|  |  |
| --- | --- |
| https://www.jmdoudoux.fr/java/dej/images/maven002.png | Maven est un outil de construction de projets (build) open source développé par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine. Il permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java. |

Le site web officiel est [http://maven.apache.org](http://maven.apache.org/)

Il permet notamment :

* d'automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet
* de gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet
* de générer des documentations concernant le projet

Au premier abord, il est facile de croire que Maven fait double emploi avec Ant. Ant et Maven sont tous les deux développés par le groupe Jakarta, ce qui prouve bien que leur utilité n'est pas aussi identique que cela. Ant, dont le but est d'automatiser certaines tâches répétitives, est plus ancien que Maven. Maven propose non seulement les fonctionnalités d'Ant mais en propose de nombreuses autres.

Pour gérer les dépendances du projet vis-à-vis de bibliothèques, Maven utilise un ou plusieurs dépôts qui peuvent être locaux ou distants.

Maven est extensible grâce à un mécanisme de plugins qui permettent d'ajouter des fonctionnalités.

Pour assurer la construction d'un projet, Maven propose notamment de prendre en charge :

* La compilation
* Le packaging
* La gestion des dépendances
* La génération de la documentation
* L'accès au gestionnaire de sources
* L'accès aux dépôts ou aux gestionnaires de dépendances
* Le déploiement
* ... et de très nombreuses autres tâches requises lors d'un build

Maven impose par défaut l'emploi de conventions notamment dans la structuration du projet.

Maven utilise une approche déclarative où la structure du projet et son contenu sont décrits dans un document XML. De plus il convient de se conformer à une structure de projets standards et de bonnes pratiques. L'observation de ces normes permet de réduire le temps nécessaire pour écrire et maintenir les scripts de build car ils sont tous structurés de la même façon.

La description d'un projet est faite dans un fichier XML nommé POM (Project Object Model). Cette description contient notamment les dépendances, les spécificités de construction (compilation et packaging), éventuellement le déploiement, la génération de la documentation, l'exécution d'outils d'analyse statique du code, ...

Maven peut aussi assurer de nombreuses autres tâches car il est conçu pour utiliser des plugins : il est donc extensible. Maven est fourni avec un grand nombre de plugins standard mais il est aussi possible d'utiliser d'autres plugins qui sont stockés dans les dépôts voire même de développer ses propres plugins.

**Installation**

Windows Il faut télécharger le fichier .exe sur le site de Maven et l'exécuter.

### La création d'un nouveau projet

mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes

-DarchetypeArtifactId=maven-archetype-quickstart

-DgroupId=com.jmdoudoux.test.monapp -DartifactId=monApplication -DinteractiveMode=false

Cette commande va créer un répertoire pour contenir le projet et un ensemble de sous-répertoires et de fichiers selon l'archétype utilisé. Parmi les fichiers générés, il y a notamment le fichier pom.xml.

Les sous-répertoires créés suivent la structure standard des répertoires d'un projet qui est définie par convention.

Le sous-répertoire src/main/java contient les sources du projet.

Le sous-répertoire src/test/java contient les sources des tests unitaires du projet.

Lors de la création du premier archétype, Maven va télécharger dans son dépôt local tous les plugins qui lui sont nécessaires.

Un archétype est un template de projet qui est combiné avec certaines informations fournies pour créer un projet Maven d'un type particulier.

### Le fichier POM

Le fichier POM (Project Object Model) contient la description du projet Maven. C'est un fichier XML nommé pom.xml. Il contient les informations nécessaires à la génération du projet : identification de l'artéfact, déclaration des dépendances, définition d'informations relatives au projet, ...

Le fichier POM doit être à la racine du répertoire du projet.

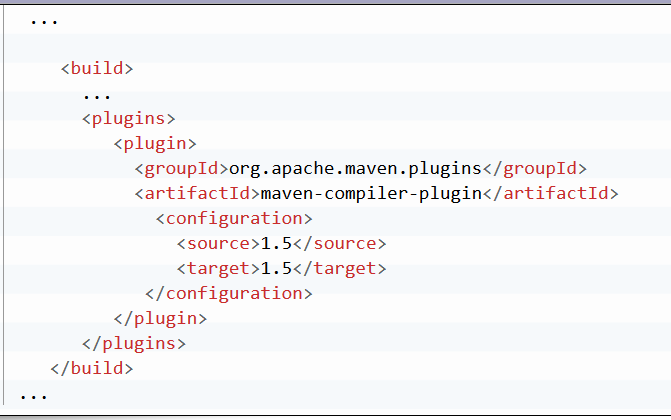


#### 

#### 

#### L'utilisation et la configuration des plugins

Maven possède de très nombreux plugins fournis par Maven ou par des tiers. Il est aussi possible d'écrire ses propres plugins.



### Le cycle de vie d'un projet

* default : il permet de générer et déployer l'artéfact du projet
* clean : il permet de nettoyer les fichiers générés du projet
* site : il permet de générer un site web pour accéder à la documentation du projet

Un cycle de vie est composé de phases qui constituent les étapes de la génération de l'artéfact.

#### La déclaration des dépendances

Le tag <dependencies> du fichier POM permet de déclarer les dépendances externes du projet que ce soit pour la compilation, les tests ou l'exécution.

Pour chaque dépendance, il est nécessaire de préciser le groupId, l'artifactId, la version et le scope. Les trois premières informations doivent correspondre à celles définies dans le POM de la dépendance.