

Atelier Maven

Auteurs : NASSRI Abdel-ali & BAYED Fadwa

Installation Eclipse

- Maven est déjà installé dans Eclipse. Si ce n'est pas le cas, installez le plugin m2eclipse.

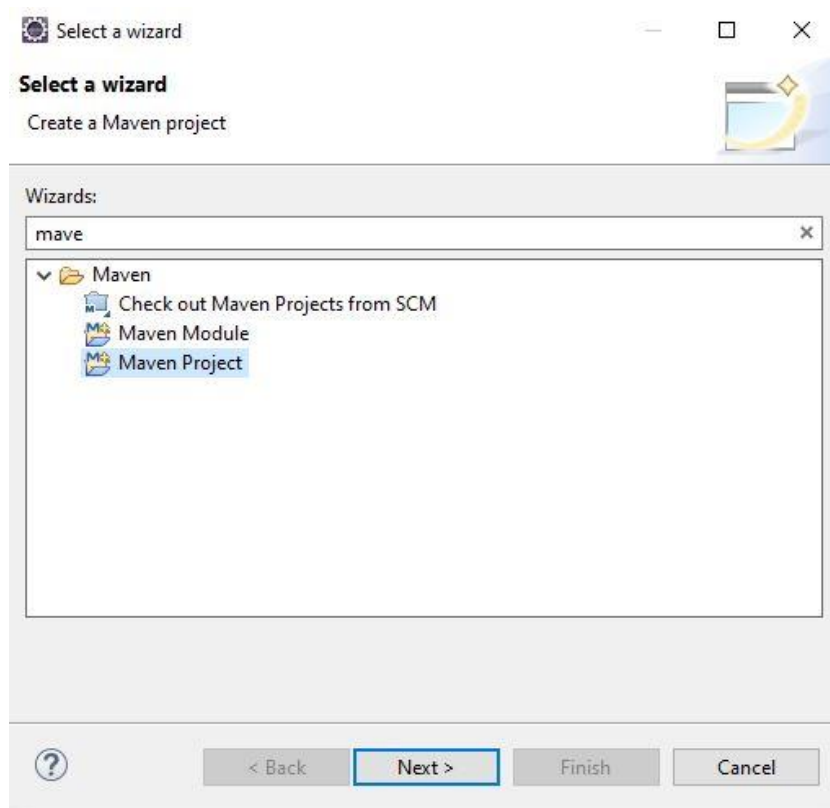
Tutorial :

Si vous ne savez pas comment aborder les ateliers suivants, vous pouvez faire le Tutorial Apache :

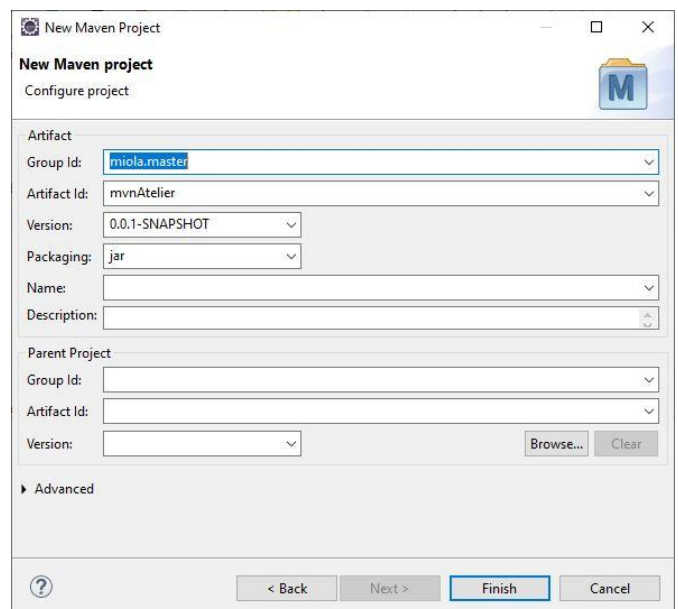
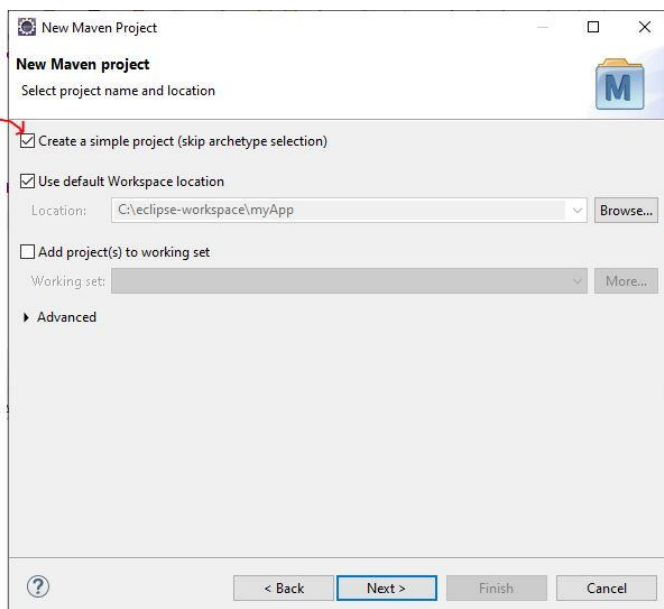
- <http://maven.apache.org/guides/getting-started/index.html>

