



TP JEE

1. Environnement de développement

PLAN

Introduction

Configuration d'Eclipse pour utiliser le JDK au lieu de JRE

Installation et configuration de Tomcat Server pour Eclipse

Installez Maven dans Eclipse

Premier projet Maven dans Eclipse

INTRODUCTION

Pour développer des application Java, plusieurs possibilités s'offrent à vous : Utiliser un éditeur de texte, puis exécuter votre première application sur une JVM manuellement ou utiliser un environnement de développement qu'on appelle souvent IDE (Integrated Development Environment).

Il existe sur le marché une multitude d'IDE :

- **Eclipse** un environnement de développement Open-Source qui est le plus utilisé.
- **NETBeans** développé par Sun et racheté par Oracle (la même entreprise qui développe la plateforme Java) .
- **IntelliJ** développé par JetBrains c'est un IDE propriétaire et payant.
- etc ...

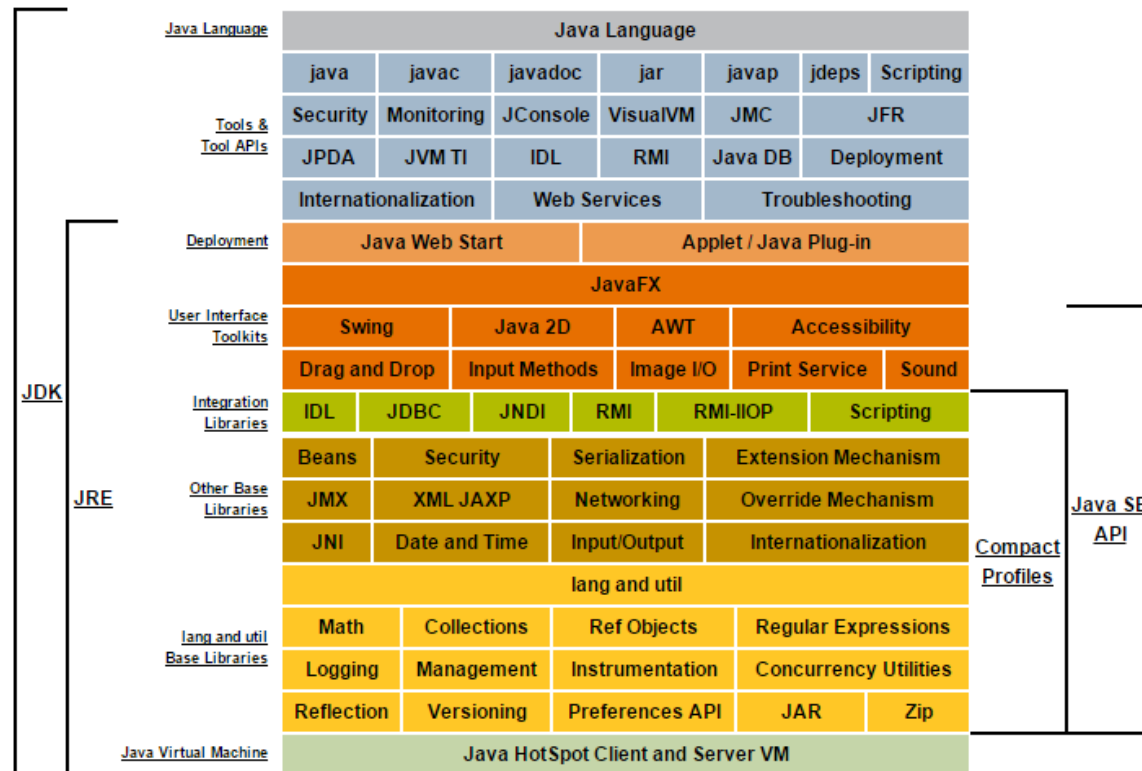
Dans ce TP, on va utiliser Eclipse java Entreprise Edition



CONFIGURATION D'ECLIPSE POUR UTILISER LE JDK AU LIEU DE JRE

Pour comprendre la différence entre JVM, JRE et JDK, considérons la figure suivante :

Description of Java Conceptual Diagram



CONFIGURATION D'ECLIPSE POUR UTILISER LE JDK AU LIEU DE JRE

JVM(Java Virtual Machine) est une machine abstraite. C'est ce qu'on appelle une machine virtuelle car elle n'existe pas physiquement. C'est une spécification qui fournit un environnement d'exécution dans lequel le bytecode Java peut être exécuté (compilant les fichiers .java en fichiers .class. Les fichiers .class contiennent les bytecodes compris par la machine virtuelle Java.).

Les machines virtuelles sont disponibles pour de nombreuses plates-formes matérielles et logicielles. JVM, JRE et JDK dépendent de la plate-forme car la configuration de chaque système d'exploitation est différente. Cependant, Java est indépendant de la plate-forme.

CONFIGURATION D'ECLIPSE POUR UTILISER LE JDK AU LIEU DE JRE

JRE (Java Runtime Environment) fournit la configuration minimale requise pour l'exécution d'une application Java. C'est l'implémentation de JVM. Cela existe physiquement. Il contient un ensemble de bibliothèques et d'autres fichiers que JVM utilise au moment de l'exécution.

JDK (Java Development Kit) est un environnement de développement logiciel utilisé pour développer des applications et des applets Java. Cela existe physiquement. Le JDK est un kit (ou package) qui comprend JRE + development tools.

Le JDK est destiné aux programmeurs et aux développeurs de logiciels, tandis que le JRE est utilisé par ceux qui souhaitent uniquement exécuter les programmes Java.

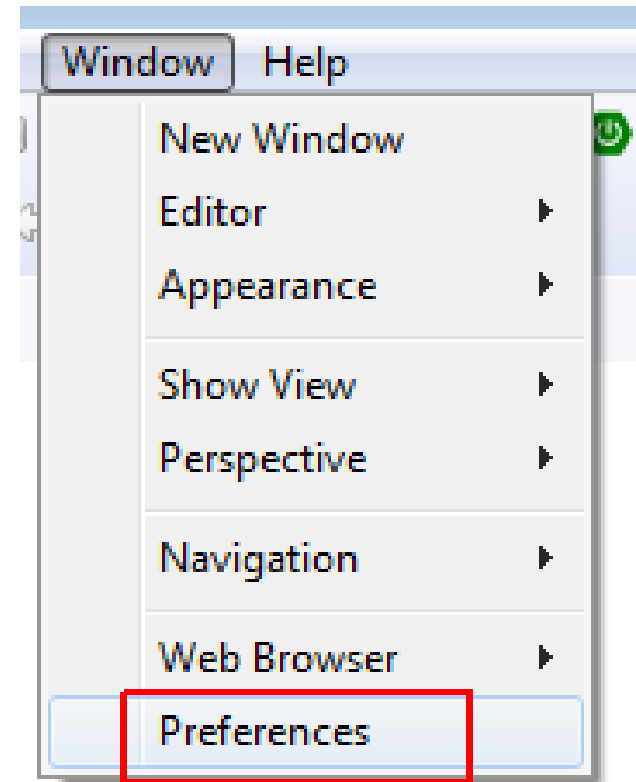
CONFIGURATION D'ECLIPSE POUR UTILISER LE JDK AU LIEU DE JRE

Si JRE et JDK sont installés sur votre machine, Eclipse utilisera JRE comme un outil par défaut afin de build vos projets. Pourtant, pour certains projets ou d'autres fonctionnements, vous devez utiliser JDK à la place de JRE.

Pour configurer jdk dans Eclipse, vérifiez que vous avez téléchargé et installé jdk sur votre machine (dans ce TP, on utilisera jdk1.8).

