

Maven Lite

See the [english documentation](#)

- Version : 2.0.0
- Auteur : Floris Robart

Table des matières

- [Maven Lite](#)
 - [Table des matières](#)
 - [Compatibilité](#)
 - [Description](#)
 - [Prérequis](#)
 - [Installation des prérequis](#)
 - [Installation de bash](#)
 - [Installation de java](#)
 - [Installation et désinstallation](#)
 - [Debian et dérivés](#)
 - [Installation automatique - Debian](#)
 - [Installation manuelle - Debian](#)
 - [Désinstallation automatique - Debian](#)
 - [Désinstallation manuelle - Debian](#)
 - [Autres distributions Linux](#)
 - [Installation automatique - Linux](#)
 - [Installation manuelle - Linux](#)
 - [Désinstallation automatique - Linux](#)
 - [Désinstallation manuelle - Linux](#)
 - [Windows](#)
 - [Installation automatique - Windows](#)
 - [Installation manuelle - Windows](#)
 - [Désinstallation automatique - Windows](#)
 - [Désinstallation manuelle - Windows](#)
 - [MacOS](#)
 - [Installation automatique - MacOS](#)
 - [Installation manuelle - MacOS](#)
 - [Désinstallation automatique - MacOS](#)
 - [Désinstallation manuelle - MacOS](#)
 - [Liste des options](#)
 - [Utilisation de Maven Lite](#)
 - [Ligne de commande](#)
 - [Possibilités de la ligne de commande](#)
 - [Fichier de configuration](#)
 - [Possibilités du fichier de configuration](#)
 - [create](#)
 - [mvc](#)
 - [compilation](#)
 - [launch](#)
 - [compile-launch](#)
 - [launch-compile](#)
 - [quiet](#)
 - [verbose](#)
 - [exclude](#)
 - [jar](#)
 - [source](#)
 - [target](#)

- [classpath](#)
- [libraries](#)
- [arguments](#)
- [main](#)
- [encoding](#)
- [export](#)
- [maven](#)
- [version](#)
- [help](#)
- [clear](#)
- [Exemple, fonctionnalisés et limites](#)

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.

Un fichier de configuration est disponible pour configurer les options de Maven Lite. Il est possible de se passer de fichier configuration et de mettre les options dans la ligne de commande mais c'est moins pratique.

L'ordre des options n'a pas d'importance.

Chaque option peut être utilisée plusieurs fois.

Chaque argument doit être précédé de son option, vous ne pouvez pas mettre d'argument sans option ni mettre toutes les options au début puis tous les arguments à la fin. Par exemple, `mvnl --source src/main/java --target target` est valide mais `mvnl --source --target src/main/java target` ne l'est pas.

Il est recommandé de laisser les options par défaut parce que Maven Lite est basé sur les conventions simplifier de Maven qui sont des conventions de nommage et d'arborescence connues de beaucoup de développeurs java.

Prérequis

- Disposer des droits administrateurs si vous voulez utiliser les installations automatiques.
- Avoir installé [bash](#)
- Avoir installé [java](#)

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

Installation des prérequis

Installation de bash

Installation de java

Installation et désinstallation

Debian et dérivés

Installation automatique - Debian

- Téléchargez le [fichier debian](#) dans la langue de votre choix dans la section 'Releases' du dépôt Github
 - [Téléchargez le fichier de la version française.](#)
 - [Téléchargez le fichier de la version anglaise.](#)
- Exécutez le fichier debian

```
sudo dpkg -i maven-lite_<langue>_2.0-1_all.deb
```

- Une fois l'installation terminée, vous pouvez supprimer le fichier `maven-lite_<langue>_2.0-1_all.deb` si vous le souhaitez.

Installation manuelle - Debian

- Téléchargez le [Fichier compressé](#) contenant les fichiers de l'applications dans la langue de votre choix dans la section 'Releases' du dépôt Github
 - [Téléchargez le fichier de la version française.](#)
 - [Téléchargez le fichier de la version anglaise.](#)
- Décompressez le fichier compressé
- Déplacez le dossier `maven-lite` dans le dossier `/etc`

```
sudo mv maven-lite /etc/
```

- Déplacez les fichiers shell dans le dossier `/usr/local/bin`

```
sudo mv mvnl /usr/local/bin/  
sudo mv mvnl-uninstall /usr/local/bin/
```

- Ajoutez les droits d'exécution au fichier `mvnl` et `mvnl-uninstall`

```
sudo chmod +x /usr/local/bin/mvnl*
```

- Déplacez le dossier `man` dans le dossier `/usr/local/` pour avoir les pages de manuel

```
sudo mv man /usr/local/
```

- Vous pouvez supprimer le reste des fichiers inutilisés.

