



Tutorial d'utilisation d'un composant MUSE

Tutorial

Référence : T Auteur(s) : Laurence Estrabaut
Version : 1.0
Date : 10/07/2014

TABLE DES MATIERES

1	RETOUR SUR UNE NORME	<u>37</u>
2	PREREQUIS.....	<u>37</u>
3	GENERATION D'UN MUSE	<u>48</u>
3.1	ETAPE PRELIMINAIRE : CONSTRUCTION DU PROJET	<u>48</u>
3.2	GESTION DES DEPENDANCES	<u>610</u>
3.2.1	UTILISER MAVEN POUR EXECUTER LE JAR COMPILE	<u>610</u>
3.2.2	CREER UN JAR INDEPENDANT	<u>812</u>
3.2.3	COPIER LES DEPENDANCES	<u>913</u>

1 RETOUR SUR UNE NORME

Notre constat est que les solutions logicielles actuelles dédiées à la modélisation du bâtiment et des systèmes sont centrées « outils », plutôt que « modèle ». La norme MUSE que nous proposons a pour objectif :

- **Généraliser l'approche « modèle ».** On entend par là aller vers des descriptions normalisées des modèles et des modalités d'interaction avec ces derniers indépendamment des outils de modélisation/simulation et du métier ciblé
- **Capacité d'import/export de modèles.** Donner la possibilité de définir et/ou d'exploiter des modèles s'appuyant sur l'offre existante en modélisation/simulation.

Le présent tutorial a pour but d'expliquer l'usage d'un composant MUSE issu de cette norme.

2 PREREQUIS

Pour suivre le présent tutorial, vous devez avoir installé Maven, outil d'automatisation de production des projets en java, en suivant les informations du site d'Apache : <http://maven.apache.org/download.cgi>

Vous devez également avoir téléchargé les sources de l'exemple que nous utiliserons tout au long de ce tutorial. Vous le trouverez sur le répertoire Archiva de Vesta System :

<http://87.98.133.27:8888/archiva/repository/Muse/org/gu/vesta/muse.sample.basicloader/1.0/muse.sample.basicloader-1.0-sources.jar>

3 GENERATION D'UN MUSE

3.1 Etape préliminaire : Construction du projet

Dans un dossier MuseLoaderExemple :

- Créer un fichier « pom.xml » dont le contenu est le suivant :

```
<project>
  <!--Description du projet -->
  <modelVersion>4.0.0</modelVersion>
  <groupId>G2Elab</groupId>
  <artifactId>LoadMUSE</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <!--Repositories : lieu de stockage des dépendances -->
  <repositories>
    <repository>
      <id>archiva.vesta.muse</id>
      <url>http://94.23.5.135:8888/archiva/repository/Muse/</url>
    </repository>
  </repositories>

  <dependencies>
    <!-- Dépendances associées au proejt -->
    <!-- Muse loader -->
    <dependency>
      <groupId>org.gu.vesta</groupId>
      <artifactId>muse.component.util</artifactId>
      <version>1.0</version>
    </dependency>

    <!-- Muse facets -->
    <dependency>
      <groupId>org.gu.vesta</groupId>
      <artifactId>muse.facet.impl</artifactId>
      <version>1.0</version>
    </dependency>
    <dependency>
      <groupId>org.gu.vesta</groupId>
      <artifactId>muse.facet.static.api</artifactId>
      <version>1.0</version>
    </dependency>
  </dependencies>

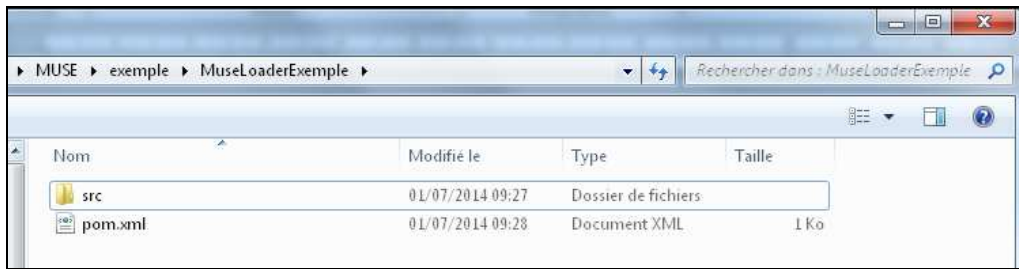
  <!--Le build -->
  <build></build>
</project>
```

Un fichier .pom se compose de différentes parties. En premier vient la description du projet. Ensuite, une seconde partie contient l'adresse du ou des repositories où télécharger les dépendances.

