

Problématique

Le déploiement d'applications est devenu aujourd'hui un vrai casse-tête. En effet, à chaque phase du projet, les développeurs doivent gérer un environnement différent : intégration, recette, pré-production, production. Chaque environnement possède ses propres caractéristiques : adresses IP, serveurs de bases de données, etc.

Aujourd'hui, les applications sont la plupart du temps générées à l'aide de scripts Ant, de shell scripts ou même à la main. Cela revient souvent à faire des multitudes de copier/coller de scripts et à les réadapter à chaque projet.

Origines du besoin

Initialement, Maven a été créé comme outil d'aide au développement d'un autre projet Jakarta : **Turbine**. Les développeurs sont en effet partis du constat que les différents projets Jakarta possédaient chacun leurs propres fichiers de compilation Ant. Les développeurs voulaient donc une façon standard de compiler les projets, une façon simple de publier les informations du projet et une façon de partager les JAR entre projets.

À quoi sert Maven ?

Maven est un outil permettant d'automatiser la gestion de projets Java. C'est un outil open-source de build pour les projets Java très populaire, conçu pour supprimer les tâches difficiles du processus de build. Maven utilise une approche déclarative, où le contenu et la structure du projet sont décrits, plutôt qu'une approche par tâche utilisée par exemple par Ant ou les fichiers Make traditionnels. Cela aide à mettre en place des standards de développements au niveau d'une société et réduit le temps nécessaire pour écrire et maintenir les scripts de build.

Il offre entre autres les fonctionnalités suivantes :

- Compilation et déploiement des applications Java (JAR, WAR)
- Gestion des bibliothèques requises par l'application
- Exécution des tests unitaires

- Génération des documentations du projet (site web, pdf, Latex)
- Intégration dans différents IDE (Eclipse, JBuilder)

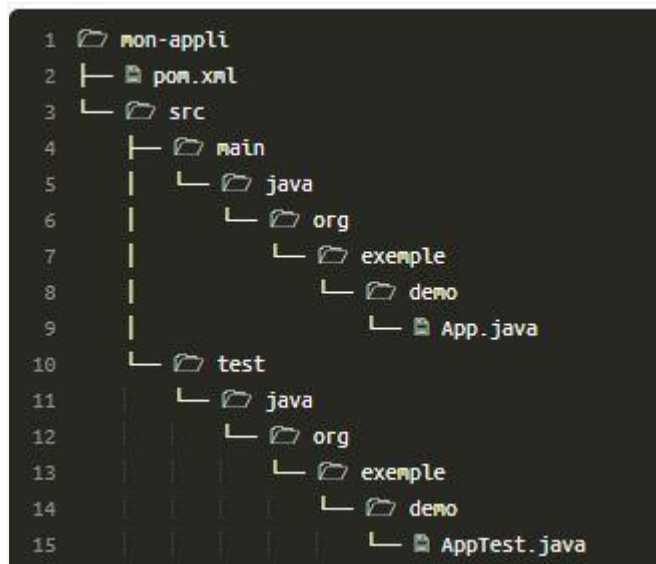
Maven impose une arborescence et un nommage des fichiers du projet selon le concept de Convention plutôt que configuration. Ces conventions permettent de réduire la configuration des projets, tant qu'un projet suit les conventions. Si un projet a besoin de s'écarter de la convention, le développeur le précise dans la configuration du projet.

Structure du répertoire Maven

Une partie de la puissance de Maven vient des pratiques standardisées qu'il encourage. Un développeur qui a déjà travaillé sur un projet Maven se sentira tout de suite familier avec la structure et l'organisation d'un autre projet Maven. Il n'y a pas besoin de gaspiller du temps à réinventer des structures de répertoires, des conventions, et à adapter des scripts de build Ant pour chaque projet. Même s'il est possible de redéfinir l'emplacement de chacun des répertoires vers une localisation spécifique, il est réellement intéressant de respecter la structure de répertoire standard de Maven autant que possible, et ce pour plusieurs raisons :

- Cela rend le fichier POM plus court et plus simple
- Cela rend le projet plus simple à comprendre et rend la vie plus simple au pauvre développeur qui devra maintenir le projet quand vous partirez.
- Cela rend l'intégration de plug-ins plus simple

La structure standard de répertoire de Maven est illustrée dans la figure 1. Dans le répertoire de base se trouve le POM (pom.xml) et deux sous répertoires : src pour tout le code source.



Le répertoire `src` contient plusieurs sous-répertoires, chacun avec une utilité précise :

- un répertoire **`src/main/java`**. C'est le répertoire contenant les sources de mon application, contenant pour l'instant un unique fichier `App.java` situé dans le package `org.exemple.demo`.
- un répertoire **`src/test/java`**. C'est le répertoire contenant les sources des tests JUnit de mon application, contenant pour le moment un unique fichier `AppTest.java` situé dans le package `org.exemple.demo`.

Conclusion

Maven est un outil très puissant qui simplifie et standardise énormément le processus de build. En favorisant une organisation standard des projets et des bonnes pratiques reconnus, Maven ôte une bonne quantité de travail. Et des plugs-in standards, comme le générateur de site web, offrent des outils collaboratifs intéressants avec peu d'effort. La dernière version en ligne de maven est la 3.6.1 (4 avril 2019). Vérifiez par vous-même !