MAVEN

Table des matières

1. Les rôles

https://fr.wikipedia.org/wiki/Gestionnaire_de_paquets

https://maven.apache.org/index.html

- ✓ Construction, compilation ;
- ✓ Documentation ;
- ✓ Rapport ;
- ✓ Gestion des dépendances ;
- ✓ Gestion des sources ;
- ✓ Mise à jour de projet ;
- ✓ Déploiement.

2. Les différents éléments

- ✓ Le fichier POM (Projet Object Model) : contient tous les éléments permettant de gérer le cycle de vie du projet.

- ✓ Archétype : Un archétype est un Template de projet. Le fait d'utiliser des archétypes pour initialiser un projet permet de gagner du temps et de respecter une certaine convention.
  https://mavefrn.apache.org/archetypes/

- ✓ GroupId/artifactId :

- Le groupId est l'identifiant du groupe, à l'origine du projet.
  Le GroupId suit les mêmes règles de nommage que les packages Java (exemple : fr.masociete.monprojet), et on choisit généralement comme groupId le nom du top package du projet.

- L'artifactId est l'identifiant du projet au sein de ce groupe.
  L'artifactId est utilisé par défaut pour construire le nom de l'artefact final (exemple : pour un artifactId=monprojet, le nom du fichier jar généré sera monprojet-version.jar).

✓ Artefact : Dans Maven, un artefact est un élément spécifique issu de la construction du logiciel. Il s'agit des dépendances et des plugins.
Il est composé d'un GrouId, d'un ArtifactId et d'un numéro de version

✓ Dépendance : Une dépendance est une référence vers un artefact spécifique contenu dans un repository. Cet artefact est nécessaire pour une ou plusieurs phases du cycle de vie du projet.

✓ Plugin : permet d'avoir des fonctionnalités de base et d'en ajouter https://maven.apache.org/plugins/index.html

✓ SNAPSHOT : version du projet

✓ Repository local : répertoire sur le poste du développeur permettant de stocker, suivant la même arborescence, tous les artefacts téléchargés depuis le(s) repository distant(s).

✓ Repository distant : Maven central (https://mvnrepository.com/repos/central).
Répertorie les dépendances et les plugins utilisables par Maven :
https://mvnrepository.com/artifact
(Ex : https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-maven-plugin)

3. Le Cycle de vie

Le Cycle de vie des builds (build lifecycle) est l'enchaînement de phases, découpées en tâches

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

Exemple du build lifecycle default

- ✓ validate : vérifie que la configuration projet est correcte (POM, pas d'éléments manquants...)
- ✓ compile : compile les sources du projet
- ✓ test : teste le code compilé avec les classes de tests unitaires contenues dans le projet
- ✓ package : package les éléments issus de la compilation dans un format distribuable (JAR, WAR...)
- ✓ install : installe le package dans votre repository local
- ✓ deploy : envoie le package dans le repository distant défini dans le POM

Les phases sont découpées en tâches (appelées goals), chacune étant assurée par un plugin Maven.

4. Installation

https://maven.apache.org/download.cgi

Binary zip archive sous Windows

Créer un dossier Maven et y mettre le zip

Dézipper

Placer ce dossier sous c:\\Programmes (où se trouve Java)

Dans les variables d'environnement :

Ajouter la variable JAVA_HOME avec chemin vers java : C:\chemin\vers\repertoire\env\java\jdk21. (Version de votre JDK)

Modifier la propriété Path : ajouter le chemin vers le fichier bin d'apache :

C:\chemin\vers\repertoire\env\maven\apache-maven-Version\bin

5. Configuration

Prendre le fichier settigns.xml se trouvant dans le dossier conf de Maven et le copier dans le répertoire home\.m2 (c:\\utilisateurs\...)

Pour avoir le Repository dans le même dossier que les projets :

Créer dans votre dossier Maven un sous dossier Repository

Modifier le fichier settings.xml à l'endroit du <localRepository> en y mettant votre chemin de dossier

6. Créer un projet Maven

Dans votre dossier Maven

https://maven.apache.org/archetypes/maven-archetype-quickstart/

groupId : pays.societe.filiale.département

Ex : fr.afpa.pompey.cda08.demo

artifactId : nomDuProjet

SNAPSHOT : laisser vide

Package : laisser vide

Valider : taper y

Se mettre sur le dossier crée avec le nom de votre projet, ne pas rester sur le dossier où vous avez lancé la génération de l'archetype

Créer le package : en mode commande sous le dossier du projet : mvn package

Puis lancer :

java -cp target/nomDuProjet-1.0-SNAPSHOT.jar pays.societe.filiale.département.App

➔ ne pas mettre l'actifactId avant .App

-cp ou classpath (le classpath permet de préciser au compilateur et à la JVM où elle peut trouver les classes requises par l'application)

Modifier dans le plugin de génération du .jar

```
<plugin>
        <artifactId>maven-jar-plugin</artifactId>
        <version>3.0.2</version>
        <configuration>
          <archive>
            <!-- Création du Manifest pour la définition de la classe Main -->
            <manifest>
              <mainClass>${project.mainClass}</mainClass>
            </manifest>
          </archive>
        </configuration>
```

```
    </plugin>
```

Ajouter le plugin permettant de lancer le projet avec mvn exec:java (nom du goal)

```
<plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <executions>
          <execution>
            <goals>
              <goal>java</goal>
            </goals>
          </execution>
        </executions>
        <configuration>
          <mainClass>${project.mainClass}</mainClass>
        </configuration>
      </plugin>
```

Dans properties :

```
<project.mainClass>groupID.App</project.mainClass>
```

Pour lancer le projet

mvn exec:java

Mettre un projet dans le repository local

mvn install

Si pb de cache : mvn clean install

Ajouter une dépendance à un projet

Ajouter dans le pom.xml :

```
<dependency>
    <groupId>${project.groupId}</groupId>
    <artifactId>classesMetier</artifactId>
    <version>${project.version}</version>
  </dependency>
```

7. Diverses commandes Maven

Arrêter le programme de commande : Ctrl c

Package : mvn package

Compilation : mvn compile

Dépendances d'un projet : mvn dependency:tree

Liste des archetypes de maven.aparche.org : Apache Maven Archetypes – Maven Archetypes

Voir tous les archetypes disponibles : mvn archetype:generate

Trouver le quickstart de maven :

Choose a number or apply filter (format : [groupId:]artifactId, case sensitive contains): (number): maven:quickstart

Choose a number or apply filter (format : [groupId:]artifactId, case sensitive contains): 6:

(Le numéro affiché est le numéro conseillé)

Dépendances

Trouver une dépendance : taper maven central + nom de la dépendance

Exemple : maven central log4j

Vous récupérez l'URL : https://mvnrepository.com/artifact/log4j/log4j

En général, prendre la dernière version de Central, cliquez dessus et copiez dans votre pom.xml