

Maven™





maven



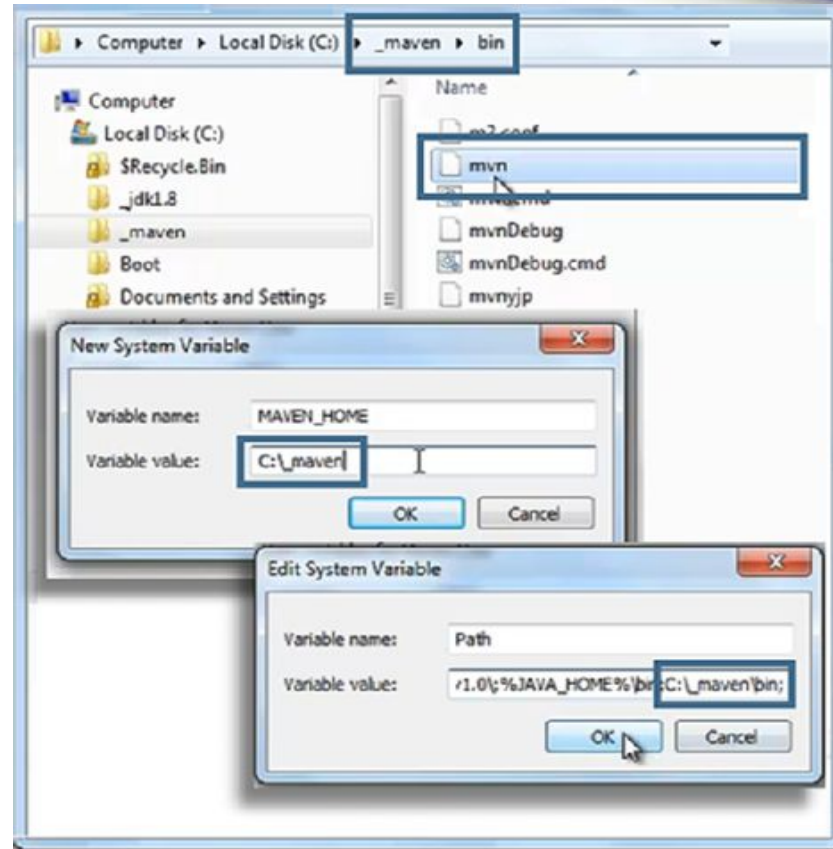
**Maven est un outil open source
de la communauté Apache qui
permet d'automatiser la gestion
et la construction d'un projet
Java.**

le processus d'installation





1. Télécharger l'archive sur le site
<http://maven.apache.org/download.html>
 2. Décompresser l'archive dans un répertoire du système
 3. Créer 2 variable d'environnement
M2_HOME Maven_Home qui pointe sur le répertoire contenant Maven
 4. Ajouter le chemin
apache_maven/bin à la variable
PATH du système
- Pour vérifier l'installation, il faut lancer la commande `mvn -version`





Fonctionnement de Maven



il découpe le cycle de construction du projet en phases prédéfinies et le développeur peut paramétrer ou ajouter des tâches à effectuer automatiquement pour chacune des phases.

Les principales phases dans Maven sont :

compile : compilation du code source du projet;

test : exécution des tests;

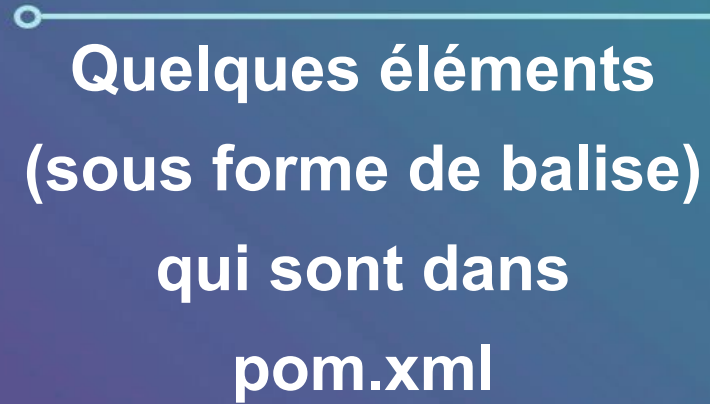
package : construction du livrable (pour une application Web, il s'agit de l'archive WAR).



Un Fichier pom.xml



Le fichier pom.xml est le descripteur de projet pour Maven. Il s'agit d'un fichier XML présent à la racine du projet qui est lu par Maven pour lui fournir les informations du projet.

The logo for Maven, featuring the word "maven" in a bold, lowercase sans-serif font. The letter "a" is orange, while the other letters are black. Below the text is a stylized feather with a black shaft and a fan of red and yellow barbs.