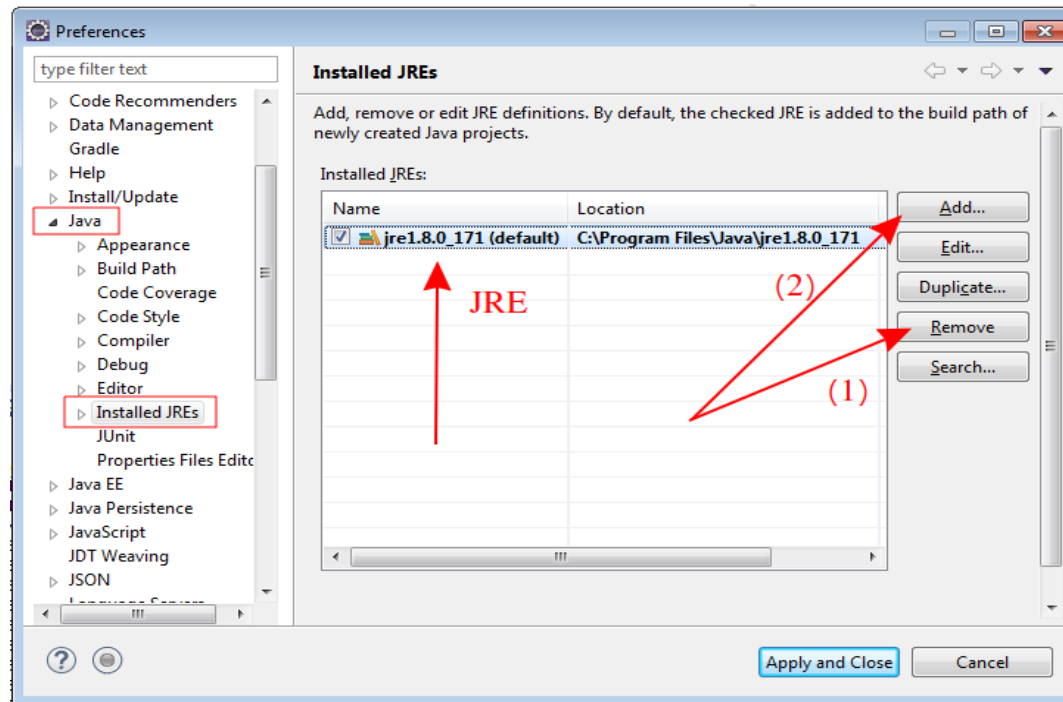
CONFIGURATION D'ECLIPSE POUR UTILISER LE JDK AU LIEU DE JRE

Dans Eclipse, sélectionnez Window/Preferences

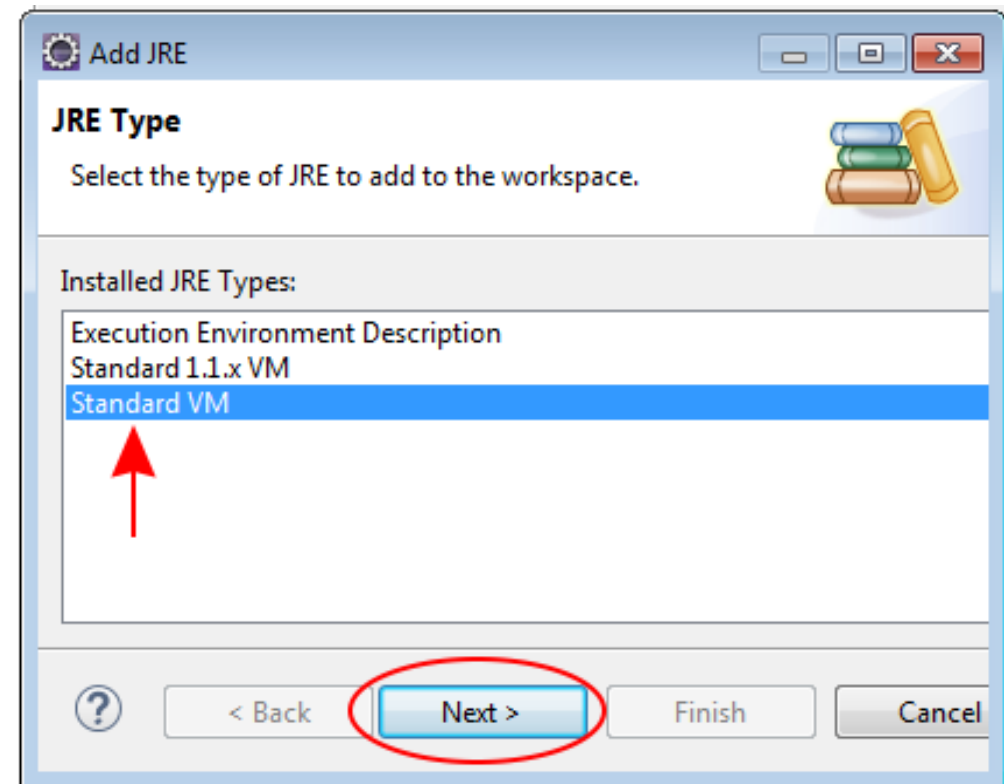
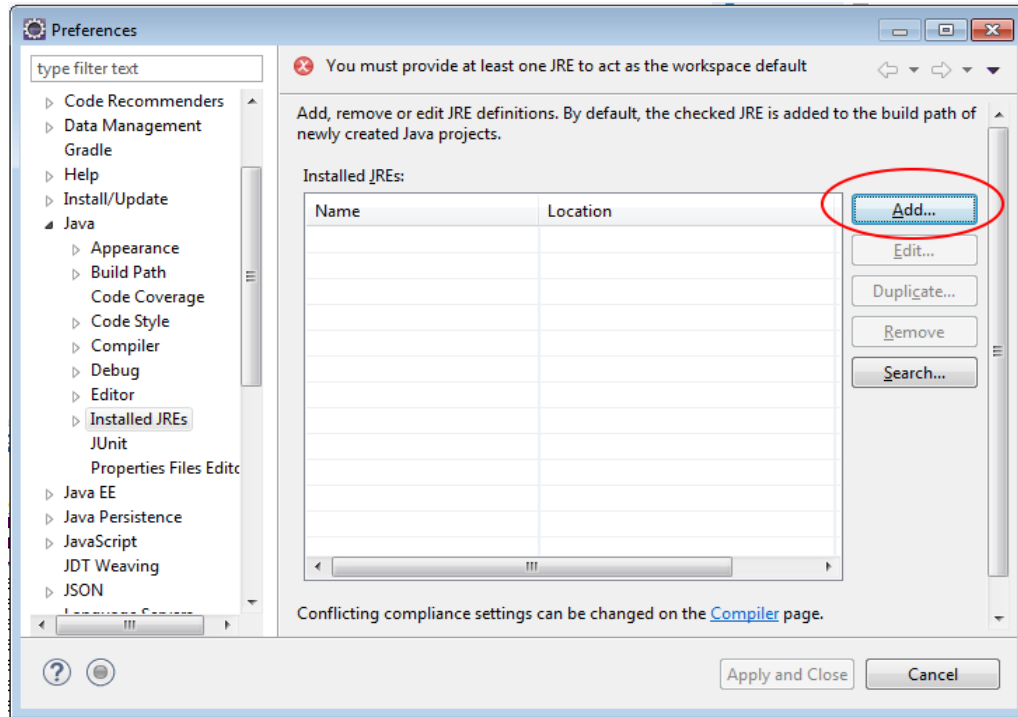


CONFIGURATION D'ECLIPSE POUR UTILISER LE JDK AU LIEU DE JRE

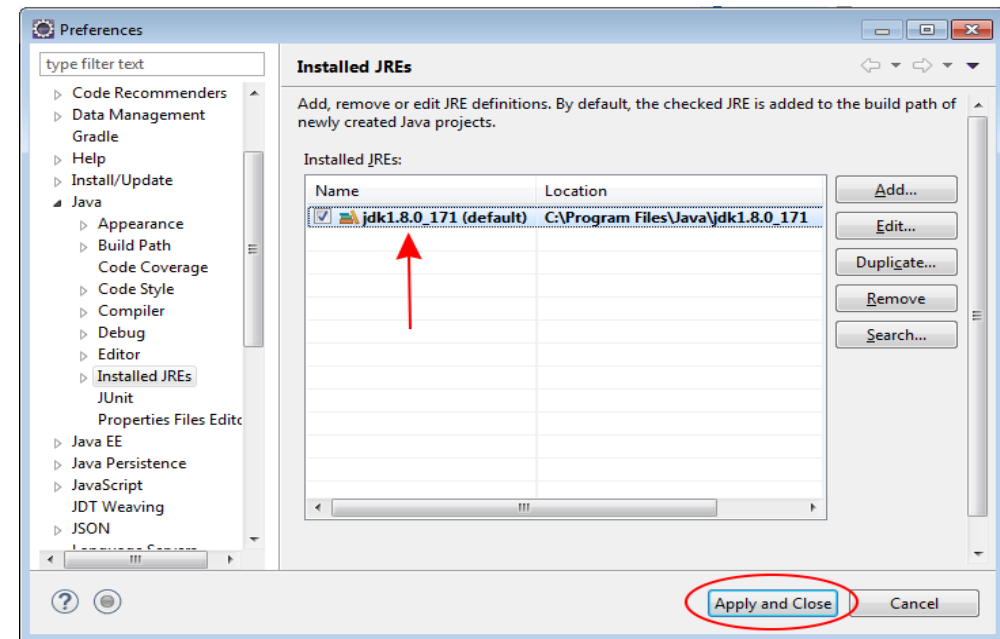
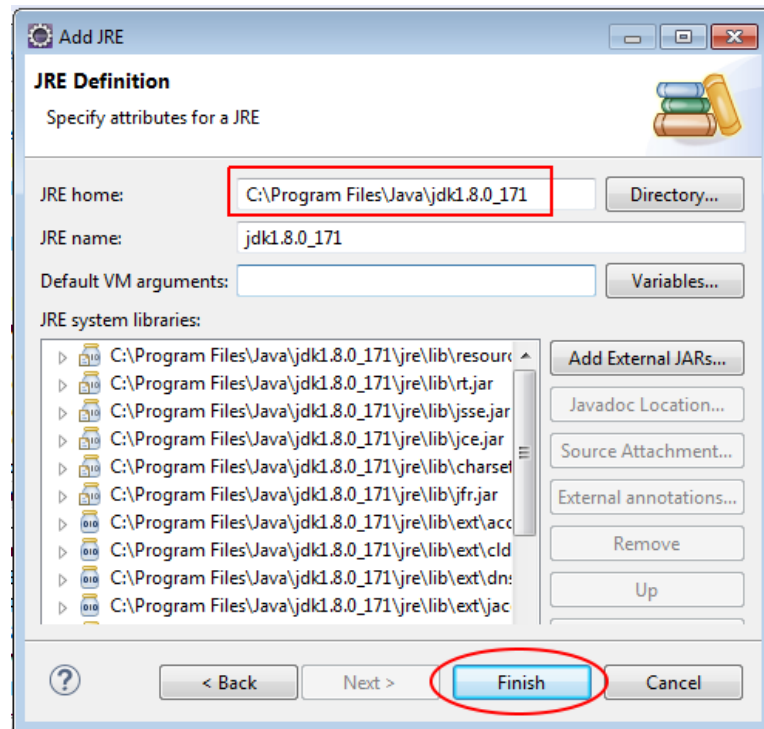
Si vous trouvez que Eclipse utilise JRE, vous devez supprimer JRE et ajouter JDK comme suit:



