

# Préparer un projet Maven + JavaFX

Petit tutoriel pour aider à la création d'un projet JavaFX vierge pour le projet d'ING 1. Le tutoriel est prévu pour fonctionner avec Linux, Windows et Mac OS.

## Installer Maven

Maven peut être installé via le terminal ou manuellement sur n'importe lequel des 3 OS.

Je vous recommande de l'installer via le terminal : <https://maven.apache.org/install.html>

Sur Windows, si vous ne possédez ni Scoop, ni Chocolatey, vous pouvez télécharger le JAR de Maven [ici](#), dézipper son contenu et ajouter le dossier de destination à votre Path. Pour ce faire, il faut ouvrir le menu « Variables d'environnement » (nom approximatif, j'ai pas touché à Windows depuis un temps) depuis Cortana, cliquer sur « Modifier les variables d'environnement » et ajouter le chemin vers le dossier qui contient le fichier « mvn.cmd ». Si votre terminal était ouvert, vous aurez probablement besoin de le fermer et d'en ouvrir un autre.

Pour vérifier que votre installation de Maven fonctionne, vous pouvez faire la commande suivante qui vous affichera la version utilisée :

```
mvn --version
```

## Maven wrapper

Le maven wrapper est un ensemble de deux petits scripts (un pour Linux/Mac, un pour Windows) permettant d'automatiquement télécharger et lancer Maven sans besoin de faire les étapes au dessus et en étant sûr d'utiliser la même version que les autres membres du projet. Ces deux fichiers ont uniquement besoin d'être placés à la racine du projet et peuvent (et devraient) être mis sur votre dépôt Git.

De cette façon, un seul camarade a besoin d'installer Maven et d'exécuter la commande ci-dessous pour que tout le monde puisse utiliser compiler et lancer le projet :

```
mvn wrapper:wrapper
```

## Créer un projet JavaFX avec Maven

Pour créer un projet JavaFX avec Maven il suffit d'une commande :

```
mvn archetype:generate \
-DarchetypeGroupId=org.openjfx \
-DarchetypeArtifactId=javafx-archetype-simple \
-DarchetypeVersion=0.0.6 \
-DgroupId=org.openjfx \
-DartifactId=sample \
-Dversion=1.0.0 \
-Djavafx-version=24.0.1
```

Les arguments « archetype... » n'ont pas à être modifiés ici. Il s'agit des arguments permettant de dire à Maven quelle template utiliser.

Voici la liste des arguments à modifier :

- groupId : Le nom du package. Par convention, doit-être le nom de domaine « à l'envers » e.g « org.atilla ». J'utilise personnellement « fr.cyu » pour les projets d'école.
- artifactId : Le nom du projet, en minuscules, les mots séparés par des espaces e.g « projet-ing1 »
- version : La version de votre projet
- javafx-version : La version de JavaFX de votre projet. Le premier nombre représente aussi la version de Java nécessaire. Par exemple sur le Ubuntu par défaut de l'école où Java 11 est installé, vous devrez utiliser JavaFX 11.0.2 ou mettre à jour la version de Java (parlez-en au prof si c'est le cas !)

Note : cette commande génère le projet dans un nouveau dossier. Pas besoin d'en créer un.

Ne vous reste plus une fois cette étape faite qu'à exécuter la commande pour générer le wrapper dans le dossier du projet et à push sur votre dépôt Git. Les autres membres du projet devront simplement utiliser ./mvnw (ou ./mvnw.cmd sur Windows) à la place de « mvn ».

## Compiler et lancer le projet

Compiler le projet :

```
mvn compile
```

Lancer le projet JavaFX :

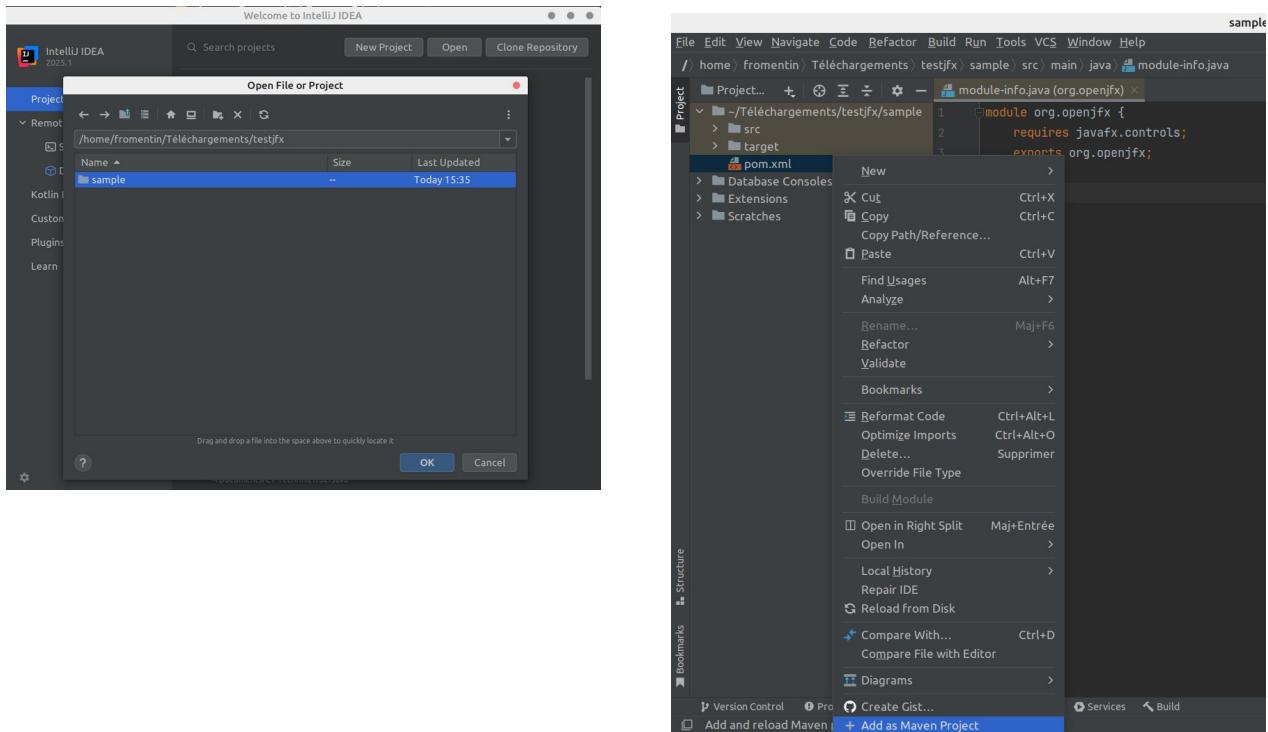
```
mvn javafx:run
```

## Import du projet dans l'IDE

### IntelliJ IDEA (recommandé)

IntelliJ IDEA détecte automatiquement qu'un projet utilise Maven lorsqu'il est ouvert pour la première fois.

Si IntelliJ ne détecte pas le projet Maven ou que vous aviez déjà ouvert le projet avant, vous pouvez clic-droit sur le pom.xml puis « Add as Maven Project ».



## Eclipse

Sur Eclipse, un projet Maven peut être importé via File → Import → Maven → Existing Maven Projects.

