*Organisez et packagez une application Java avec Apache Maven*

**Chapitre 1 : Organisez votre projet**

Maven est un outil de gestion de projet logiciel pour Java maintenu par Apache. Il facilite la compilation du code mais aussi et surtout la gestion des dépendances et des bibliothèques tierces.

1. **Configurez votre environnement de développement :**

La configuration de Maven est enregistrée dans un fichier xml « settings.xml » à créer dans le répertoire « /.m2 » situé dans le homdir de l’utilisateur. Les dépendances, elles, seront téléchargées et gérées depuis le même endroit mais dans le répertoire « /.m2/repository » par defaut. Il est possible de changer le répertoire où sont téléchargées les dépendances en modifiant le fichier de settings.

1. **Créez votre premier projet Maven :**

Maven utilise une approche qu’on appelle « de convention plutôt que de configuration », autrement dit son fonctionnement s’appuie sur un certain nombre de conventions qui rendent le développement plus facile en en automatisant une bonne partie.

Pour l’arborescence du projet, par exemple, Maven s’appuie sur des « archétypes » (sortes de modèles prédéfinis) et celui utilisé dans ce cours sera « Quickstart ».

Pour créer un projet Maven on utilise la commande « mvn archetype :generate » avec l’archétype et/ou d’autres options passées en paramètres. Un fichier de configuration « pom.xml » est généré ainsi que le répertoire « src » contenant celui du « main » et celui réservé aux tests.

1. **Construisez votre nouveau projet :**

Un programme Java se compile sous la forme d’un fichier « .jar » qui est en fait une simple archive exécutable contenant toutes les classes du programme et ses fichiers de configuration. Maven s’occupe de compiler notre application avec la commande « mvn pakage » (à effectuer depuis la racine du projet là où se trouve le pom.xml). En réalité Maven fait même plus que ça… Il compile mais exécute également les classes de tests, par exemple !

1. **Décrivez votre projet :**

Avec Maven, tout ou presque se passe dans le fichier « pom.xml » (pour Project Object Model). Beaucoup d’informations présentes dans le « pom.xml » sont facultatives mais il est bon de prendre l’habitude de les renseigner car elles serviront à Maven qui est capable de générer automatiquement un site descriptif du projet.

* C’est dans ce fichier qu’il est indiqué, par la présence ou nom du terme « SNAPSHOT » dans le numéro de version, selon si ladite version de l’application est en cours de développement ou si c’est une version release.
* Les propriétés sont comme des constantes qui seront, à l’exécution du code, remplacées par leur valeur définie dans le « pom.xml ». En réalité ce fichier se base sur un « super pom » qui contient donc d’autres propriétés qui ne sont donc pas forcément surchargées dans le « pom.xml ».
* Il est également possible de définir des propriétés dans un fichier ressources à part (enregistré dans « src/main/resources ») et de le ‘linker’ dans le fichier « pom.xml »
* On peut aussi influencer sur le déroulement du build directement depuis le fichier « pom.xml »

Maven propose également des profils qui permettent un comportement différent au build, selon qu’on soit sous tel ou tel profil.