CONFIGURATION D'ECLIPSE POUR UTILISER LE JDK AU LIEU DE JRE



CONFIGURATION D'ECLIPSE POUR UTILISER LE JDK AU LIEU DE JRE



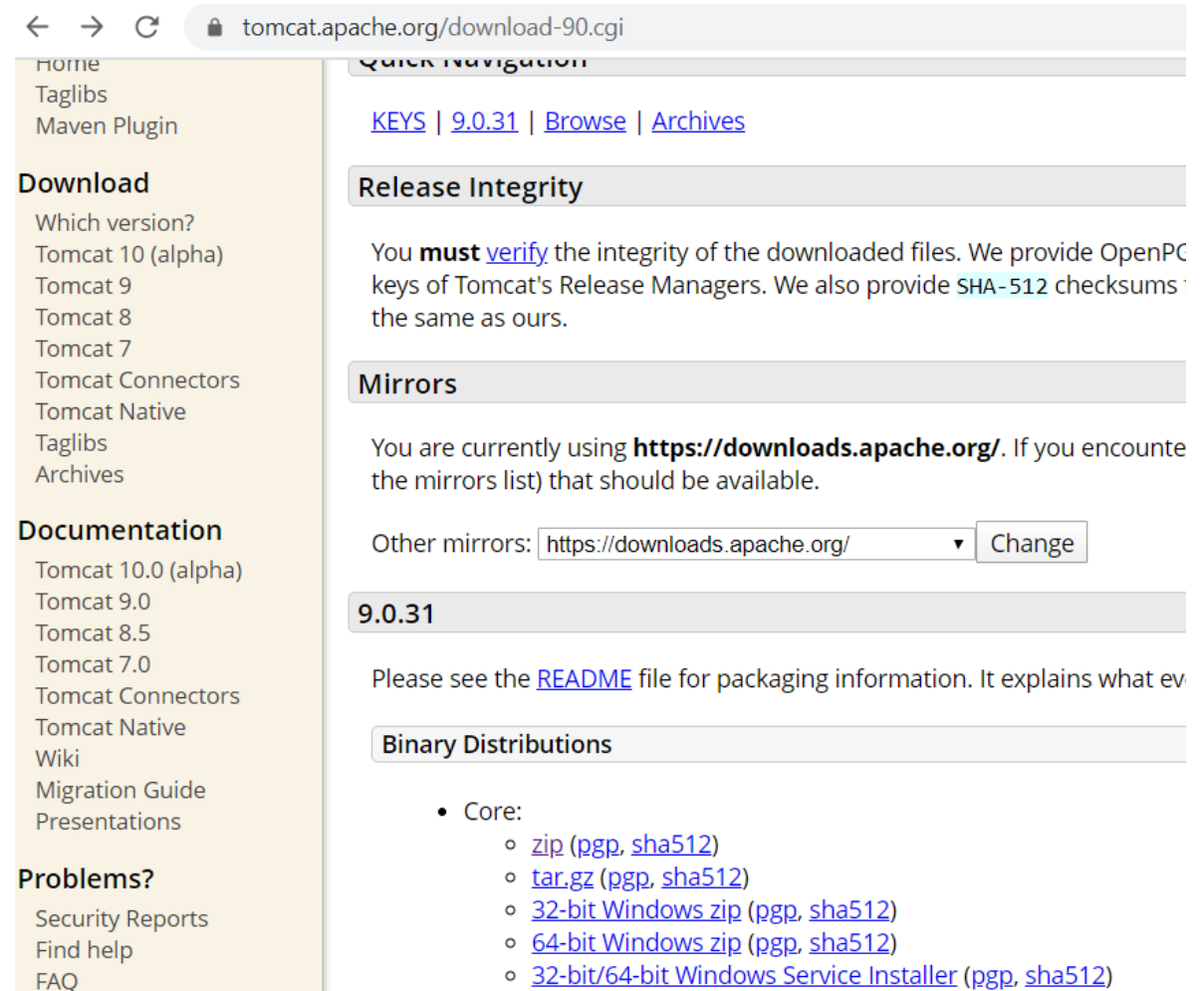
INSTALLATION ET CONFIGURATION DE TOMCAT SERVER POUR ECLIPSE

Allez vers URL <https://tomcat.apache.org/download-90.cgi>

Téléchargez le fichier zip

Décompressez Tomcat dans un dossier spécifique sur le disque dur

Sur **Eclipse**, sélectionnez **Window/References**

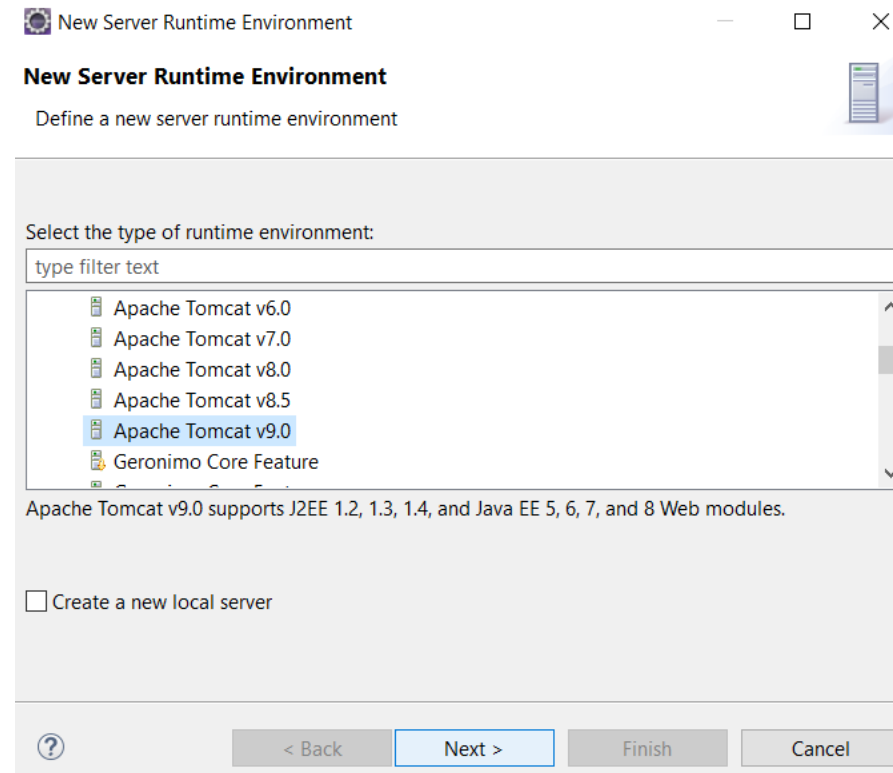
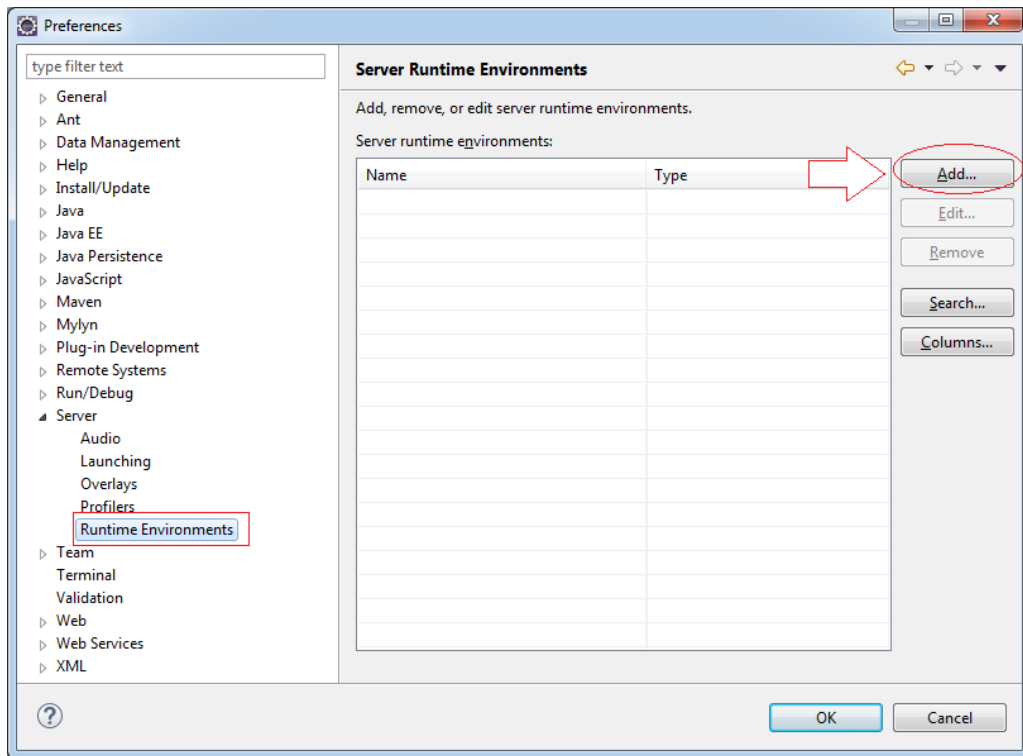


The screenshot shows the Apache Tomcat download page for version 9.0.31. The browser address bar shows 'tomcat.apache.org/download-90.cgi'. The page has a left sidebar with navigation links: Home, Taglibs, Maven Plugin, Download, Documentation, and Problems?. The 'Download' section is active, showing options for 'Which version?' (Tomcat 10 (alpha), Tomcat 9, Tomcat 8, Tomcat 7), 'Tomcat Connectors', 'Tomcat Native', 'Taglibs', and 'Archives'. The 'Documentation' section lists links for Tomcat 10.0 (alpha), Tomcat 9.0, Tomcat 8.5, Tomcat 7.0, Tomcat Connectors, Tomcat Native, Wiki, Migration Guide, and Presentations. The 'Problems?' section includes Security Reports, Find help, and FAQ. The main content area has a 'Quick Navigation' bar with links for KEYS, 9.0.31, Browse, and Archives. Below this is a 'Release Integrity' section with a warning to verify the integrity of downloaded files. A 'Mirrors' section shows the current mirror as 'https://downloads.apache.org/' with a 'Change' button. The '9.0.31' section contains a link to the README file. The 'Binary Distributions' section lists the following options:

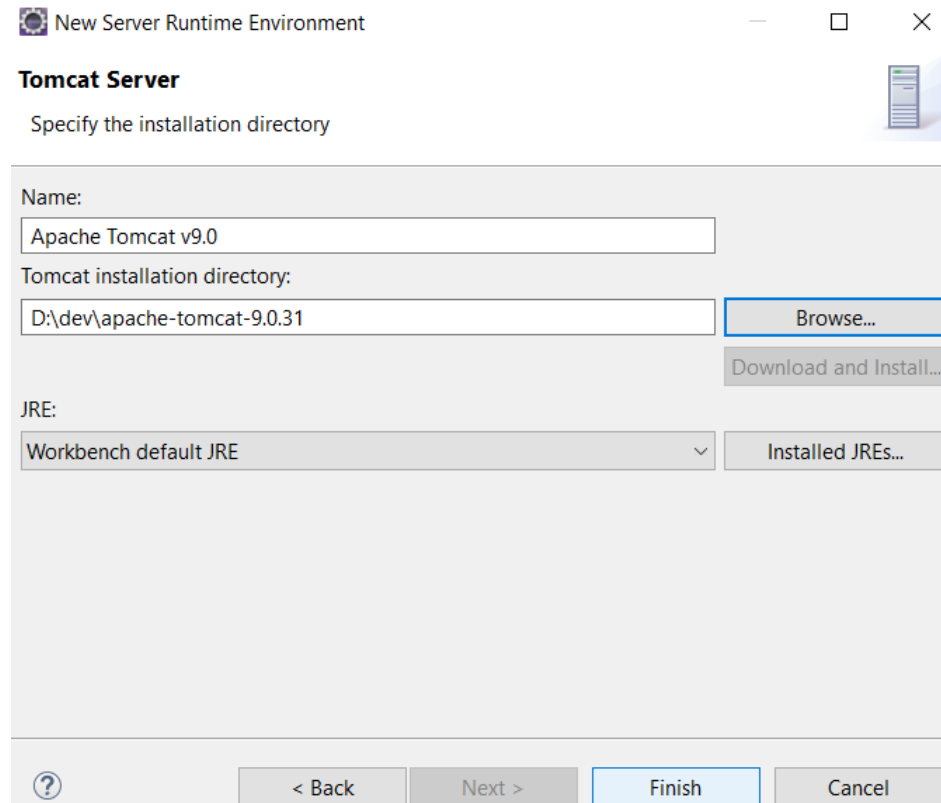
- Core:
 - zip (pgp, sha512)
 - tar.gz (pgp, sha512)
 - 32-bit Windows zip (pgp, sha512)
 - 64-bit Windows zip (pgp, sha512)
 - 32-bit/64-bit Windows Service Installer (pgp, sha512)

INSTALLATION ET CONFIGURATION DE TOMCAT SERVER POUR ECLIPSE

Créez un nouveau Runtime Server



INSTALLATION ET CONFIGURATION DE TOMCAT SERVER POUR ECLIPSE



New Server Runtime Environment

Tomcat Server

Specify the installation directory

Name:
Apache Tomcat v9.0

Tomcat installation directory:
D:\dev\apache-tomcat-9.0.31 Browse...

