Compiler et construire ses applications avec maven

Open in Visual Studio Code

Ce TP vise à être capable de compiler un fichier Java sans être dépendant de son IDE favori. Cela permet :

- d'être indépendant de tout IDE
- de faire construire l'application de manière automatique (c'est pour plus tard, mais c'est ce qu'on appelle l'intégration continue)

Mise en place

Les illustrations utilisent VS Code, mais peu importe que vous utilisiez un autre IDE comme Eclipse ou IntelliJ, il vous suffit d'adapter à votre IDE (qu'il vous faut apprendre à connaître).

C'est parti

1ère compilation : erreur de comilation

Les étapes suivantes utilisent les commandes Maven en ligne de commande, c'est à dire à taper dans un terminal linux, mais vous pouvez utiliser l'équivalent en "click de souris" de votre IDE favori.

1. Lancer `mvn clean`

   C'est une bonne habitude à prendre (de nettoyer tout ce qui a pu se passer avant).

2. Lancer `mvn compile`

   Observez l'erreur de compilation qui en résulte. Par exemple :

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.040 s
[INFO] Finished at: 2022-02-20T16:04:15+01:00
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.8.1:compile (default-compile)
on project tp_qualite: Compilation failure
[ERROR] /Users/jmb/localdev/teaching/2022/qualite-r2.03/td4-maven/src/main/java/HelloJava.java:[3,16] invalid
method declaration; return type required
```

Figure 1. Erreur de compilation

3. Corrigez le code en conséquence.

```
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ tp_qualite ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/jmb/localdev/teaching/2022/qualite-r2.03/td4-maven/target/classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.085 s
[INFO] Finished at: 2022-02-20T16:12:52+01:00
[INFO] ------------------------------------------------------------------------
```

Figure 2. Erreur corrigée

4. Observez la création du répertoire `target` qui contient entre autre la version compilée `HelloJava.class` dans le répertoire `target/classes`.

2ème compilation : construire une application

On ne peut malheureusement pas exécuter d'application puisqu'on n'a pas de main.

1. Ajoutez un `main` dans la classe.

```
class HelloJava {
        public static void main(String[] args) {
        System.out.println("Hello Blagnac");
    }

    public void afficherCancan(){
    }
}
```

2. Une fois que vous avez réussi à compiler (`mvn compile`), lancez la fabrication d'une version exécutable :

```
mvn package
```

3. Observez la création du fichier JAR et testez-le :

```
java -jar target/tp_qualite-1.0.jar
```

```
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ tp_qualite ---
[INFO] Building jar: /Users/jmb/localdev/teaching/2022/qualite-r2.03/td4-maven/target/tp_qualite-1.0.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.257 s
[INFO] Finished at: 2022-02-20T16:24:35+01:00
[INFO] ------------------------------------------------------------------------

~/localdev/teaching/2022/qualite-r2.03/td4-maven » java -jar target/tp_qualite-1.0.jar                jmb@mbp-jmb
Hello Blagnac
```

Figure 3. Et voilà!

Améliorations

1. Reprenez un de vos exercices précédents et mettez les sources dans `src/java`.
2. Ajustez éventuellement le `pom.xml` pour que les étapes précédentes produisent les résultats escomptés. Pensez à vérifier que votre fichier .jar est exécutable. Que devez-vous corriger dans votre pom.xml ? Quelle est la commande pour le lancer ?
3. Ajoutez les commentaires vus en dev.
4. Cherchez dans la documentation Maven les commandes permettant de générer une documentation javadoc, et les adaptions à faire sur votre projet (et éventuellement `pom.xml`) afin que Maven génère la documentation automatiquement pour vous.
5. Documentez votre projet: remplacez le contenu de ce fichier README.doc par les instructions de compilation, de documentation et de lancement de votre projet.

Consignes et rendus

Pour ce TP, il vous faudra simplement rendre le projet complet (`src`) ainsi que la dernière version du fichier `pom.xml` sur votre dépôt et la javadoc de votre code.

Appendix A: Import dans Eclipse

TO BE DONE...