Désinstallation automatique - Debian

- Exécutez la commande suivante

```
mvnl-uninstall
```

Désinstallation manuelle - Debian

- Supprimez le dossier `/etc/maven-lite`

```
sudo rm -r /etc/maven-lite
```

- Supprimez les fichiers shell

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

- Supprimez les pages de manuel

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

Autres distributions Linux

Installation automatique - Linux

- Téléchargez le [Fichier compressé](#) contenant les fichiers de l'application dans la langue de votre choix dans la section 'Releases' du dépôt Github
 - [Téléchargez le fichier de la version française.](#)
 - [Téléchargez le fichier de la version anglaise.](#)
- Décompressez le fichier compressé
- Exécuter le script d'installation

```
sudo ./installer.sh
```

- Vous pouvez supprimer le reste des fichiers inutilisés.

Installation manuelle - Linux

- Téléchargez le [Fichier compressé](#) contenant les fichiers de l'application dans la langue de votre choix dans la section 'Releases' du dépôt Github
 - [Téléchargez le fichier de la version française.](#)
 - [Téléchargez le fichier de la version anglaise.](#)
- Décompressez le fichier compressé
- Déplacez le dossier `maven-lite` dans le dossier `/etc`

```
sudo mv maven-lite /etc/
```

- Ajoutez les droits d'exécution au fichier `mvnl` et `mvnl-uninstall`

```
sudo chmod +x /usr/local/bin/mvnl*
```

- Déplacez le dossier `man` dans le dossier `/usr/local/share` pour avoir les pages de manuel

```
sudo mv man /usr/local/share/
```

- Vous pouvez supprimer le reste des fichiers inutilisés.

Désinstallation automatique - Linux

- Exécutez la commande suivante

```
mvnl-uninstall
```

Désinstallation manuelle - Linux

- Supprimez le dossier `/etc/maven-lite`

```
sudo rm -r /etc/maven-lite
```

- Supprimez les fichiers shell

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

- Supprimez les pages de manuel

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

Windows

Installation automatique - Windows

- Téléchargez le [Fichier compressé](#) contenant les fichiers de l'application dans la langue de votre choix dans la section 'Releases' du dépôt Github
 - [Téléchargez le fichier de la version française.](#)
 - [Téléchargez le fichier de la version anglaise.](#)
- Décompressez le fichier compressé

Installation manuelle - Windows

- Téléchargez le [Fichier compressé](#) contenant les fichiers de l'application dans la langue de votre choix dans la section 'Releases' du dépôt Github
 - [Téléchargez le fichier de la version française.](#)
 - [Téléchargez le fichier de la version anglaise.](#)
- Décompressez le fichier compressé

Désinstallation automatique - Windows

Désinstallation manuelle - Windows

MacOs

Installation automatique - MacOS

- Téléchargez le [Fichier compressé](#) contenant les fichiers de l'application dans la langue de votre choix dans la section 'Releases' du dépôt Github
 - [Téléchargez le fichier de la version française.](#)

- [Téléchargez le fichier de la version anglaise.](#)
- Décompressez le fichier compressé
- Exécuter le script d'installation

```
sudo ./installer.sh
```

- Vous pouvez supprimer le reste des fichiers inutilisés.

Installation manuelle - MacOS

- Téléchargez le [Fichier compressé](#) contenant les fichiers de l'application dans la langue de votre choix dans la section 'Releases' du dépôt Github
 - [Téléchargez le fichier de la version française.](#)
 - [Téléchargez le fichier de la version anglaise.](#)
- Décompressez le fichier compressé
- Déplacez le dossier `maven-lite` dans le dossier `/etc`

```
sudo mv maven-lite /etc/
```

- Ajoutez les droits d'exécution au fichier `mvnl` et `mvnl-uninstall`

```
sudo chmod +x /usr/local/bin/mvnl*
```

- Déplacez le dossier `man` dans le dossier `/usr/local/share` pour avoir les pages de manuel

```
sudo mv man /usr/local/share/
```

- Vous pouvez supprimer le reste des fichiers inutilisés.