Download and Install...

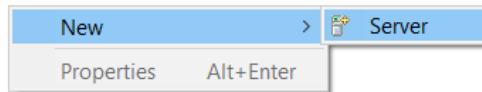
JRE:
Workbench default JRE Installed JREs...

? < Back Next > Finish Cancel

INSTALLATION ET CONFIGURATION DE TOMCAT SERVER POUR ECLIPSE

Markers Properties Servers Data Source Explorer Snippets Console

No servers are available. Click this link to create a new server.



projet-maven/pom.xml projet-maven/pom.xml app.java Tomcat v9.0 Server at localhost

her common settings.

mcats v9.0 Server at localhost

localhost

Apache Tomcat v9.0

servers/Tomcat v9.0 Server at localhost Browse...

atalina.base) and deploy path. Server must be resent to make changes.

1 (does not modify Tomcat installation)

akes control of Tomcat installation)

es not modify Tomcat installation)

plugins\org.eclipse.wst.server.co Browse...

ult value (currently set)

js Browse...

Publishing

Timeouts

Specify the time limit to complete server operations.

Start (in seconds): 45

Stop (in seconds): 15

Ports

Modify the server ports.

Port Name	Port Number
Tomcat admin port	8005
HTTP/1.1	8080

MIME Mappings

Overview Modules

Markers Properties Servers Data Source Explorer Snippets Console

Tomcat v9.0 Server at localhost [Stopped, Republish]

INSTALLEZ MAVEN DANS ECLIPSE

Maven est un outil de construction de projets (build) open source développé par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine.

Le site web officiel est **<http://maven.apache.org>**

Il permet notamment :

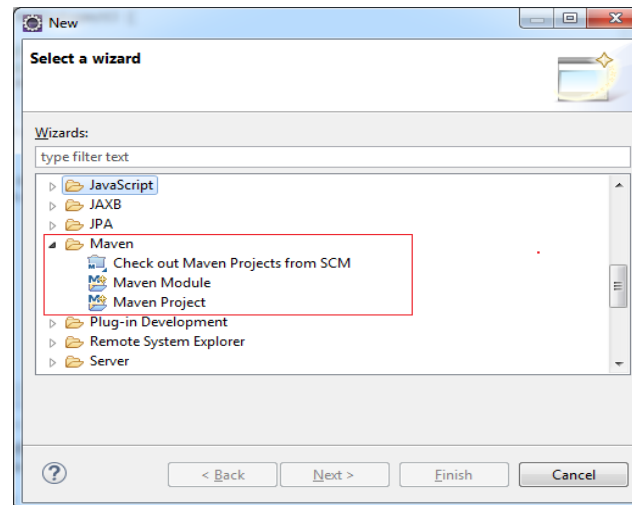
- d'automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet
- de gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet
- de générer des documentations concernant le projet

Pour gérer les dépendances du projet vis-à-vis de bibliothèques, Maven utilise un ou plusieurs dépôts qui peuvent être locaux ou distants.

INSTALLEZ MAVEN DANS ECLIPSE

Tout d'abord, vérifiez si votre Eclipse a installé Maven ou non.

Cliquez sur File/New/Other..



S'il y a Maven Wizard cela signifie que votre Eclipse a été installé Maven Plugin , vous pouvez être prêt à travailler. Cependant, vous pouvez toujours réinstaller dans le but d'avoir la dernière version.

INSTALLEZ MAVEN DANS ECLIPSE

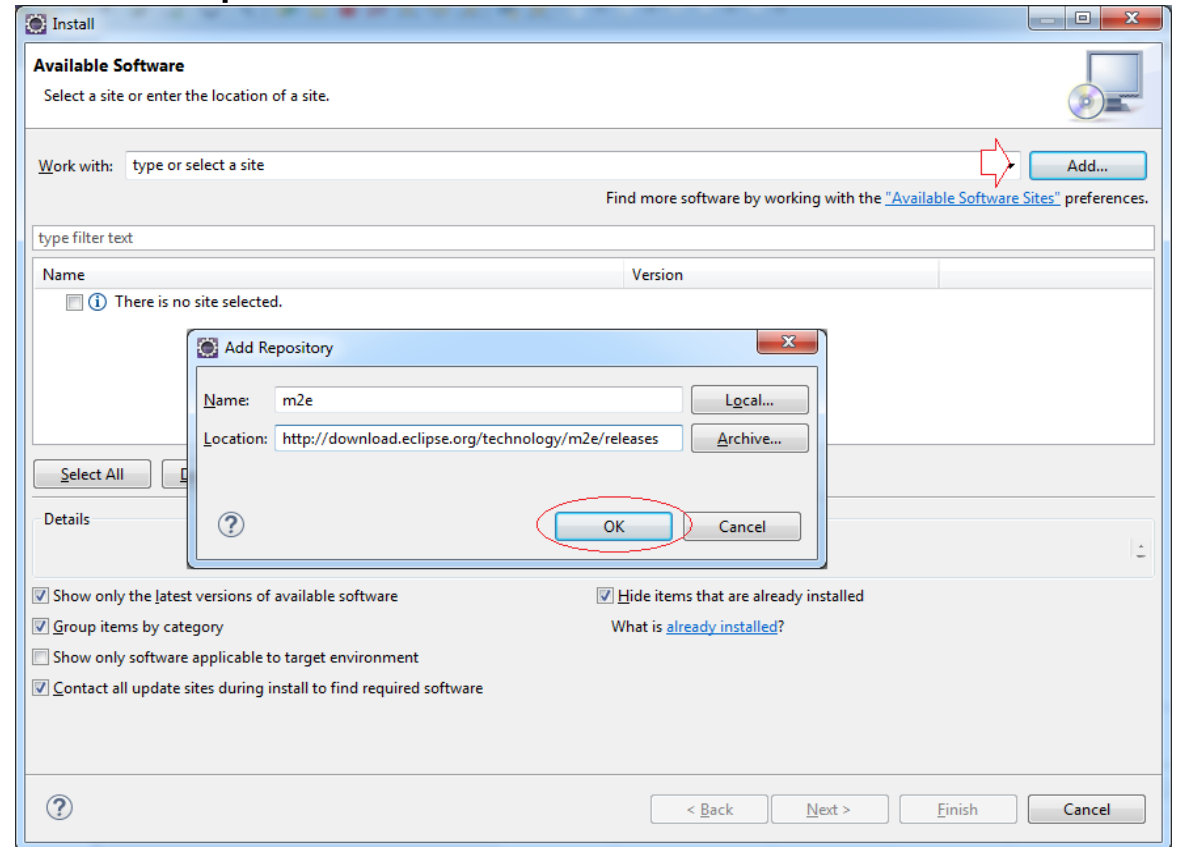
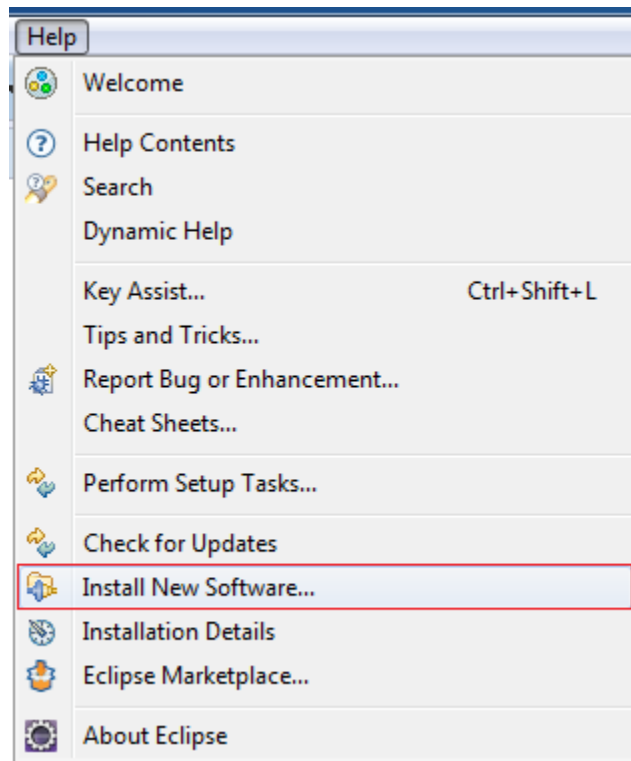
Vous avez deux façons pour installer Maven dans Eclipse :

- 1 - Installez Maven comme un plugin intégré sur Eclipse.
- 2 - Installez Maven indépendamment sur Windows et déclarez-le pour Eclipse peut l'utiliser.

INSTALLEZ MAVEN DANS ECLIPSE

Afin d'installer Maven comme un plugin intégré sur Eclipse.