Ensuite viennent la liste des dépendances associées au projet.

Et enfin une partie « build » qui permet d'embarquer toutes les dépendances.

- Créer l'arborescence « src », « main », « java » et y dé-jarer les sources téléchargées du site de Vesta



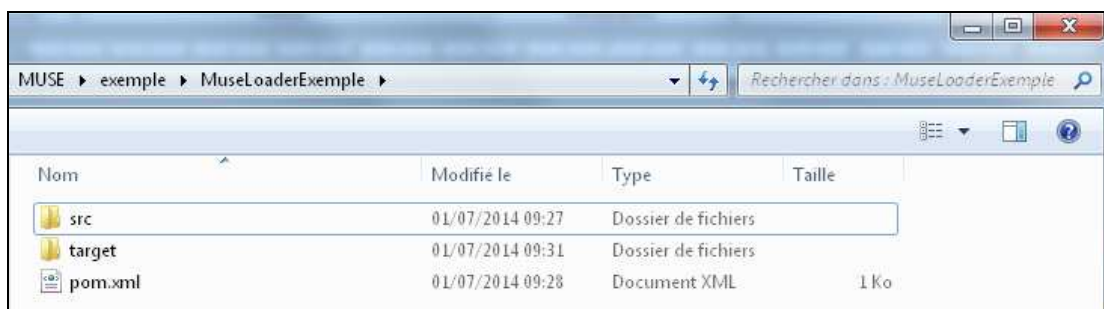
- Exécuter la commande « mvn install » à partir du dossier créé, ce qui va lire le fichier .pom, télécharger les dépendances et compiler le code java. (L'exécutable « mvn » de maven, doit être dans la variable d'environnement « path ».)

```

C:\Windows\system32\cmd.exe
Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple>mvn install
Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple>set MAVEN_OPTS="-XX:MaxPermSize=512M"
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for G2Elab:loadMUSE:jar:1.0-SNAPSHOT
[WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-jar-plugin is missing. @ line 37, column 13
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -----
[INFO] Building LoadMUSE 1.0-SNAPSHOT
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ LoadMUSE ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ LoadMUSE ---
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 1 source file to Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ LoadMUSE ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ LoadMUSE ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ LoadMUSE ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ LoadMUSE ---
[INFO] Building jar: Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\target\LoadMUSE-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ LoadMUSE ---
[INFO] Installing Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\target\LoadMUSE-1.0-SNAPSHOT.jar to C:\_Users\Datas\estrabal\.m2\G2Elab\LoadMUSE\1.0-SNAPSHOT\LoadMUSE-1.0-SNAPSHOT.jar
[INFO] Installing Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\pom.xml to C:\_Users\Datas\estrabal\.m2\G2Elab\LoadMUSE\1.0-SNAPSHOT\LoadMUSE-1.0-SNAPSHOT.pom
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.358s
[INFO] Finished at: Tue Jul 01 15:08:27 CEST 2014
[INFO] Final Memory: 9M/22M
[INFO] -----
Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple>

```

- Un dossier « target » a été créé contenant en particulier le fichier JAR du projet compilé



3.2 Gestion des dépendances

Le jar compilé n'est pas exécutable à l'aide de l'appel à java habituel « java -jar ».

En effet, il faut prendre en considération les dépendances au Jar. Les dépendances ont bien été téléchargées lors de la phase de compilation du projet mais il faut préciser au Jar où les trouver lors de la phase d'exécution. C'est généralement le rôle du Manifest de mettre à disposition l'information nécessaire à l'exécution d'un jar.

Seulement la commande « mvn install » crée le Jar sans ajouter le Manifest.

Trois choix s'offrent alors :

3.2.1 Utiliser Maven pour exécuter le jar compilé

L'utilisation de Maven pour l'exécution du jar compilé va permettre de prendre en compte les dépendances liées au Jar.

Dans un premier temps, il va falloir à minima ajouter la manifest au Jar compilé.

Pour se faire, il faut rajouter au fichier .pom présent à la racine le code suivant :

```
<build>
  <plugins>
    <plugin>
      <artifactId>maven-jar-plugin</artifactId>
      <configuration>
        <archive>
          <manifest>
            <mainClass>org.gu.vesta.muse.sample.MuseLoaderExample</mainClass>
            <addClasspath>true</addClasspath>
            <classpathPrefix>dependency</classpathPrefix>
          </manifest>
        </archive>
      </configuration>
    </plugin>
  </plugins>
</build>
```