Atelier 1 - Créer un projet Maven avec Eclipse

Créer un projet de type maven



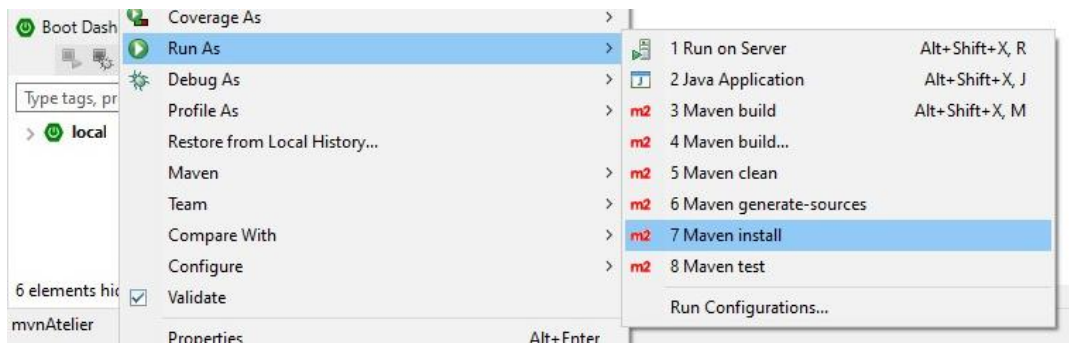
On coche la case pour forcer la création d'un projet vierge puis On l'appelle « mvnAtelier », avec comme groupid « miola.master » .



Dans ce projet, créez une classe MyClass avec un main() affichant « Hello Miola ».

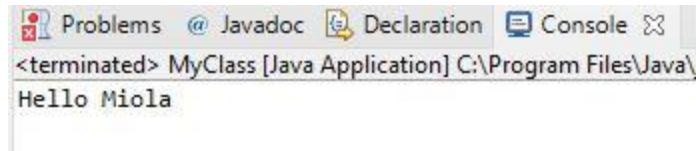
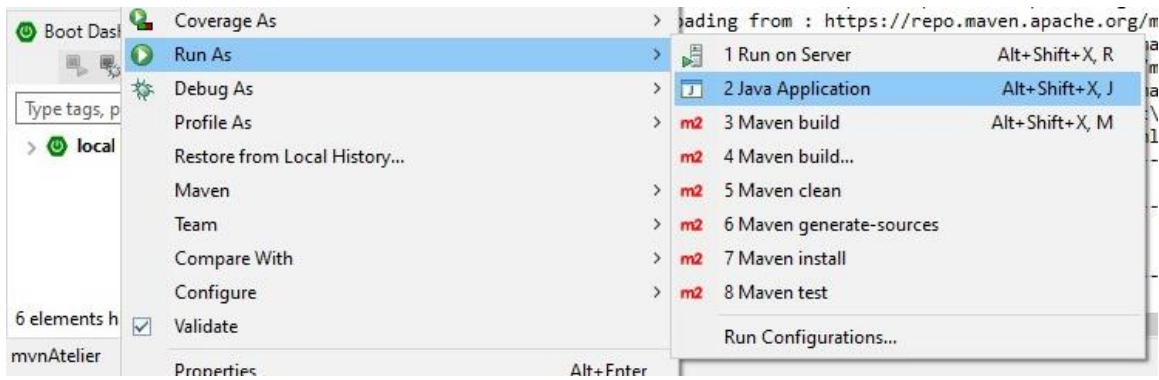
```
*MyClass.java
1 package mvnAtelier;
2
3 public class MyClass {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Hello Miola");
8     }
9 }
10
11 }
12
```

