« Les plugins »

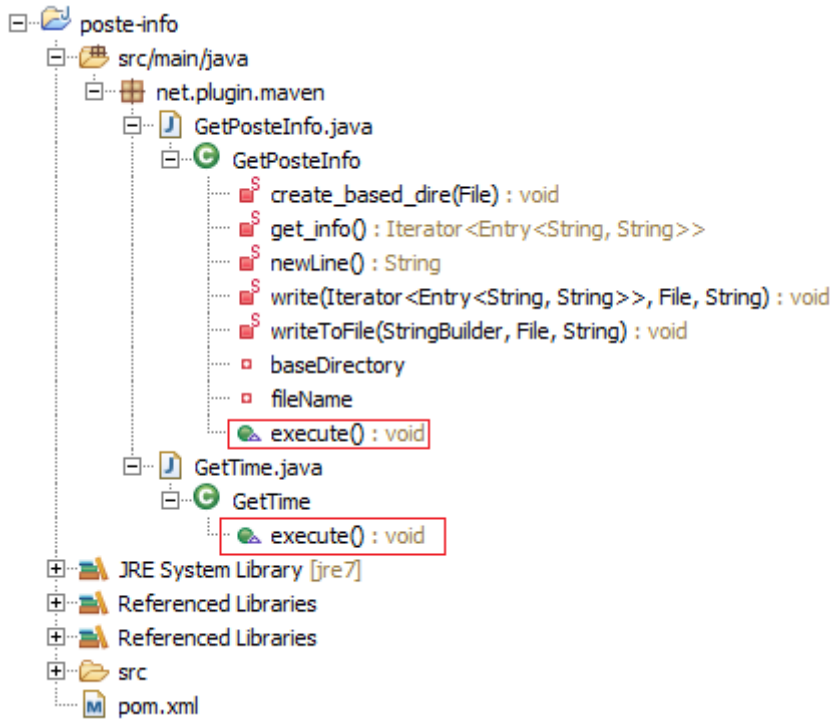
Maven7

Table des matières

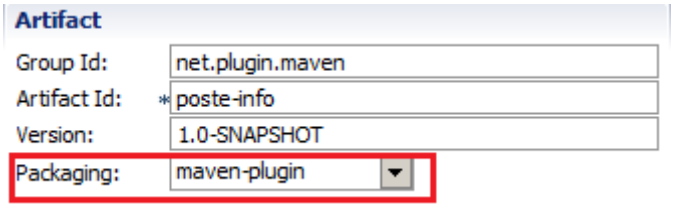
I- Description de l'arborescence du projet :

Après avoir importé le fichier .ZIP poste_info.zip dans le workspace :

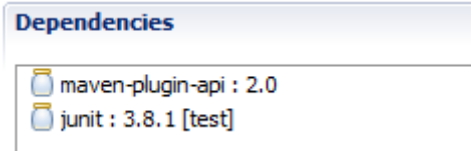
On note principalement que la structure d'un projet « poste_info » doit être comme suit :



La balise Packaging aura comme valeur maven-plugin :



La balise dependencies contiendra les deux dépendances :



La balise Build est comme suit :

```
<build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-plugin-plugin</artifactId>
        <version>3.2</version>
        <configuration>
          <goalPrefix>poste-info</goalPrefix>
          <skipErrorNoDescriptorsFound>true</skipErrorNoDescriptorsFound>
        </configuration>
        <executions>
          <execution>
            <id>mojo-descriptor</id>
            <goals>
              <goal>descriptor</goal>
            </goals>
          </execution>
          <execution>
            <id>help-goal</id>
            <goals>
              <goal>helpmojo</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
 </build>
```

II-Objectif du projet :

Le projet présente deux exemples classiques de plugin :

1- Le premier permet d'afficher l'heure courante
2- Le deuxième permet d'écrire dans un fichier info.txt plusieurs informations système :
   Les valeurs des variables systèmes : JAVA_HOME, MAVEN_HOME et Path.
3- On souligne que pour exécuter un plugin on utilise l'annotation `@goal nameplugin` :

```
/**
 * @goal GetTime
 */
```

4- La classe du plugin doit hériter de la classe mére : `AbstractMojo`

```
public class GetTime extends AbstractMojo
```

5- Plusieurs import sont indispensable :

```
import java.util.Date;
import org.apache.maven.plugin.AbstractMojo;
import org.apache.maven.plugin.MojoExecutionException;
import org.apache.maven.plugin.MojoFailureException;
```

6- Une fonction principal qui contient les instruction a exécuter par le plugin : execute()

```
public void execute() throws MojoExecutionException, MojoFailureException
```