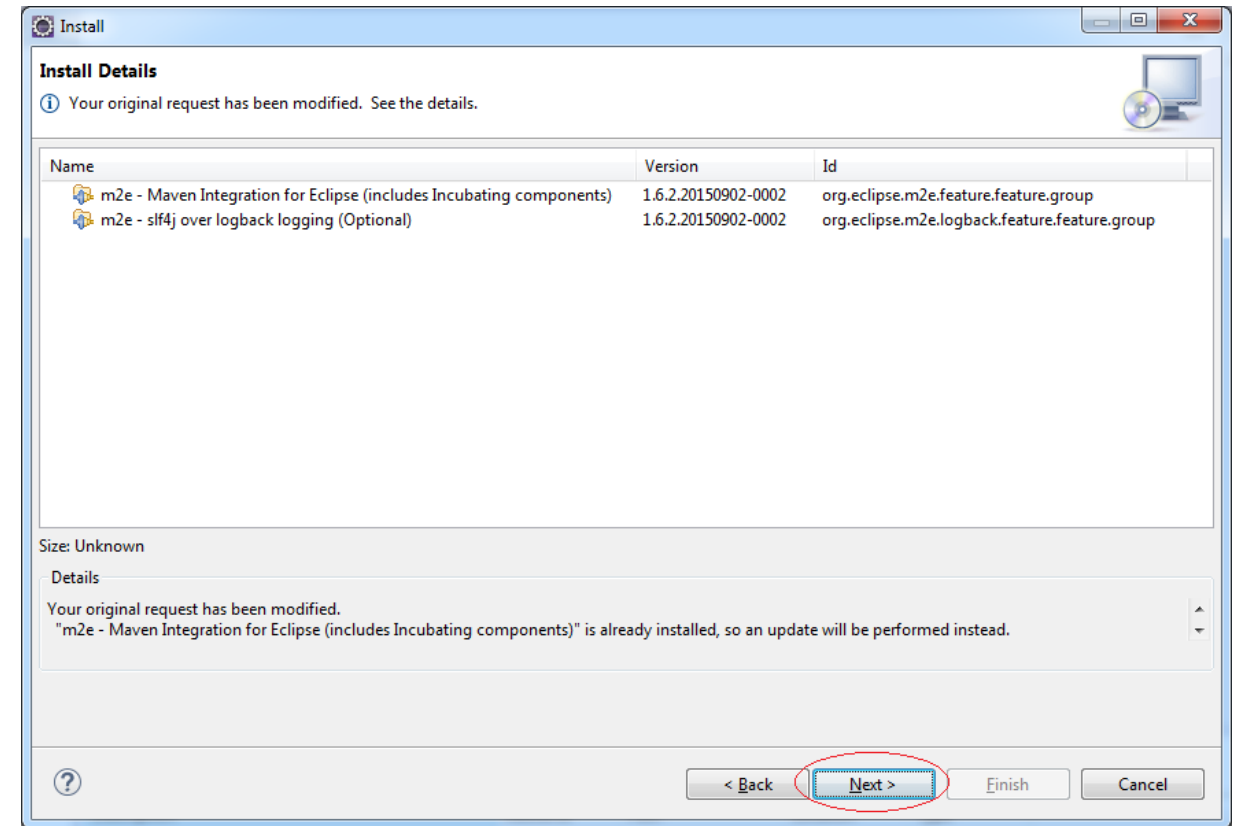
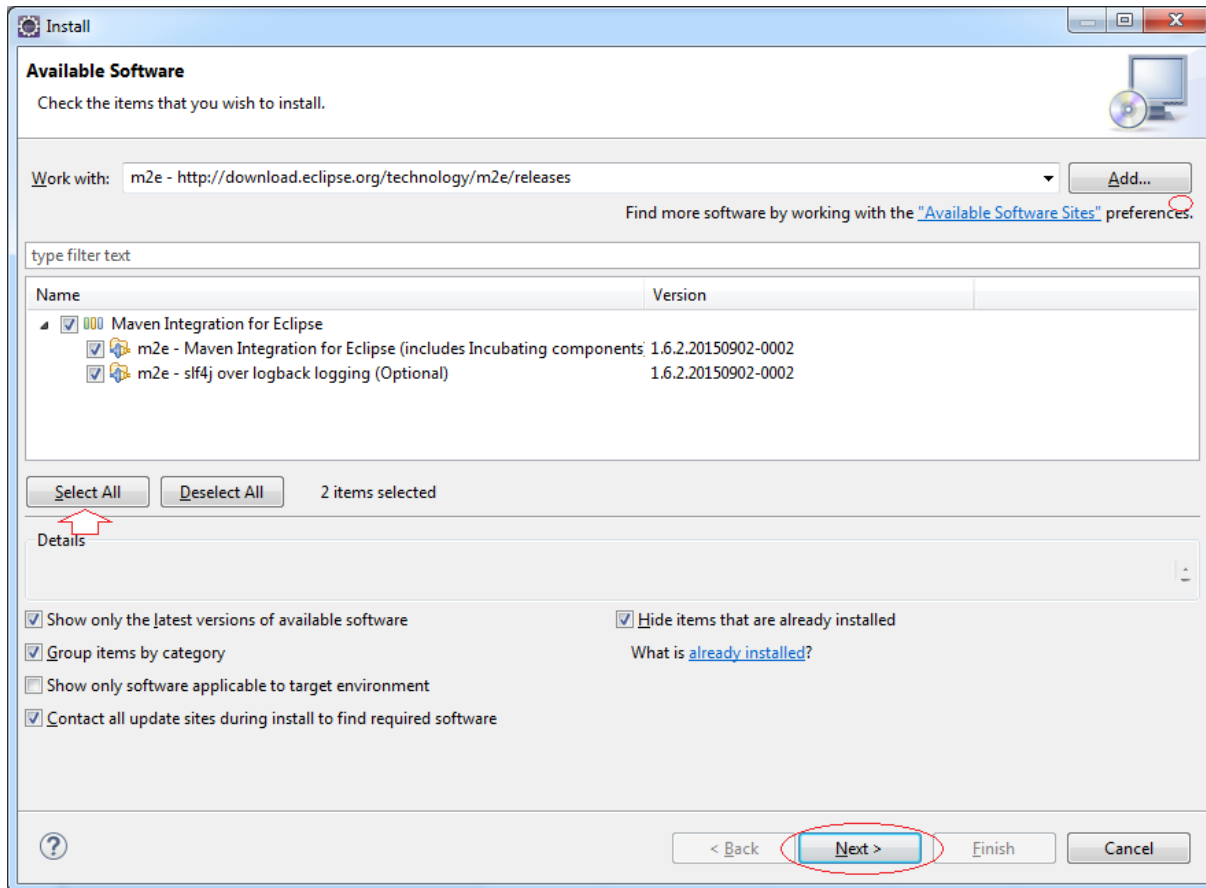
Cliquez sur Help/install New Software



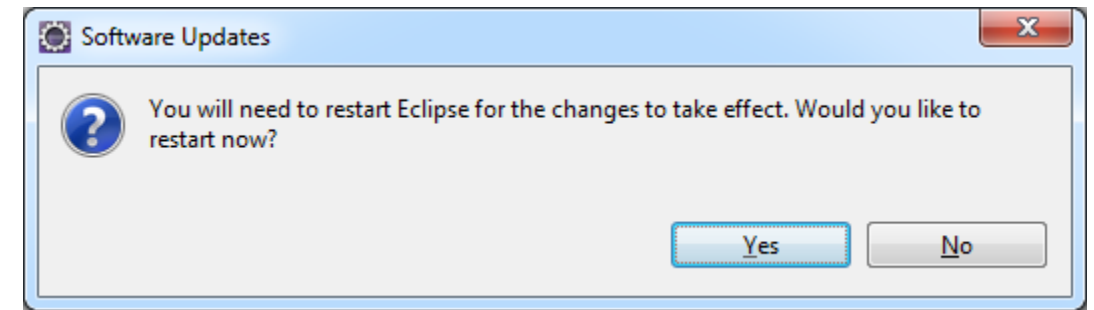
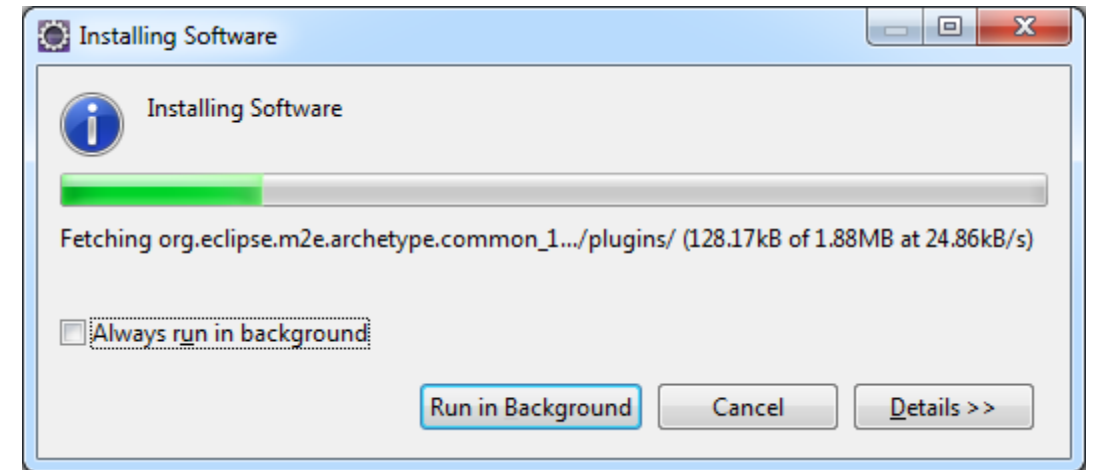
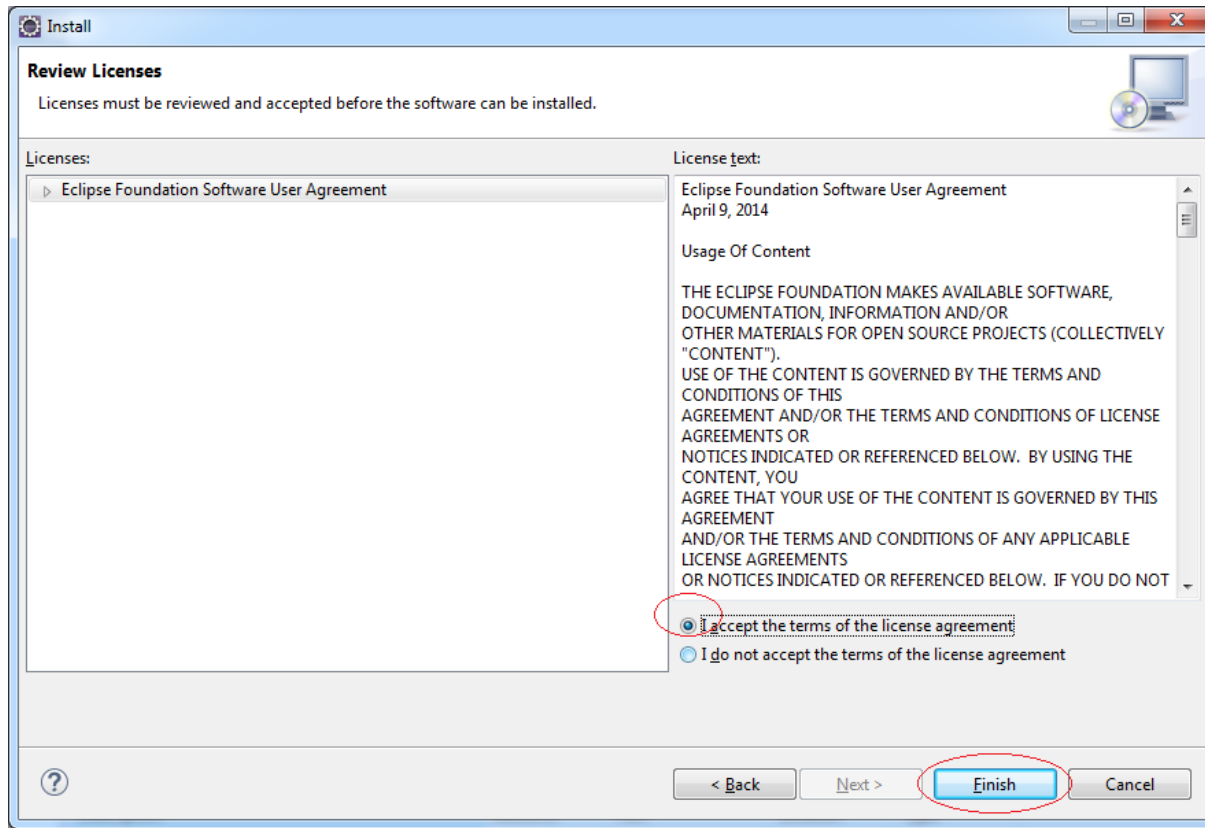
Saisissez Name: m2e

Location: <http://download.eclipse.org/technology/m2e/releases>

INSTALLEZ MAVEN DANS ECLIPSE



INSTALLEZ MAVEN DANS ECLIPSE



Vous avez réussi à installer Maven dans Eclipse , redémarrez Eclipse .