Compilez et installez votre projet avec Maven (Run As->Maven Install).



```
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ mvnAtelier ---
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/
[INFO] Installing C:\eclipse-workspace\mvnAtelier\target\mvnAtelier-0.0.1-SNAPSHOT
[INFO] Installing C:\eclipse-workspace\mvnAtelier\pom.xml to C:\Users\Administratc
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 10.956 s
[INFO] Finished at: 2021-06-03T20:33:41+01:00
[INFO] -----
```

Testez votre projet (Run As->Java Application)



Atelier 2 – Gérer les dépendances de projets

On va tester la dépendance de projets.

Commencez par modifier le projet appelé « mvnAtelier », avec comme groupid « miola.master ».

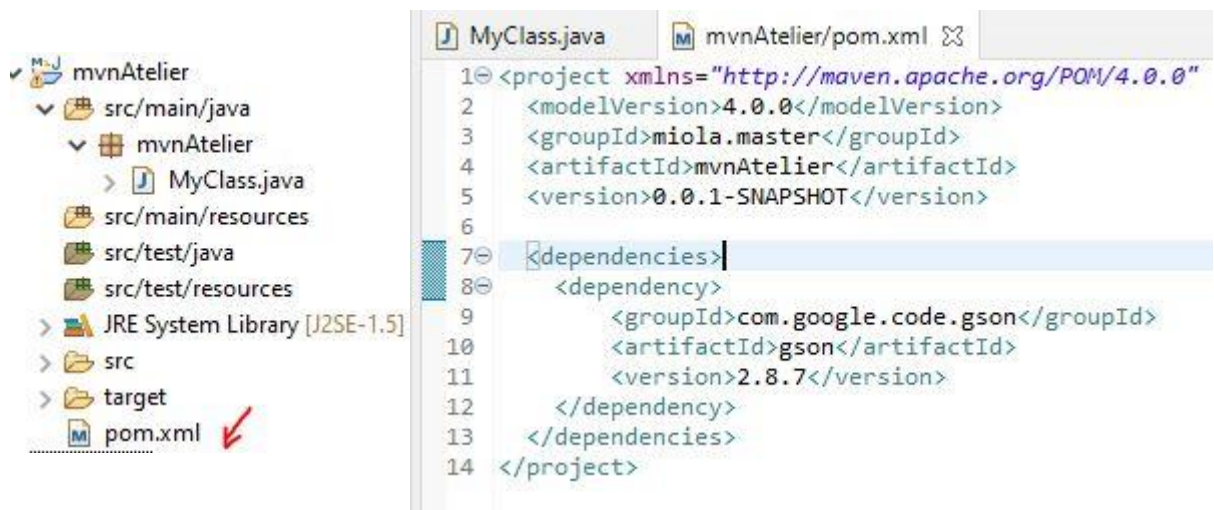
Dans ce projet, on modifie la classe MyClass en utilisant un convertisseur de json qui va convertir la chaine « Hello Miola » de String au Json utilisant la classe Gson de Google.

```
1 package mvnAtelier;
2
3 import com.google.gson.Gson;
4
5 public class MyClass {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Gson gs = new Gson();
10        System.out.println(gs.toJson(new String("Hello Miola")));
11    }
12 }
13
14 }
15
```