La commande à utiliser pour compiler le Jar est la suivante :

```
mvn clean install
```

```
C:\Windows\system32\cmd.exe
Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple>mvn install
Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple>set MAVEN_OPTS="-XX:MaxPermSize=512M"
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for G2Elab:LoadMUSE:jar:1.0-SNAPSHOT
[WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-jar-plugin is missing. @ line 37, column 13
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -----
[INFO] Building LoadMUSE 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ LoadMUSE ---
[WARNING] Using platform encoding (cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ LoadMUSE ---
[WARNING] File encoding has not been set, using platform encoding (cp1252, i.e. build is platform dependent!
[INFO] Compiling 1 source file to Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ LoadMUSE ---
[WARNING] Using platform encoding (cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ LoadMUSE ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ LoadMUSE ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ LoadMUSE ---
[INFO] Building jar: Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\target\LoadMUSE-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ LoadMUSE ---
[INFO] Installing Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\target\LoadMUSE-1.0-SNAPSHOT.jar to C:\Users\Datas\estrabal\.m2\G2Elab\LoadMUSE\1.0-SNAPSHOT\LoadMUSE-1.0-SNAPSHOT.jar
[INFO] Installing Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple\pom.xml to C:\Users\Datas\estrabal\.m2\G2Elab\LoadMUSE\1.0-SNAPSHOT\LoadMUSE-1.0-SNAPSHOT.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO]
[INFO] Total time: 3.517s
[INFO] Finished at: Tue Jul 01 16:08:11 CEST 2014
[INFO] Final Memory: 9M/22M
[INFO] -----
Z:\DOCUMENTS - PROJETS\MUSE\exemple\MuseLoaderExemple>
```

L'exécution du jar se fait en utilisant Maven qui va être alors capable de recréer l'environnement d'exécution.

```
mvn exec:java -Dexec.mainClass="org.gu.vesta.muse.sample.MuseLoaderExample"
```

3.2.2 Créer un jar indépendant

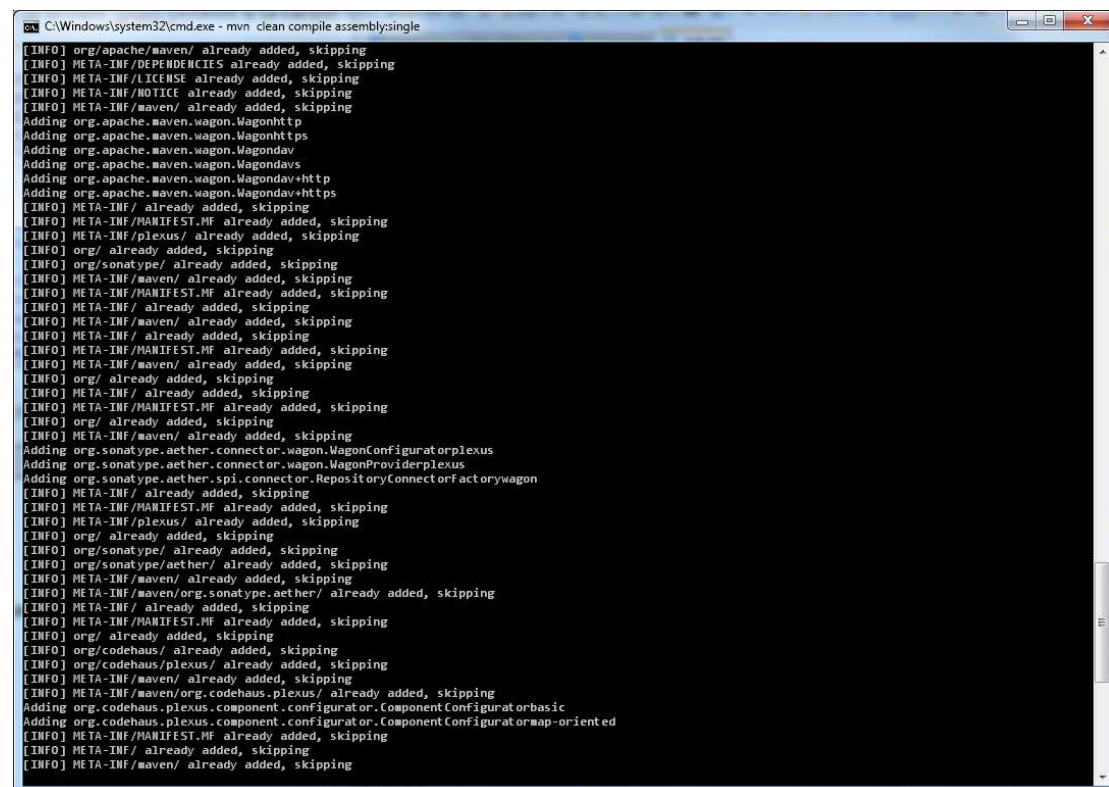
Créer un Jar indépendant, revient à copier à l'intérieur du Jar l'ensemble des dépendances dont il aura besoin pour son exécution.