INSTALLEZ MAVEN DANS ECLIPSE

Si vous voulez installer Maven indépendamment

- Télécharger Maven <http://maven.apache.org/download.cgi>, la dernière version de Maven est 3.6.3, nous allons utiliser cette version.

Welcome

License

ABOUT MAVEN

What is Maven?

Features

Download

Use

Release Notes

DOCUMENTATION

Maven Plugins

Index (category)

User Centre

Plugin Developer Centre

Maven Central Repository

Maven Developer Centre

Books and Resources

Security

COMMUNITY

Community Overview

Project Roles

How to Contribute

Getting Help

Issue Management

Getting Maven Source

The Maven Team

PROJECT DOCUMENTATION

Project Information

Downloading Apache Maven 3.6.3

Apache Maven 3.6.3 is the latest release and recommended version for all users.

The currently selected download mirror is <https://downloads.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available. You may also consult the [complete list of mirrors](#).

Other mirrors: Change

System Requirements

Java Development Kit (JDK)	Maven 3.3+ require JDK 1.7 or above to execute - they still allow you to build against 1.3 and other JDK versions by Using Toolchains
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts and Windows batch files.

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.6.3-bin.tar.gz	apache-maven-3.6.3-bin.tar.gz.sha512	apache-maven-3.6.3-bin.tar.gz.asc
Binary zip archive	apache-maven-3.6.3-bin.zip	apache-maven-3.6.3-bin.zip.sha512	apache-maven-3.6.3-bin.zip.asc
Source tar.gz archive	apache-maven-3.6.3-src.tar.gz	apache-maven-3.6.3-src.tar.gz.sha512	apache-maven-3.6.3-src.tar.gz.asc

INSTALLEZ MAVEN DANS ECLIPSE

Après avoir téléchargé Maven, vous pouvez décompresser le fichier zip dans un dossier spécifique sur votre disque dur.

Dans l'étape suivante, vous allez déclarer des variables d'environnement pour maven .

`JAVA_HOME=C:\DevPrograms\Java\jdk1.8.0_40`

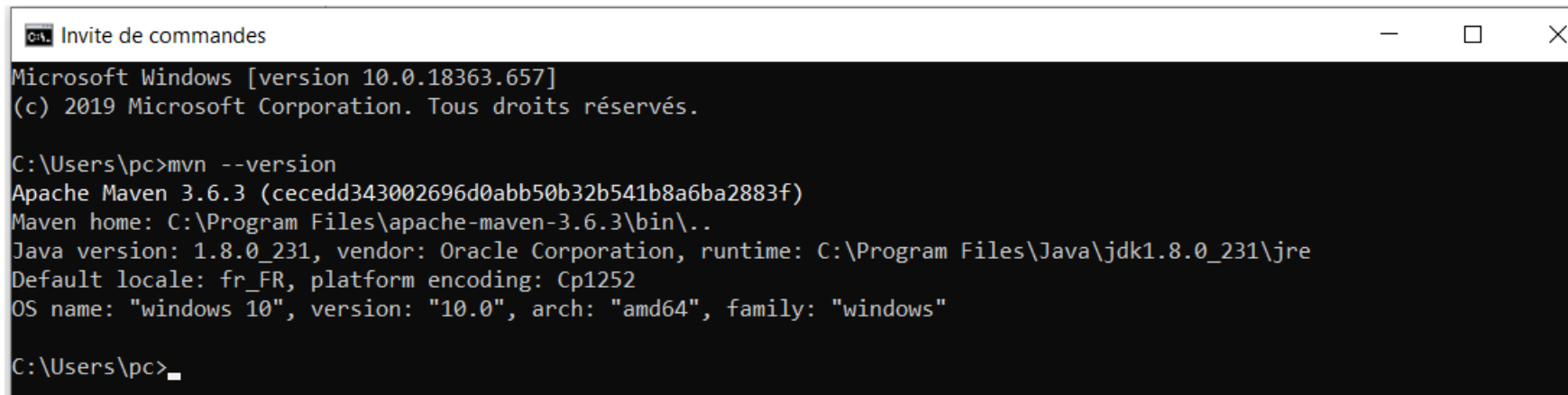
`M2_HOME=D:\DevPrograms\apache-maven-3.6.3`

`M2=%M2_HOME%\bin`

`Path=;%M2%;%JAVA_HOME%\bin`

Ensuite, vérifiez vos déclarations environnementales ont été réussies ou non

Ouvrez la fenêtre de Commande et tapez la ligne de commande :`mvn --version`



```
Invite de commandes
Microsoft Windows [version 10.0.18363.657]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\pc>mvn --version
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: C:\Program Files\apache-maven-3.6.3\bin\..
Java version: 1.8.0_231, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_231\jre
Default locale: fr_FR, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

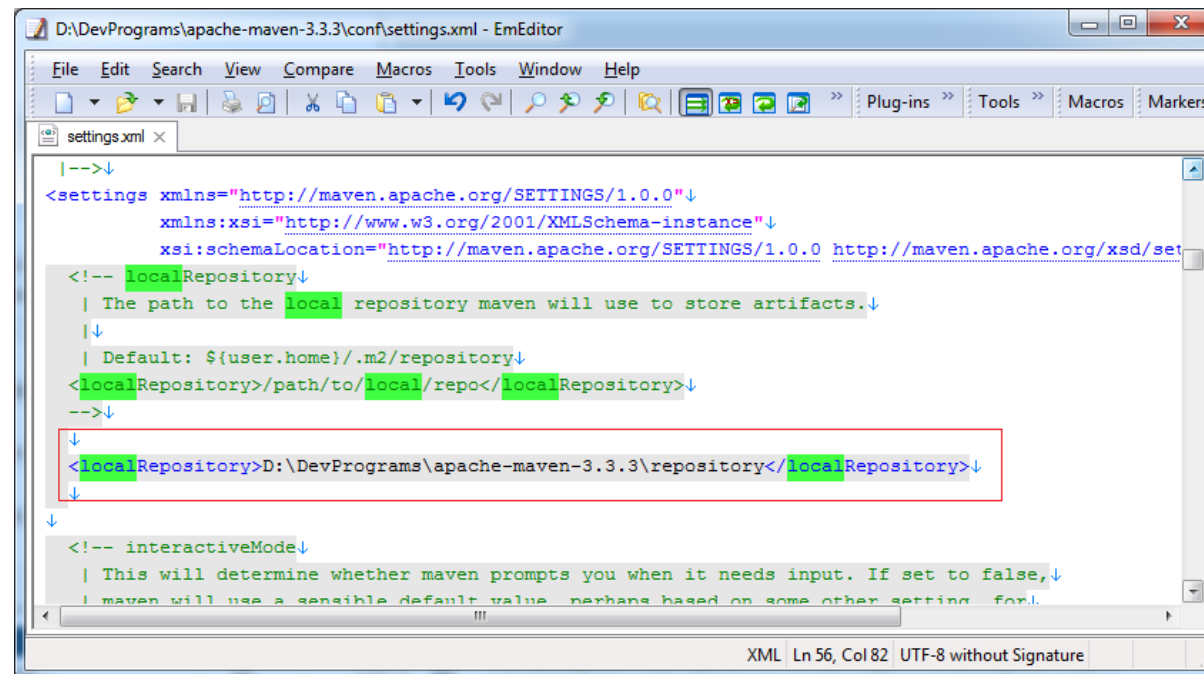
C:\Users\pc>
```

INSTALLEZ MAVEN DANS ECLIPSE

Maintenant, on va declare setup location Maven on Eclipse, Pour faire,

Ouvrez le fichier ...\\apache-maven-3.6.3\\conf**settings.xml** pour modifier certains paramètres

Ajoutez : <localRepository>D:\\DevPrograms\\apache-maven-3.3.3\\repository</localRepository>