Désinstallation automatique - MacOS

- Exécutez la commande suivante

```
mvnl-uninstall
```

Désinstallation manuelle - MacOS

- Supprimez le dossier `/etc/maven-lite`

```
sudo rm -r /etc/maven-lite
```

- Supprimez les fichiers shell

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

- Supprimez les pages de manuel

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

Liste des options

- Courte, Longue
 - Description
 - Nombre d'arguments
 - Valeur par défaut
- -f , --file
 - Permet de charger les options à partir d'un fichier de configuration. En savoir plus :
 - Nombre de d'arguments : de 0 à 1.
 - Valeur par défaut : `LPOM.conf`.
- -cr , --create
 - Créer l'arborescence du projet ainsi qu'un fichier de config par défaut.
 - Nombre de d'arguments : de 0 à 1.
 - Valeur par défaut : `NewProject`.
- -mvc , --model-view-controller
 - Permet de spécifier à l'option '--create' de créer l'arborescence d'un projet MVC.
 - Nombre de d'argument : 0.
- -c , --compilation
 - Compile le projet.
 - Nombre de d'argument : 0.
- -l , --launch
 - Lance le projet.
 - Nombre de d'argument : 0.
- -cl , --compile-launch
 - Compile et lance le projet. (équivalent à -c -l)
 - Nombre de d'argument : 0.
- -lc , --launch-compile
 - Compile et lance le projet. (équivalent à -c -l)
 - Nombre de d'argument : 0.
- -q , --quiet
 - Permet de supprimer l'affichage de java dans le terminal lors de l'exécution du projet.
 - Nombre de d'argument : 0.
- -v , --verbose
 - Permet d'afficher les commandes exécutées.
 - Nombre de d'argument : 0.
- -ex , --exclude
 - Permet d'exclure des fichiers java et des dossiers de la compilation. Si vous voulez passé un argument qui commencer par '-' échapper le caractère '-' avec deux \" comme ceci : '-ex \-fichier'.
 - Nombre de d'argument : `unlimited`.
- -s , --source
 - Dossier contenant les fichiers java à compiler.
 - Nombre de d'argument : 1.
 - Valeur par défaut : `src/main/java`.
- -t , --target
 - Dossier de sortie des fichiers compilés. Ce dossier sera créer si il n'existe pas et sera automatiquement ajouter au classpath lors de la compilation et du lancement.
 - Nombre de d'argument : 1.
 - Valeur par défaut : `target`.
- -cp , --classpath
 - Permet de spécifier le classpath à utiliser lors de la compilation et du lancement. Si vous voulez ajouter plusieurs éléments au classpath, il faut les séparer par des ':'
 - Nombre de d'argument : 1.
- -lib , --libraries
 - Dossier contenant les fichiers jar utiliser par le programme. Tout les fichiers jar seront ajoutés au classpath lors de la compilation et du lancement.

- Nombre de d'arguments : de 0 à 1.
- Valeur par défaut : `src/main/resources/lib`.
- `-args, --arguments`
 - Tous les arguments à passer à la classe principale. Si vous voulez passé un argument qui commencer par '-' échapper le caractère '-' avec deux " comme ceci : `-args \-argument_pour_le_main`.
 - Nombre de d'argument : `unlimited`.
- `-m, --main`
 - Classe principale à lancer. Si vous voulez lancer une classe qui se trouve dans un package, il faut spécifier le package avec le nom de la classe comme ceci : `'package.nom.MainClass'`
 - Nombre de d'argument : 1.
- `-e, --encoding`
 - Permet de changer l'encodage des fichiers java à compiler.
 - Nombre de d'argument : 1.
 - Valeur par défaut : `UTF-8`.
- `-exp, --export`
 - Permet de créer un fichier jar exécutable permettant de lancer le projet sans avoir installer MavenLite.
 - Nombre de d'arguments : de 0 à 1.
 - Valeur par défaut : `run.java`.
- `-mvn, --maven`
 - Convertie le projet en projet maven en créant un fichier pom.xml et en déplaçant les fichiers si nécessaire.
 - Nombre de d'argument : 0.
- `-V, --version`
 - Affiche la version.
 - Nombre de d'argument : 0.
 - Valeur par défaut : `2.0.0`.
- `-h, --help`
 - Affiche l'aide et quitte.
 - Nombre de d'argument : 0.
- `-cle, --clear`
 - Permet de supprimer les fichiers dans le dossier de sortie des fichiers compilés.
 - Nombre de d'argument : 0.