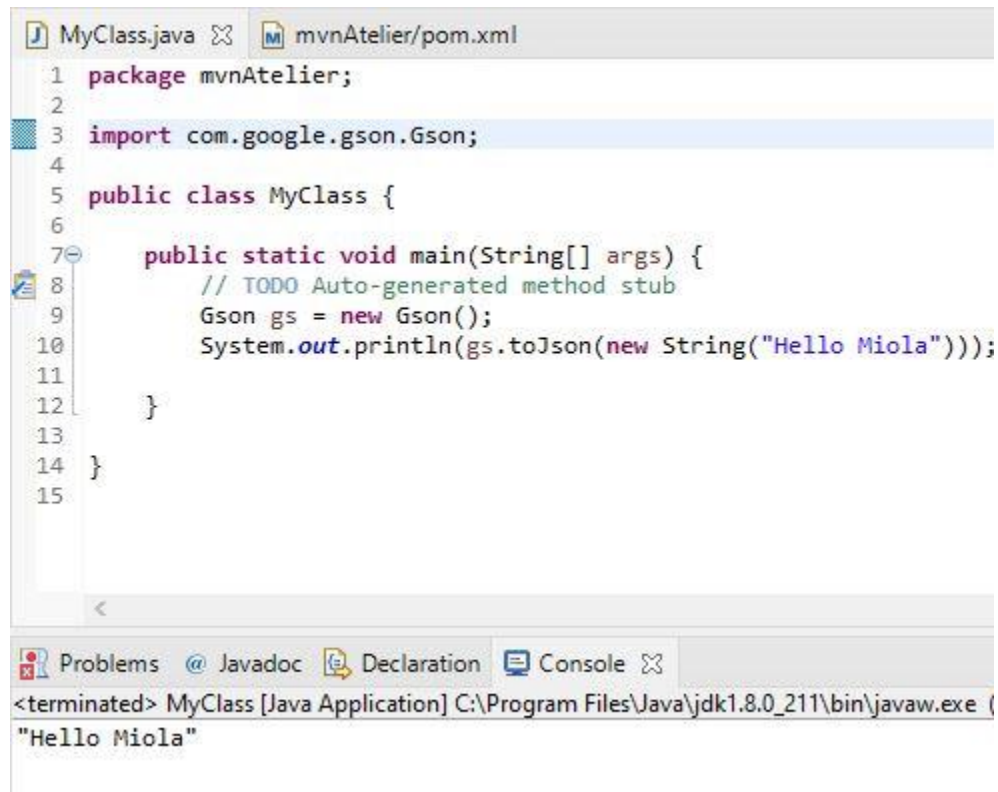
On remarque que le IDE n'a pas pu de connaitre la librairie com.google.gson .

Maintenant, on modifie notre projet pour qu'il arrive à concevoir la classe Gson.

Donc il faut ajouter une dépendance de cette librairie dans notre fichier POM.



Compile et installe notre projet avec Maven. Exécutant notre classe 'MyClass'.



The screenshot shows an IDE window with two tabs: 'MyClass.java' and 'mvnAtelier/pom.xml'. The 'MyClass.java' tab is active, displaying the following code:

```
1 package mvnAtelier;
2
3 import com.google.gson.Gson;
4
5 public class MyClass {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Gson gs = new Gson();
10        System.out.println(gs.toJson(new String("Hello Miola")));
11    }
12
13 }
14
15
```