Cette opération s'effectue en utilisant le plugin « **maven-assembly-plugin** ». Il faut alors modifier le fichier .pom présent à la racine du projet en ajoutant le code ci-dessous :

```
<build>
  <plugins>
    <plugin>
      <artifactId>maven-assembly-plugin</artifactId>
      <configuration>
        <archive>
          <manifest>
            <mainClass>org.gu.vesta.muse.sample.MuseLoaderExample</mainClass>
          </manifest>
        </archive>
        <descriptorRefs>
          <descriptorRef>jar-with-dependencies</descriptorRef>
        </descriptorRefs>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Puis lancer la compilation du Jar en employant la commande suivante :

```
mvn clean compile assembly:single
```



```
C:\Windows\system32\cmd.exe - mvn clean compile assembly:single
[INFO] org/apache/maven/ already added, skipping
[INFO] META-INF/DEPENDENCIES already added, skipping
[INFO] META-INF/LICENSE already added, skipping
[INFO] META-INF/NOTICE already added, skipping
[INFO] META-INF/maven/ already added, skipping
Adding org.apache.maven.wagon.WagonHttp
Adding org.apache.maven.wagon.WagonHttps
Adding org.apache.maven.wagon.WagonDav
Adding org.apache.maven.wagon.WagonDav+http
Adding org.apache.maven.wagon.WagonDav+https
[INFO] META-INF/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] META-INF/plexus/ already added, skipping
[INFO] org/ already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] META-INF/maven/ already added, skipping
Adding org.sonatype.aether.connector.wagon.WagonConfiguratorplexus
Adding org.sonatype.aether.connector.wagon.WagonProviderplexus
Adding org.sonatype.aether.spi.connector.RepositoryConnectorFactorywagon
[INFO] META-INF/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] META-INF/plexus/ already added, skipping
[INFO] org/ already added, skipping
[INFO] org/sonatype/ already added, skipping
[INFO] org/sonatype/aether/ already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] META-INF/maven/org.sonatype.aether/ already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] org/ already added, skipping
[INFO] org/codehaus/ already added, skipping
[INFO] org/codehaus/plexus/ already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] META-INF/maven/org.codehaus.plexus/ already added, skipping
Adding org.codehaus.plexus.component.configurator.ComponentConfiguratorbasic
Adding org.codehaus.plexus.component.configurator.ComponentConfiguratorap-oriented
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] META-INF/maven/ already added, skipping
```

Le jar s'exécute ensuite avec la commande suivante :