Utilisation de Maven Lite

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

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

Ligne de commande

La ligne de commande est la méthode classique pour utiliser Maven Lite bien qu'elle soit moins pratique que l'utilisation d'un fichier de configuration.

Vous pouvez mettre autant d'options que vous le souhaitez dans n'importe quel ordre.

Possibilités de la ligne de commande

- Elle ne prend pas en charge les commentaires.
- Il est possible de mettre des options en utilisant le format `--option` ou `-o`. Vous pouvez mettre n'importe quel option cité dans [la liste des options](#).
- Il est possible de passer des arguments avec des espaces à la class main de votre projet en utilisant des cotes simples plus des guillemets, par exemple `mvnl --args "mon argument"`.
- Il est possible de mettre plusieurs options sur la même ligne en les séparant par un espace, par exemple `mvnl --quiet --verbose` ou `mvnl -q -v`.
- Il est possible d'échapper les caractères spéciaux avec un antislash `\`, par exemple `--args "exemp\le"`. Il est fortement recommandé d'utiliser le système côte simple + guillemets pour passer des arguments à la class main de votre projet avec l'option `-args` et `--arguments` pour éviter tous problèmes d'échappement.

- Les caractères spéciaux dans la ligne de commande sont : \, "
 - `--args "exemp\"le"` devient `exemp"le`
 - `--args "-exemple"` devient `-exemple`
 - `--args "--exemple"` devient `--exemple`
 - `--args "exemp\\le"` devient `exemp\le`
 - `--args "exemp\\\le"` devient `exemp\"le.`

Fichier de configuration

L'utilisation d'un fichier de configuration se fait avec l'option `--file` ou `-f`.

Le fichier de configuration est un fichier unique à chaque projet qui permet de configurer les options que Maven Lite devra utiliser.

Le nom par défaut du fichier de configuration est `LPOM.conf` et doit être à la racine du projet. Il est possible de le renommer mais dans ce cas il faudra préciser son nom l'ors de l'utilisation de l'option, par exemple `mvnl -f monFichier`.

Possibilités du fichier de configuration

- Il est possible de mettre des commentaires en utilisant le caractère `#` au début de la ligne.
- Il est possible de mettre des options en utilisant le même format que dans la ligne de commande, par exemple `--option` ou `-o`.
- Il est possible de passé des arguments avec des espaces en utilisant des guillemets, par exemple `mvnl -args "mon argument"`.
- Il est possible de mettre plusieurs options sur la même ligne en les séparant par un espace, par exemple `mvnl --quiet --verbose` ou `mvnl -q -v`.
- Il est possible d'échapper les caractères spéciaux avec un antislash `\`, par exemple `--args exemp\"le`.
 - Les caractères spéciaux dans le fichier de configuration sont : \, " et - uniquement s'il sont au début de l'argument et que ce dernier n'a pas de guillemet. Par exemple :
 - `--args exemp\"le` devient `exemp"le`
 - `--args \-exemple` devient `-exemple`
 - `--args "--exemple"` devient `--exemple`
 - `--args \--exemple` devient `--exemple`
 - `--args exemp\\le` devient `exemp\le`
 - `--args exemp\\\le` devient `exemp\"le.`
- Une option et ses arguments peuvent être sur plusieurs lignes, par exemple

```
--args
"mon argument

sur plusieurs lignes"
```

- Un exemple type de fichier de configuration est le suivant :

```
# Source du projet
--source src/main/java

# Dossier de sortie des fichiers compilés
--target target

# Liste des libraries à ajouter au classpath
--libraries src/main/resources/lib

# Affiche les commandes exécutées
--verbose
```

create

Il est impossible de créer un projet avec un espace dans le nom.

mvc

compilation

launch

compile-launch

launch-compile

quiet

verbose

exclude

jar

source

target

classpath

libraries

arguments

main

encoding

export

maven

version

help

clear

Exemple, fonctionnalisés et limites