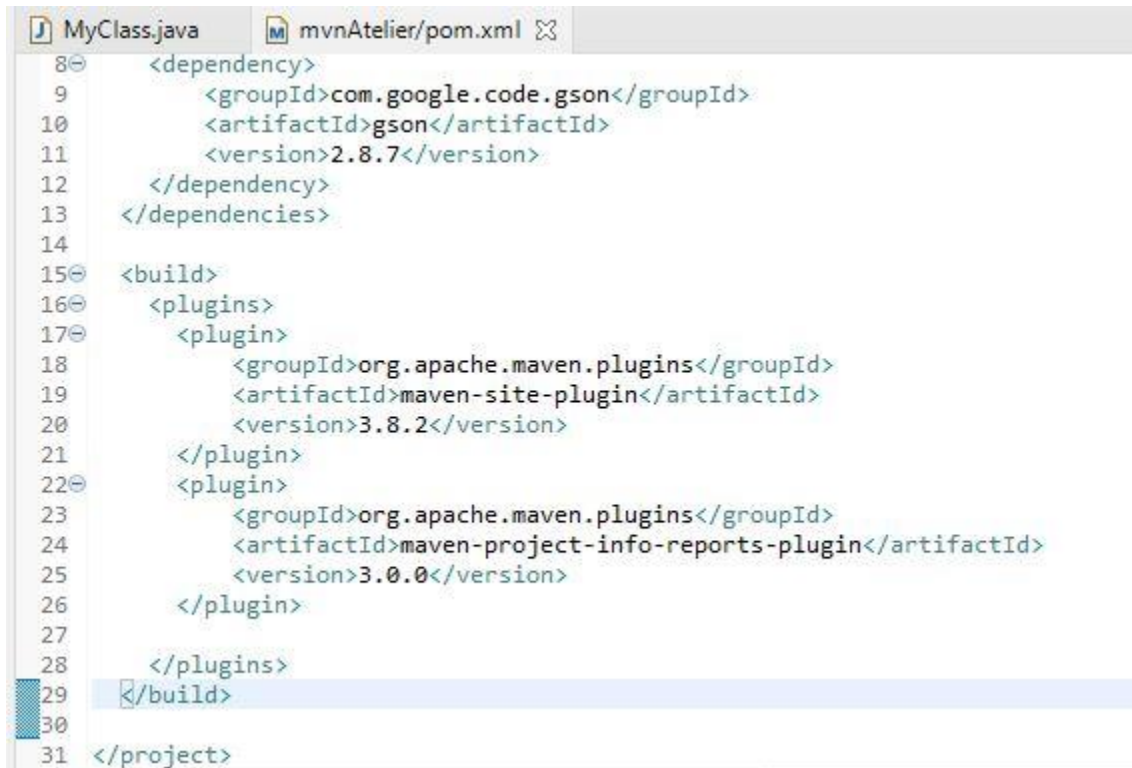
Below the code editor, there is a toolbar with icons for 'Problems', 'Javadoc', 'Declaration', and 'Console'. The 'Console' tab is selected, showing the output of the program:

```
<terminated> MyClass [Java Application] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (
"Hello Miola"
```

- **Atelier 3 – générer des documentations concernant le projet**

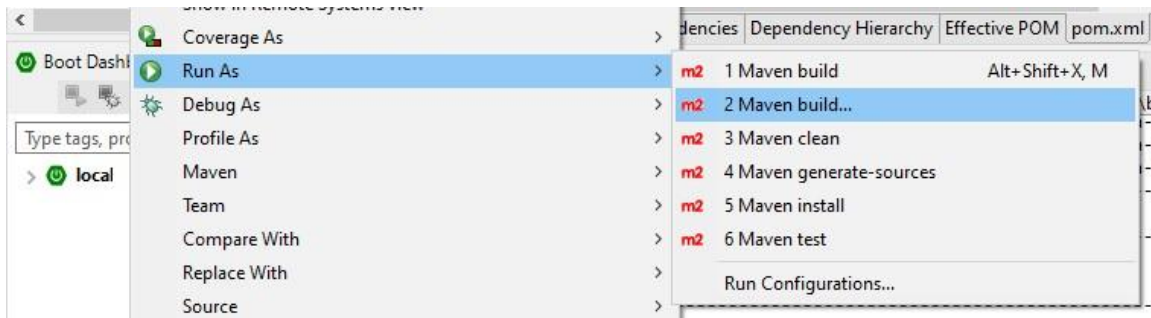
Maven propose une fonctionnalité qui permet de générer automatiquement une doc pour le projet regroupant un certain nombre d'informations utiles le concernant.