```
java -jar .\target\LoadMUSE-1.0-SNAPSHOT-jar-with-dependencies.jar
```

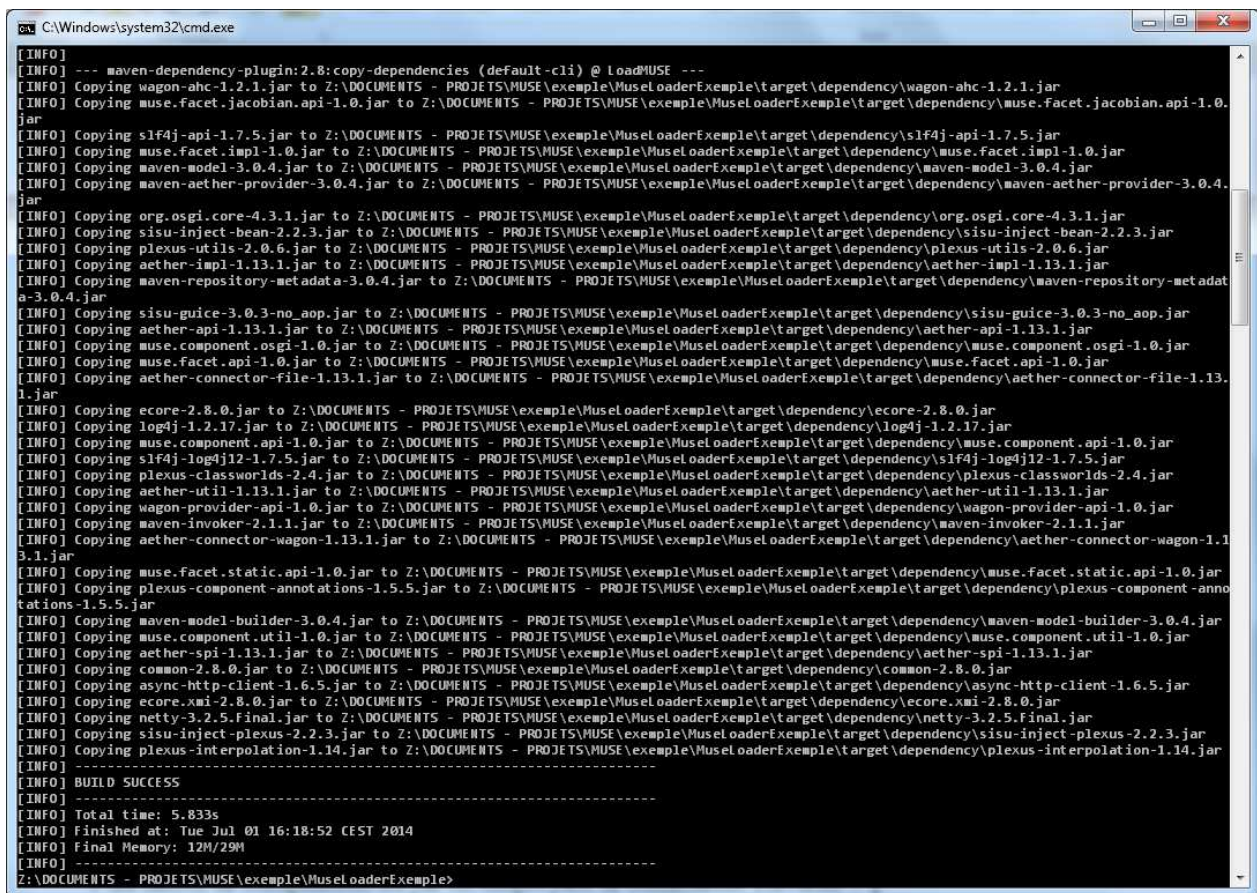

3.2.3 Copier les dépendances

Une dernière méthode consiste à copier toutes les dépendances nécessaires dans un répertoire local.

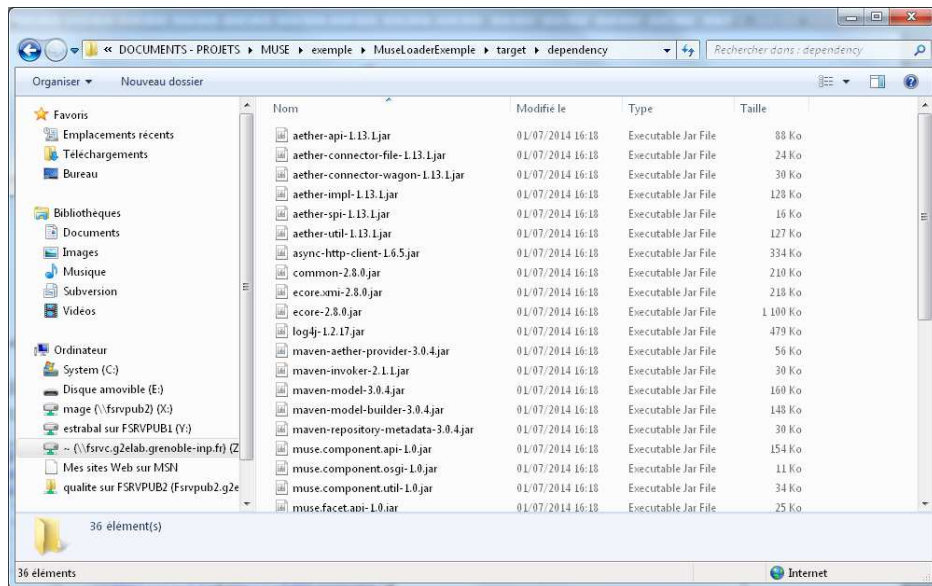
```
<build>
  <plugins>
    <plugin>
      <artifactId>maven-jar-plugin</artifactId>
      <configuration>
        <archive>
          <manifest>
            <mainClass>org.gu.vesta.muse.sample.MuseLoaderExample</mainClass>
            <addClasspath>true</addClasspath>
            <classpathPrefix>dependency/</classpathPrefix>
          </manifest>
        </archive>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Puis lancer la compilation du Jar en employant la commande suivante :

```
mvn clean install dependency:copy-dependencies
```



En se rendant sur le répertoire de travail, on constate qu'un répertoire « dependency » a été créé et qu'il contient toutes les dépendances nécessaires à l'exécution du Jar.



Le jar s'exécute ensuite avec la commande suivante :

```
java -jar .\target\LoadMUSE-1.0-SNAPSHOT.jar
```