

# Maven Lite

---

- Version : 2.0.0

## Table des matières

- [Maven Lite](#)
  - [Table des matières](#)
  - [Compatibilité](#)
  - [Description](#)
  - [Prérequis](#)
  - [Installation](#)
    - [Windows](#)
      - [Installation automatique - Windows](#)
      - [Installation manuelle - Windows](#)
      - [Désinstallation automatique - Windows](#)
      - [Désinstallation manuelle - Windows](#)
    - [Linux](#)
      - [Installation automatique - Linux](#)
      - [Installation manuelle - Linux](#)
      - [Désinstallation automatique - Linux](#)
      - [Désinstallation manuelle - Linux](#)
    - [MacOS](#)
      - [Installation automatique - MacOS](#)
      - [Installation manuelle - MacOS](#)
      - [Désinstallation automatique - MacOS](#)
      - [Désinstallation manuelle - MacOS](#)
  - [Utilisation](#)
    - [Exemple, fonctionnalités et limites](#)
  - [Installation automatique](#)
    - [Windows](#)
    - [Linux](#)
  - [Installation manuelle](#)
    - [Windows](#)
    - [Linux Avec les droits administrateurs](#)
    - [Linux Sans les droits administrateurs](#)
  - [Désinstallation automatique](#)
    - [Windows](#)
    - [Linux](#)
  - [Désinstallation manuelle](#)
    - [Windows](#)
    - [Linux](#)

## Compatibilité

- Windows 10 (Non testé)
- Windows 11 (Non testé)
- Linux possédant `bash` (Testé sur Ubuntu 23.10 Mantic Minotaur)
- MacOS Mojave 10.14.6 et inférieur (Non testé)
- MacOS Catalina 10.15 et supérieur à condition d'avoir installé `bash` (Non testé)

## Description

Maven Lite est un système de gestion de projet java qui permet de compiler et lancer des projets java très simplement et rapidement.

Le but est de pouvoir compiler et/ou lancer un projet java en une seule courte commande, comme maven mais en plus simple.

Maven Lite est plus simple que Maven car il ne gère pas les dépendances hors local, les plugins, les phases de build, les déploiements automatique, etc. Il est donc adapté pour les projets simples qui ne nécessitent pas de gérer des dépendances externes, cependant Maven Lite gère les dépendance local sous forme de fichier jar. Pour des projets plus complexes, il est conseillé d'utiliser Maven.

Maven lite permet de compiler le projet dans un dossier target, gérer les dépendances locales (fichier jar) et exécuter le projet.

L'intérêt principale de maven lite dans la gestion des dépendances est de fournir un dossier qui contient toutes les dépendances, le script ajoutera automatiquement toutes les dépendances dans le classpath.

Il est possible que cette deuxième version de Maven Lite contienne des bugs. Si vous en trouvez, n'hésitez pas à les signaler.

## Prérequis

- Disposer des droits administrateurs si vous voulez utiliser les installations automatiques.

Si vous ne disposez pas des droits administrateurs, vous pouvez installer maven lite manuellement en suivant les instructions de la partie 'Installation manuelle'.

## Installation

### Windows

#### Installation automatique - Windows

#### Installation manuelle - Windows

#### Désinstallation automatique - Windows

#### Désinstallation manuelle - Windows

### Linux

#### Installation automatique - Linux

#### Installation manuelle - Linux

#### Désinstallation automatique - Linux

#### Désinstallation manuelle - Linux

### MacOs

#### Installation automatique - MacOS

#### Installation manuelle - MacOS

#### Désinstallation automatique - MacOS

#### Désinstallation manuelle - MacOS

## Utilisation

- placer vous dans le dossier de votre projet java et exécuter la commande suivante :

```
mvnl [options] [arguments]
```

Exemple, fonctionnalités et limites

Vous avez un projet java dans le dossier 'C:\Users\user\Documents\MonProjetJava' Tout le code source se trouve dans le dossier 'src' et les fichiers .class doivent être compilés dans le dossier 'bin'. Votre projet contient des dépendances sous forme de fichier jar qui se trouve dans le dossier 'lib'.

Pour compiler le projet, vous devez ouvrir un terminal, vous placer dans le dossier 'C:\Users\user\Documents\MonProjetJava' puis exécuter la commande suivante :

```
mvn -s src -o bin -d lib -c
```

Maintenant que votre projet est compilé, vous voulez lancer la classe principal 'Main.java' dans le package 'com.exemple'. Pour cela, vous devez exécuter la commande suivante :

```
mvn -cp bin -d lib -m com.exemple.Main -l
```

Il est possible de combiner les deux commandes en une seule :

```
mvn -s src -o bin -d lib -cp bin -m com.exemple.Main -cl
```

Comme vous pouvez le voir, ça fait une commande assez longue à taper. C'est pour cela que vous pouvez créer un fichier de configuration, par exemple 'config.txt' puis mettre les arguments dedans.

Le fichier de configuration doit contenir les options et argument de la commande. Soit une option avec son argument par ligne, soit plusieurs options et arguments sur la même ligne séparé par un espace.

Les options du fichier de configuration prévalent sur les options de la ligne de commande, il n'est donc pas possible de modifier une option dans la ligne de commande si elle est déjà présente dans le fichier de configuration. Cependant, il est possible de mettre une partie des options dans le fichier de configuration et le reste dans la ligne de commande.