Le nom du projet

`<artifactId>hello</artifactId>`

La version du format du fichier pom.

Actuellement

`<modelVersion>4.0.0</modelVersion>`

La version de notre projet.

`<version>0.0.1-SNAPSHOT</version>`

Le type de packaging, c'est-à-dire le type de projet.

Ici, on indique à Maven que le projet doit être packagé sous la forme d'un WAR. Donc pour Maven, il s'agit d'une application Web.

`<packaging>war</packaging>`

Le group ID de l'application.

`<groupId>fr.epsi.b3</groupId>`

Les propriétés de notre projet. On peut définir des propriétés spécifiques au projet ou des propriétés standard à Maven pour paramétrer la construction du projet.

`<properties>`

Repository Maven et 3 types de Repository Maven



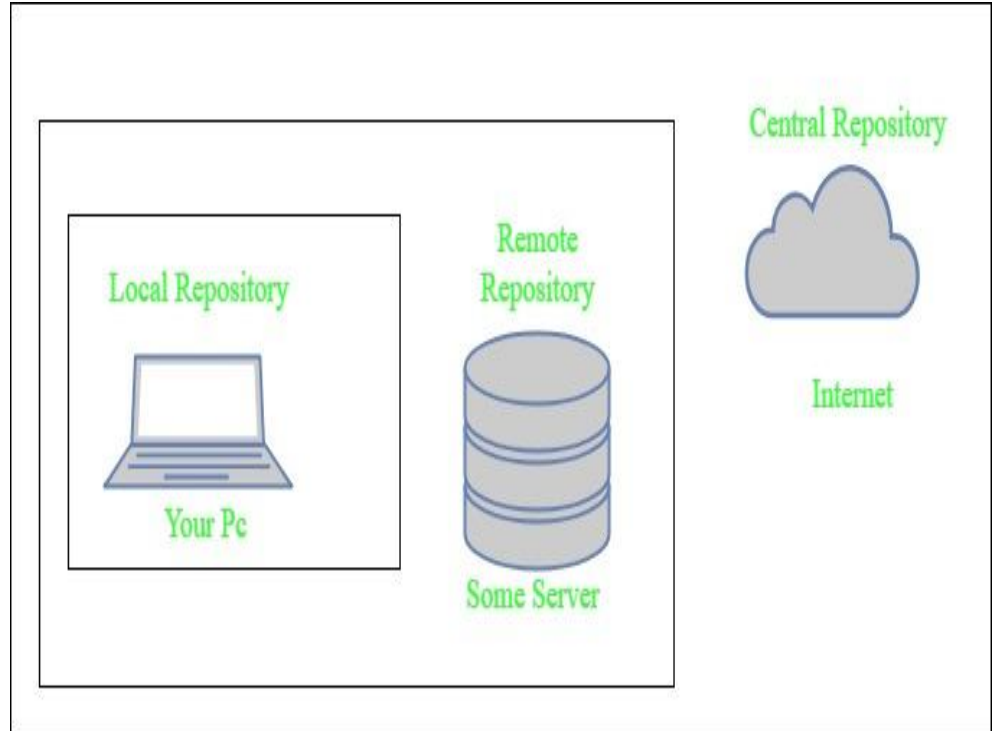


Le repository Maven est un repère de stockage qui contient les artefacts de construction et les dépendances des différents types.

Référentiel local - Un référentiel local est un répertoire sur la machine du développeur.

Référentiel central - Le référentiel Maven central est créé par la communauté Maven.

Référentiel distant - le référentiel distant est un référentiel sur un serveur Web à partir duquel Maven peut télécharger des dépendances.



Maven Life Cycle



maven

Afin d'automatiser la construction d'un projet, Maven s'appuie sur des cycles de vie de construction appelés build lifecycle dans le jargon de Maven.

Il y a 3 build lifecycles de base dans Maven :

- default : qui permet de construire et déployer le projet
- clean : qui permet de nettoyer le projet en supprimant les éléments issus de la construction de celui-ci
- site : qui permet de créer un site web pour le projet

Ces build lifecycles sont découpés en phases qui sont exécutées séquentiellement les unes à la suite des autres.