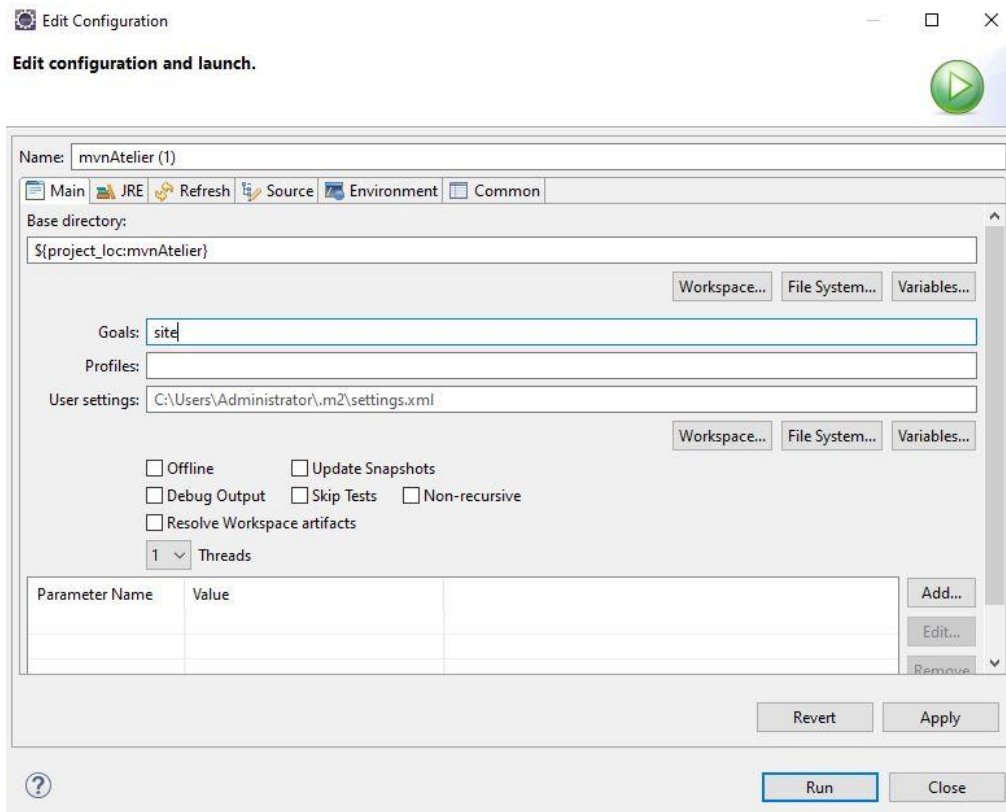
Pour demander la génération du doc, il suffit d'ajouter des plugins au fichier pom.xml



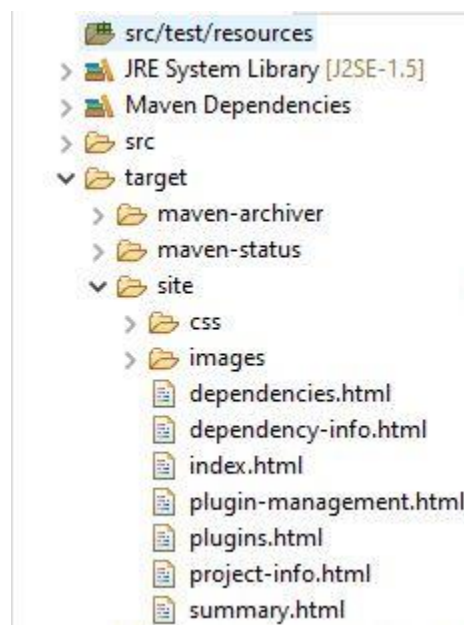
```
8<
9    <groupId>com.google.code.gson</groupId>
10    <artifactId>gson</artifactId>
11    <version>2.8.7</version>
12  </dependency>
13</dependencies>
14
15<build>
16  <plugins>
17    <plugin>
18      <groupId>org.apache.maven.plugins</groupId>
19      <artifactId>maven-site-plugin</artifactId>
20      <version>3.8.2</version>
21    </plugin>
22    <plugin>
23      <groupId>org.apache.maven.plugins</groupId>
24      <artifactId>maven-project-info-reports-plugin</artifactId>
25      <version>3.0.0</version>
26    </plugin>
27  </plugins>
29</build>
30
31</project>
```

Ensuite il suffit de saisir la commande Site dans la case Goals





On remarque qu'un dossier site a été ajouter sous le dossier target d'où on trouve notre documentation générée.



