressources nécessaires aux tests, tels que des données d'entrée, des fichiers de configuration ou d'autres ressources nécessaires à des fins de test.

### c) Fichier pom.xml :

Le fichier Project Object Model (POM), est au cœur d'un projet Maven. Il contient des informations de configuration sur le projet, ses dépendances, les plugins, les objectifs, et divers autres paramètres.

- On y définis les métadonnées du projet, les dépendances, les paramètres de construction, et d'autres configurations à l'intérieur de ce fichier XML.

- Composé de 2 parties primordiales :
  - **Dépendances**

- Responsables du téléchargement des bibliothèques tierces (third party libraries) et des pilotes (drivers) liés à notre projet.

- la balise <dependencies>... ... </dependencies> , où nous spécifierons toutes les dépendances

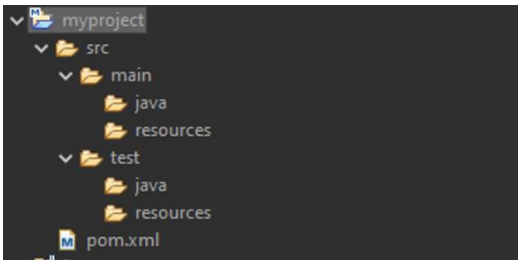
- Répertoire local Maven : lors de la création de votre projet Maven, le répertoire local Maven est créé par défaut. Lorsque vous ajoutez des dépendances dans la balise des dépendances, il téléchargera automatiquement toutes les bibliothèques, tous les fichiers JAR tiers dans le répertoire local Maven. Ces bibliothèques seront téléchargées depuis le référentiel Maven distant https://mvnrepository.com/.
- Si vous souhaitez utiliser la dernière version d'une bibliothèque, il vous suffit de changer le numéro de version dans la balise des dépendances.

  - **Plugins**

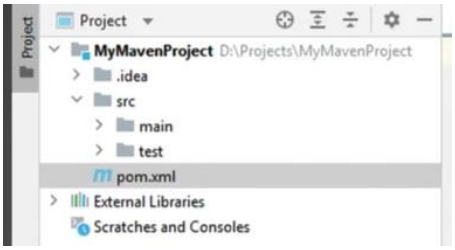
    - Cette partie contient les différents types de configurations sur lesquelles notre projet s'exécutera, par exemple, le plugin Maven Surefire, le plugin Maven Compiler...

- Dans la balise **<plugins>...** où nous spécifierons tous les plugins... **</plugins>**.

**d) Structure de dossiers typique pour un projet Java qui suit les conventions de Maven sur eclipse**




**e) Structure de dossiers typique pour un projet Java qui suit les conventions de Maven sur IntelliJ IDEA**




## 4) Mini Projet :

**Voir la capsule**