Gestion des dépendances

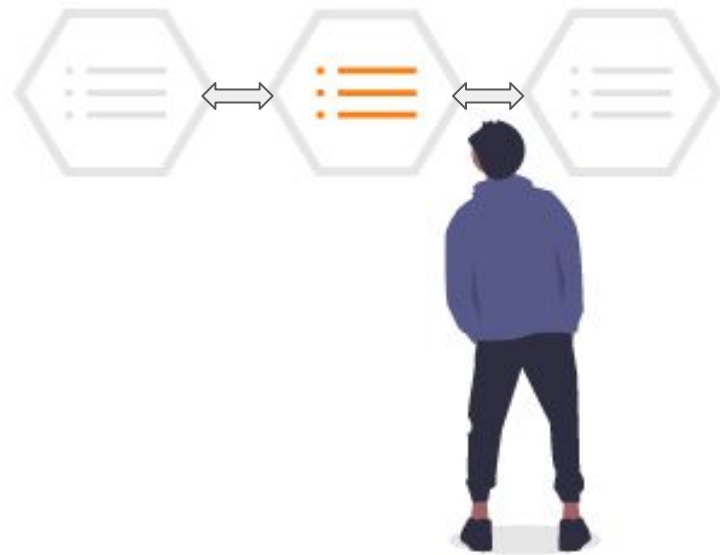
La gestion des dépendances de Maven repose sur plusieurs concepts :

les dépôts : permet de stocker les artéfacts

la portée : permet de préciser dans quel contexte une dépendance est utilisée

la transitivité : permet de gérer les dépendances de dépendances

l'héritage



Quelques commandes maven

maven -v: pour la version

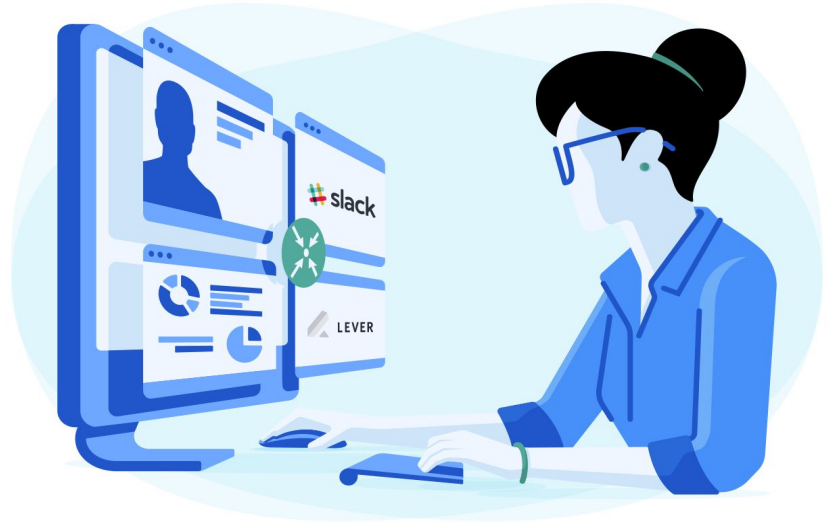
maven java:compile: pour la compilation

maven -g: pour l'obtention de la liste
complète des plugins à disposition de
maven

maven -clean: effacer tous les fichiers
générés par maven.

IDE

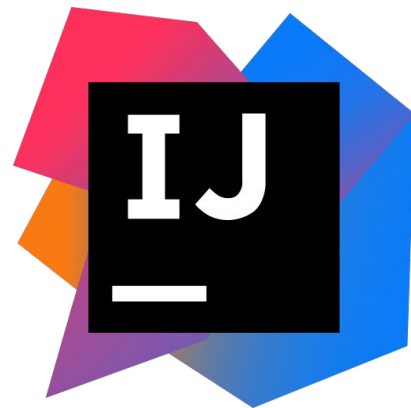
Integrated Development Environment



maven



Visual Studio Code



Apache
NetBeans IDE

jGRASP



Les Types de Projets

Maven est utilisé principalement pour les projets [Java](#) . Maven peut également être utilisé pour créer et gérer des projets écrits en [C#](#) , [Ruby](#) , [Scala](#) et d'autres langages



Avantages et Inconvénients





Avantages

1. Maven peut ajouter automatiquement toutes les dépendances requises pour le projet en lisant le fichier pom.
2. On peut facilement construire leur projet en pot, en guerre, etc. selon leurs besoins en utilisant Maven.
3. Maven permet de démarrer facilement un projet dans différents environnements et il n'est pas nécessaire de gérer l'injection de dépendances, les builds, le traitement, etc.
4. L'ajout d'une nouvelle dépendance est très simple. Il suffit d'écrire le code de dépendance dans le fichier pom.



1. Maven a besoin de l'installation de maven dans le système pour fonctionner et du plugin maven pour l'ide.
2. Si le code maven d'une dépendance existante n'est pas disponible, vous ne pouvez pas ajouter cette dépendance à l'aide de maven.

Groupe 1:

Fatimata	DIAGOURAGA
Hilaire	DALIWA
Thomas	CISSE
Modibo	SANGARE
Amagaraï	GUINDO
Ousmane	KANE
Ibrahima	KONATE
Aïssata	SANKARE