Par défaut, le doc généré contient un certain nombre de pages accessibles par le menu de gauche.

← → ↻ ⓘ File | C:/eclipse-workspace/mvnAtelier/target/site/dependencies.html


Apps Apps DoctorCafe - Softw... Free Axure widgets... Mutaz.net | موقع تح... Prototyping

mvnAtelier

Last Published: 2021-06-03 | Version: 0.0.1-SNAPSHOT

Project Documentation

- Project Information
 - Dependencies**
 - Dependency
 - Information
 - About
 - Plugin Management
 - Plugins
 - Summary



Project Dependencies

compile

The following is a list of compile dependencies for this project. These dependencies are used at compile time.

GroupId	ArtifactId
com.google.code.gson	gson ↗

Project Transitive Dependencies

← → ↻ ⓘ File | C:/eclipse-workspace/mvnAtelier/target/site/index.html


Apps Apps DoctorCafe - Softw... Free Axure widgets... Mutaz.net | موقع تح... Prototyping

mvnAtelier

Last Published: 2021-06-03 | Version: 0.0.1-SNAPSHOT

Project Documentation

- Project Information
 - Dependencies
 - Dependency
 - Information
 - About**
 - Plugin Management
 - Plugins
 - Summary



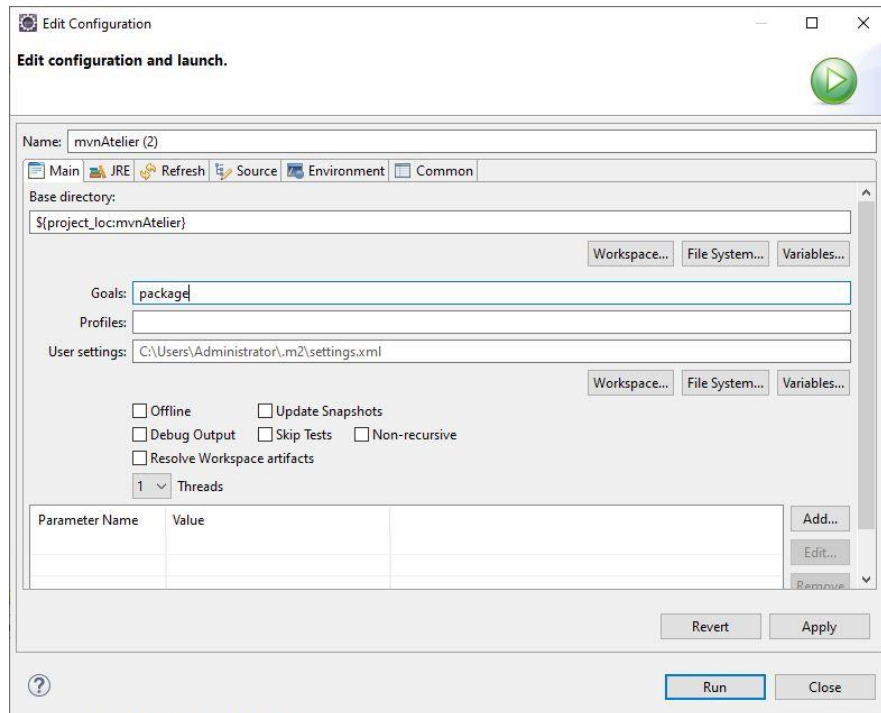
About mvnAtelier

There is currently no description associated with this project.

- **Atelier 4 – Générer Le packaging de projets**

On reste sur notre projet « mvnAtelier » Maven nous propose une fonctionnalité de extraire et créer le fichier Jar/War de notre projet pour le faire déployer.

Il suffit de saisir la commande package dans la case Goals.



Et on remarque que notre fichier Jar à été crée dans le dossier target