7- Pour faire tourner le plugin suivre ce qui suit:

a- Sur le POM exécuter run→Build…
b- Executer la commande maven :

   clean install

c- Une deuxième fois Sur le POM exécuter run→Build…
d- Exécuter la commande suivante :

   groupId: artifactId:nameplugin

III- Fichier du projet :

1- GetTime.java :

```java
package net.plugin.maven;

import java.util.Date;

import org.apache.maven.plugin.AbstractMojo;
import org.apache.maven.plugin.MojoExecutionException;
import org.apache.maven.plugin.MojoFailureException;

/**
 * @goal GetTime
 */
public class GetTime extends AbstractMojo {

	public void execute() throws MojoExecutionException, MojoFailureException {

		Date currentTime = new Date();
		getLog().info("Now is:" + currentTime);
	}

}
```

2- GetPosteInfo.java :

```
package net.plugin.maven;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.util.Iterator;
import java.util.Map;

import org.apache.maven.plugin.AbstractMojo;
import org.apache.maven.plugin.MojoExecutionException;
import org.apache.maven.plugin.MojoFailureException;

/**
 * @goal About_Poste
 */
public class GetPosteInfo extends AbstractMojo{

	/**
	 * @parameter
	 *     default-value="info.txt"
	 *     expression="${environment.filename}"
	 */
	private String fileName;

	/**
	 * @parameter
	 *     default-value="C:\\temp"
	 *     expression="${environment.base_dir}"
	 */
	private File baseDirectory;


	public void execute() throws MojoExecutionException, MojoFailureException {
		Iterator<Map.Entry<String, String>> lentrie=null;

		//création de c:/temp
		create_based_dire(baseDirectory);
		//Obtenir les informations sur le poste : lire la varriable systeme
"JAVA_HOME,MAVEN_HOME,path"
		lentrie=get_info();
		//ecriture dans le fichier info.txt des informations obtenues
		write(lentrie,baseDirectory,fileName);


	}

private static void write(Iterator<Map.Entry<String, String>> le,File bd,String
fn)
{
	StringBuilder fileContents = new StringBuilder();

	fileContents.append("Description de votre poste:" + newLine());
	fileContents.append("" + newLine());

	while (le.hasNext()){

		Map.Entry<String, String> entry = le.next();
```

```
            if (entry.getKey().equalsIgnoreCase("JAVA_HOME"))
            {
                  fileContents.append(entry.getKey() + ":" + entry.getValue() +
newLine());
                  fileContents.append("" + newLine());
            }
            if (entry.getKey().equalsIgnoreCase("MAVEN_HOME"))
            {
                  fileContents.append(entry.getKey() + ":" + entry.getValue() +
newLine());
                  fileContents.append("" + newLine());
            }
            if (entry.getKey().equalsIgnoreCase("Path"))
            {
            fileContents.append(entry.getKey() + ":" + entry.getValue() +
newLine());
            }
            }
      fileContents.append("" + newLine());
      writeToFile(fileContents,bd,fn);
}

private static Iterator<Map.Entry<String, String>> get_info()
{
      Map<String, String> environment = System.getenv();
      Iterator<Map.Entry<String, String>> entries =
environment.entrySet().iterator();
      return entries;
}

private static void create_based_dire(File F)
{
      if (F.exists()){//création du dossier C:\\temp

            F.mkdirs();
      }

}

private static String newLine(){
            return System.getProperty("line.separator");
      }

private static void writeToFile(StringBuilder fileContents,File b,String f){

            FileWriter fWriter = null;
            BufferedWriter bWriter = null;

            try{

                  fWriter = new FileWriter(b + File.separator + f);
                  bWriter = new BufferedWriter(fWriter);
                  bWriter.write(fileContents.toString());

            }catch (Exception e){
                  //ne rien faire
            }finally{

                  try{
```

```
                        if (bWriter != null){
                              bWriter.close();
                        }
                        if (fWriter != null){
                              fWriter.close();
                        }
                  }catch (Exception e){
                        //ne rien faire
                  }
            }
      }
}
```

3- POM :

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>net.plugin.maven</groupId>
  <artifactId>poste-info</artifactId>
  <packaging>maven-plugin</packaging>
  <version>1.0-SNAPSHOT</version>

  <name>poste-info Maven Mojo</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>org.apache.maven</groupId>
      <artifactId>maven-plugin-api</artifactId>
      <version>2.0</version>
    </dependency>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

    <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-plugin-plugin</artifactId>
        <version>3.2</version>
        <configuration>
          <goalPrefix>poste-info</goalPrefix>
          <skipErrorNoDescriptorsFound>true</skipErrorNoDescriptorsFound>
        </configuration>
        <executions>
          <execution>
            <id>mojo-descriptor</id>
            <goals>
              <goal>descriptor</goal>
            </goals>
          </execution>
          <execution>
            <id>help-goal</id>
            <goals>
              <goal>helpmojo</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>

</project>
```