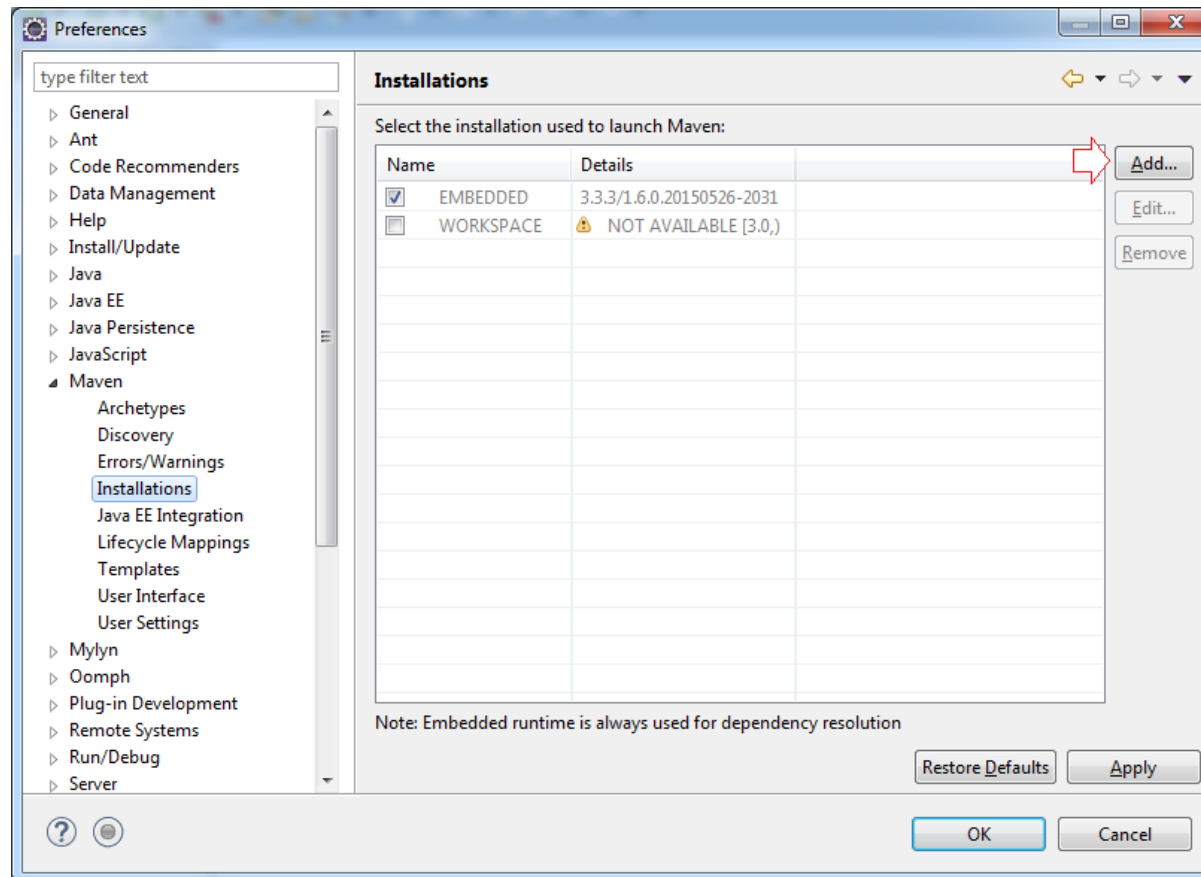
```

D:\\DevPrograms\\apache-maven-3.3.3\\conf\\settings.xml - EmEditor
File Edit Search View Compare Macros Tools Window Help
settings.xml x
<!-->
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/set
  <!-- localRepository
  | The path to the local repository maven will use to store artifacts.
  |
  | Default: ${user.home}/.m2/repository
  <localRepository>path/to/local/repo</localRepository>
  -->
  <localRepository>D:\\DevPrograms\\apache-maven-3.3.3\\repository</localRepository>
  <!-- interactiveMode
  | This will determine whether maven prompts you when it needs input. If set to false,
  | maven will use a sensible default value, perhaps based on some other setting for
  XML Ln 56, Col 82 UTF-8 without Signature

```


INSTALLEZ MAVEN DANS ECLIPSE

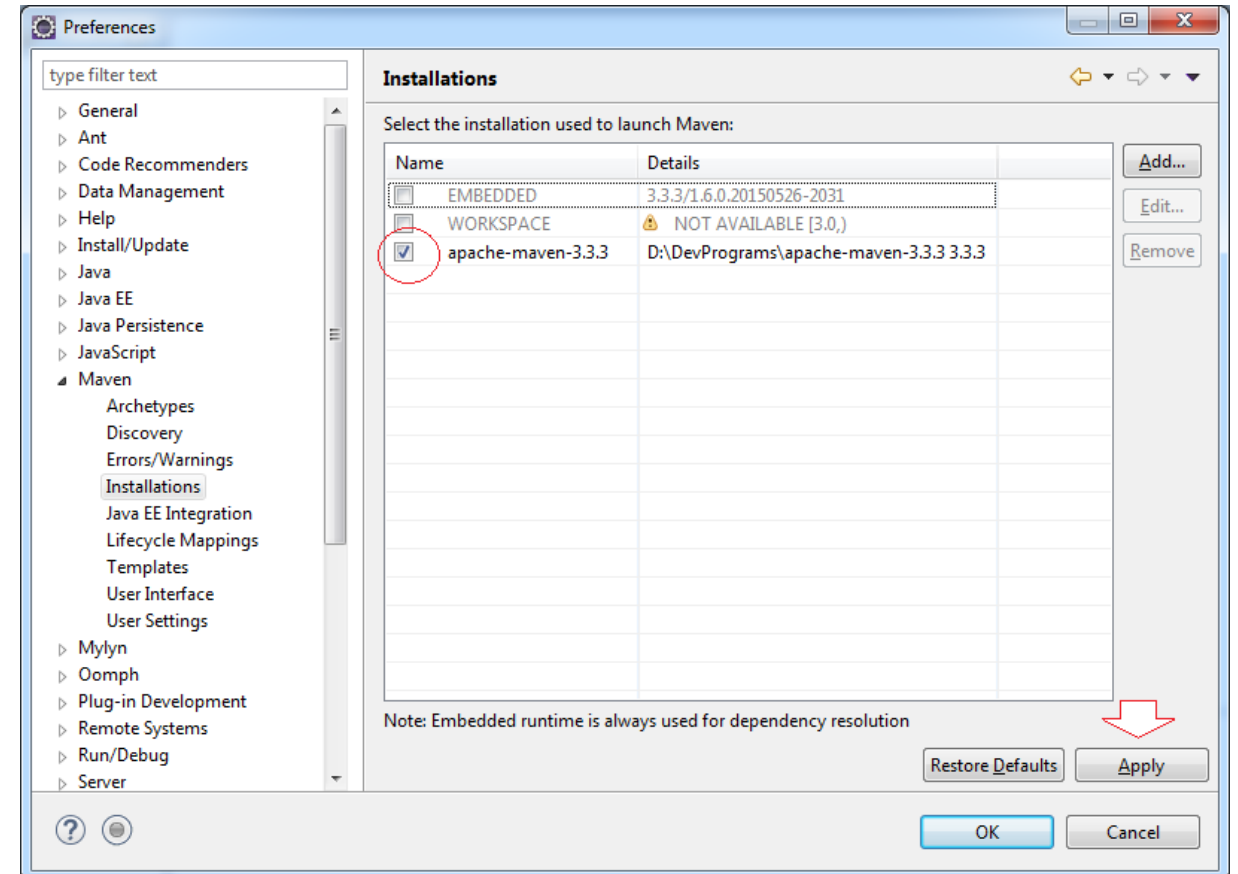
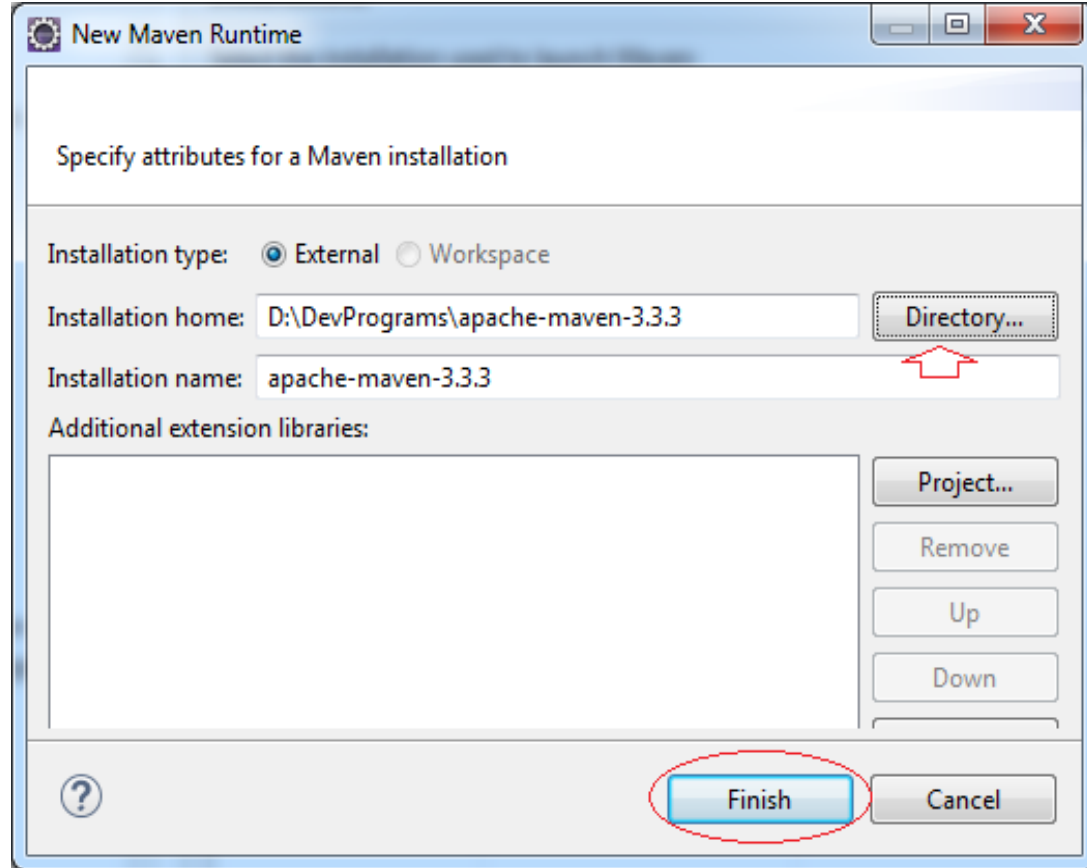
Sur Eclipse sélectionnez Window/Preference...



INSTALLEZ MAVEN DANS ECLIPSE

Sur Eclipse sélectionnez Window/Preference...