Exemple de fichier de configuration :

```
-s src -o bin  
-d lib -cp bin -m com.exemple.Main
```

- Une fois que vous avez créé le fichier de configuration, vous pouvez exécuter la commande suivante pour compiler et lancer le projet :

```
mvn -f config.txt -cl
```

Je vous recommande de ne pas mettre les options '-c' et '-l' dans le fichier de configuration car vous n'avez pas besoin de compiler et lancer le projet à chaque fois. Cela vous permettra de les mettre dans la commande quand vous en aurez besoin. Malgré tout, vous pouvez les mettre dans le fichier de configuration si vous le souhaitez.

- Pour compiler le projet, vous pouvez exécuter la commande suivante :

```
mvn -f config.txt -c
```

- Pour lancer le projet, vous pouvez exécuter la commande suivante :

```
mvn -f config.txt -l
```

Les options inutiles comme l'option -s pour lancer le projet seront ignorées, donc pas de problème si vous les laissez dans le fichier de configuration.

L'ordre des options n'a pas d'importance sauf les options -f, -h et -v qui doivent être la première option si elles sont utilisées.

## Installation automatique

### Windows

- Ouvrir un terminal en tant qu'administrateur
- Se placer dans le dossier de votre choix, par exemple 'C:\Users\user\Documents'
- Clonez le dépôt git :

```
git clone https://github.com/FloRobart/Maven_lite.git
```

- Se placer dans le dossier 'Maven\_lite'

```
cd Maven_lite
```

- Exécutez le script d'installation :

```
.\installer.bat
```

- Une fois l'installation terminée, vous pouvez supprimer le dossier 'Maven\_lite' si vous le souhaitez.

### Linux

- Téléchargez le [fichier debian](#) dans la section 'Releases' du dépôt git
- Ouvrir un terminal
- Installez le paquet :

```
sudo dpkg -i mavenLite_1.2-1_all.deb
```

- Une fois l'installation terminée, vous pouvez supprimer le fichier 'mavenLite\_1.2-1\_all.deb' si vous le souhaitez.

## Installation manuelle

Si vous utilisez les mêmes chemins d'installation que dans les exemples, vous pourrez utiliser les scripts de désinstallation automatique pour désinstaller maven lite.

### Windows

- copier le fichier 'maven\_lite.bat' dans le dossier de votre choix, par exemple 'C:\Program Files\Maven\_Lite'
- renommez le fichier 'maven\_lite.bat' en 'mvnl.bat'
- Ajoutez le dossier 'C:\Program Files\Maven\_Lite' dans la variable d'environnement utilisateur 'PATH' soit grâce à l'interface graphique, soit en exécutant la commande suivante :

```
setx path "%PATH%;C:\Program Files\Maven_Lite"
```

### Linux Avec les droits administrateurs

- Ajoutez les droits d'exécution au fichier 'maven\_lite.sh' :

```
chmod +x /home/$USER/.maven_lite/maven_lite.sh
```

- copier le fichier 'maven\_lite.sh' dans le dossier `/bin`

```
sudo cp maven_lite.sh /bin/maven_lite.sh
```

- renommez le fichier 'maven\_lite.sh' en 'mvnl'

```
sudo mv /bin/maven_lite.sh /bin/mvnl
```

## Linux Sans les droits administrateurs

- copier le fichier 'maven\_lite.sh' dans le dossier de votre choix, par exemple '/home/\$USER/.maven\_lite'

```
cp maven_lite.sh /home/$USER/.maven_lite/maven_lite.sh
```

- Ajoutez les droits d'exécution au fichier 'maven\_lite.sh' :

```
chmod +x /home/$USER/.maven_lite/maven_lite.sh
```

- Ajoutez une alias dans le fichier de configuration de votre shell, par exemple dans le fichier '.bashrc' :

```
# Maven Lite
alias mvnl='bash /home/$USER/.maven_lite/maven_lite.sh'
# Fin Maven Lite
```

- Assurez-vous de bien mettre les commentaires comme dans l'exemple ci-dessus pour pouvoir désinstaller Maven Lite facilement en cas de besoins ou pour éviter les problèmes en cas d'exécution du script d'installation.
- Rechargez votre fichier de configuration de votre shell, par exemple en exécutant la commande suivante :

```
source /home/$USER/.bashrc
```

## Désinstallation automatique

### Windows

- Téléchargez le fichier de désinstallation `unistaller.bat`.
- Exécutez le script de désinstallation :

```
.\unistaller.bat
```

- Une fois la désinstallation terminée, vous pouvez supprimer le fichier 'unistaller.bat' si vous le souhaitez.

### Linux

- Ouvrir un terminal
- Exécutez la commande suivante :

```
mvnl-uninstall
```

## Désinstallation manuelle

### Windows

- Supprimer le dossier d'installation de maven lite, par exemple le dossier 'C:\Program Files\Maven\_Lite'
- Supprimer ce même dossier, par exemple 'C:\Program Files\Maven\_lite' de la variable d'environnement utilisateur 'PATH' grâce à l'interface graphique

### Linux

- Supprimer le fichier `/usr/bin/mvnl` et `/usr/bin/mvnl-uninstall`

```
sudo rm /usr/bin/mvnl /usr/bin/mvnl-uninstall
```

- Supprimer les pages de manuel

```
sudo rm /usr/local/man/en/man1/mvnl.1.gz /usr/local/man/fr/man1/mvnl.1.gz
```