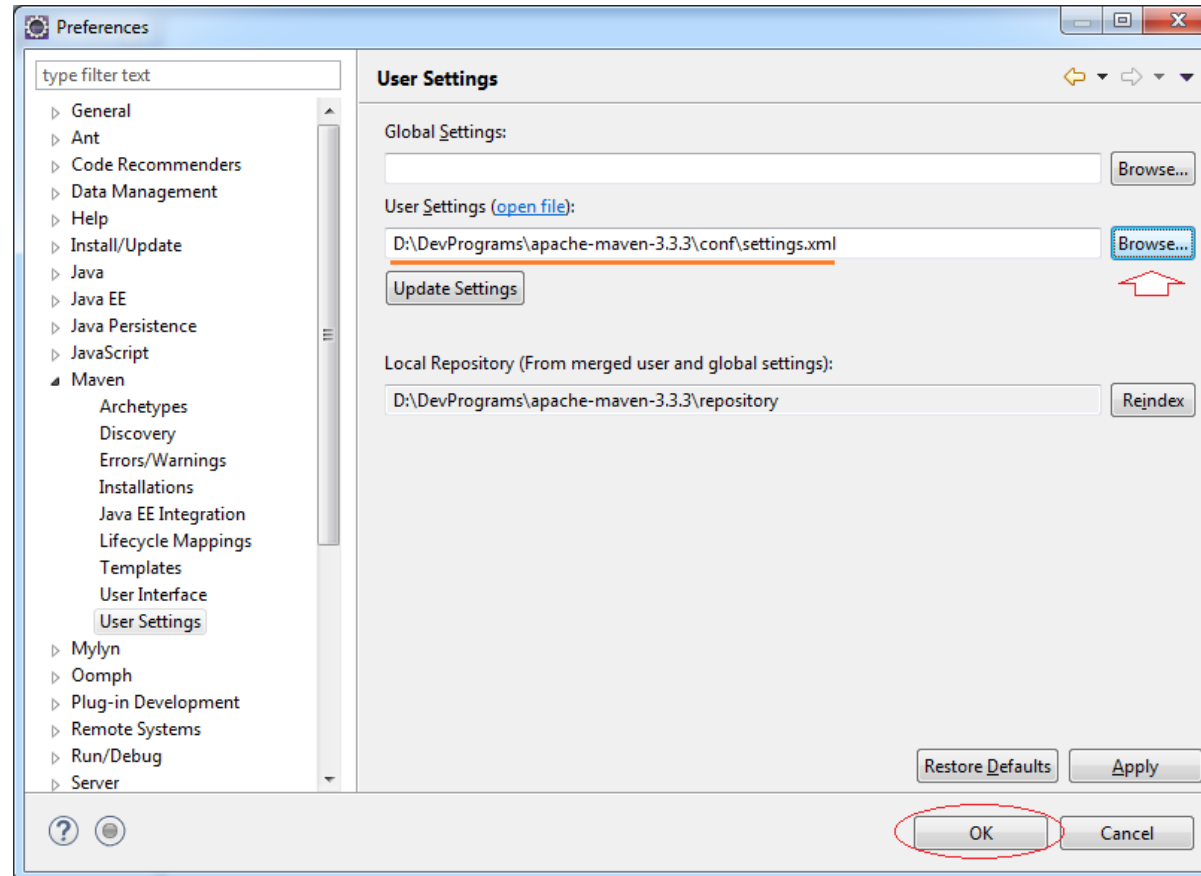
Cliquez sur Add pour ajouter une nouvelle configuration, pointez sur l'emplacement de l'installation de Maven .



INSTALLEZ MAVEN DANS ECLIPSE

Ensuite, déclarez la position du fichier de configuration de Maven . C'est le fichier que vous venez de changer précédemment.

Ici, l'installation de **Maven** pour **Eclipse** est terminée.

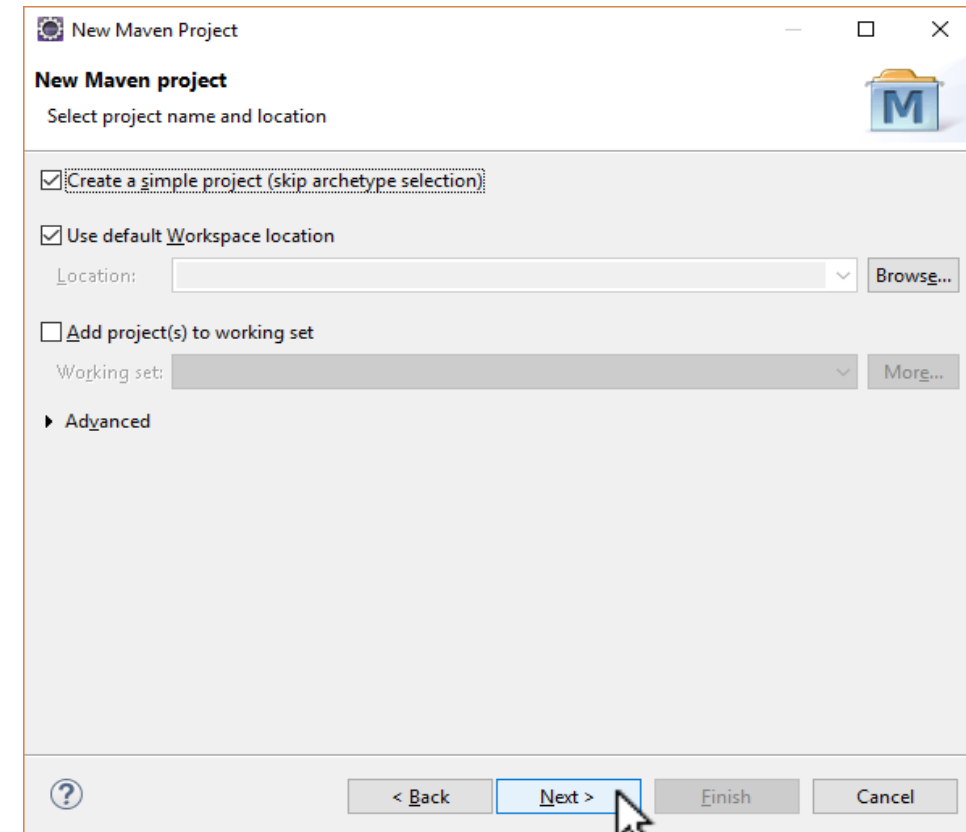


PREMIER PROJET MAVEN DANS ECLIPSE

Créer un projet Maven simple requiert tout de même un peu plus d'informations que la création d'un projet Java de base.

Le panneau de création du projet Maven nous demande essentiellement deux choses. Tout d'abord si nous souhaitons utiliser un archétype Maven. Un archétype Maven est en fait un modèle de projet, utile pour les projets un peu complexe ou nécessitant une configuration fastidieuse à créer. C'est le cas d'applications web, Java EE, Spring, etc... La seconde chose est l'endroit où l'on souhaite enregistrer notre projet. Il n'y a pas de raison pour nous de choisir un autre endroit que l'endroit par défaut.

Nous n'allons pas utiliser d'archétype, on coche donc la case qui permet de sauter la phase de sélection de l'archétype, et on laisse la localisation par défaut pour ce projet.



PREMIER PROJET MAVEN DANS ECLIPSE

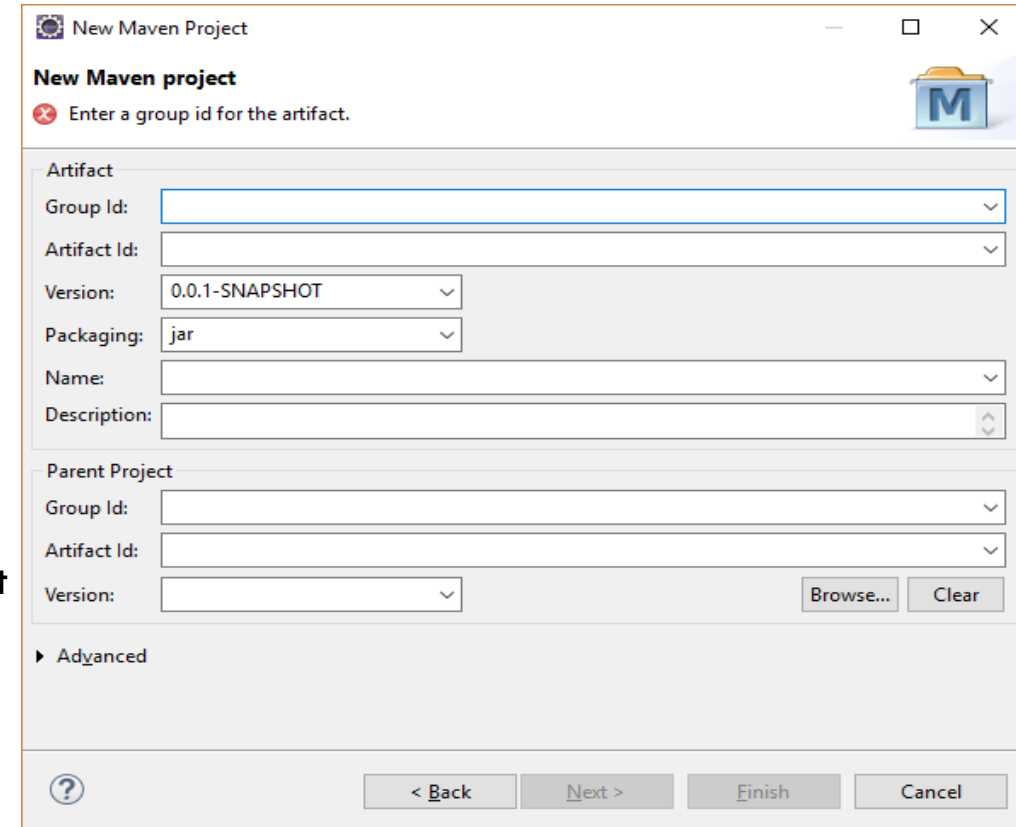
Une fois cette première étape passée, Maven nous demande des informations très importantes, dont il faut bien comprendre la signification.

Tout d'abord, il faut savoir que Maven reconnaît tous les projets qu'il gère par trois paramètres.

Le **groupid**, qui constitue la racine du nom de tous les projets gérés par Maven. Le groupid est unique pour une organisation qui crée des projets logiciels (entreprises, associations, etc...) et appartient en général à cette organisation. La façon naturelle pour choisir ce nom racine est de procéder comme pour le choix des noms de package : d'utiliser un nom de domaine que l'on possède.

Vient ensuite l'**artifactId**. Il s'agit du nom du projet proprement dit. Un projet ne change pas de nom ni de groupe, il garde donc le même groupid et artifactId tout au long de son existence. L'artifactId doit donc être choisi avec soin.

Le dernier paramètre est le numéro de version. Un numéro de version évolue dans la vie d'un projet. Un projet donné, repéré par un groupid et artifactId peut donc exister en plusieurs versions.



The screenshot shows the 'New Maven Project' dialog box in Eclipse. The title bar says 'New Maven Project'. Below the title bar, there's a section titled 'New Maven project' with a red error icon and the message 'Enter a group id for the artifact.' The dialog is divided into two main sections: 'Artifact' and 'Parent Project'. The 'Artifact' section contains fields for 'Group Id', 'Artifact Id', 'Version' (set to '0.0.1-SNAPSHOT'), 'Packaging' (set to 'jar'), 'Name', and 'Description'. The 'Parent Project' section contains fields for 'Group Id', 'Artifact Id', and 'Version', along with 'Browse...' and 'Clear' buttons. At the bottom, there's an 'Advanced' section that is currently collapsed. The bottom of the dialog has a help icon, '< Back', 'Next >', 'Finish', and 'Cancel' buttons.

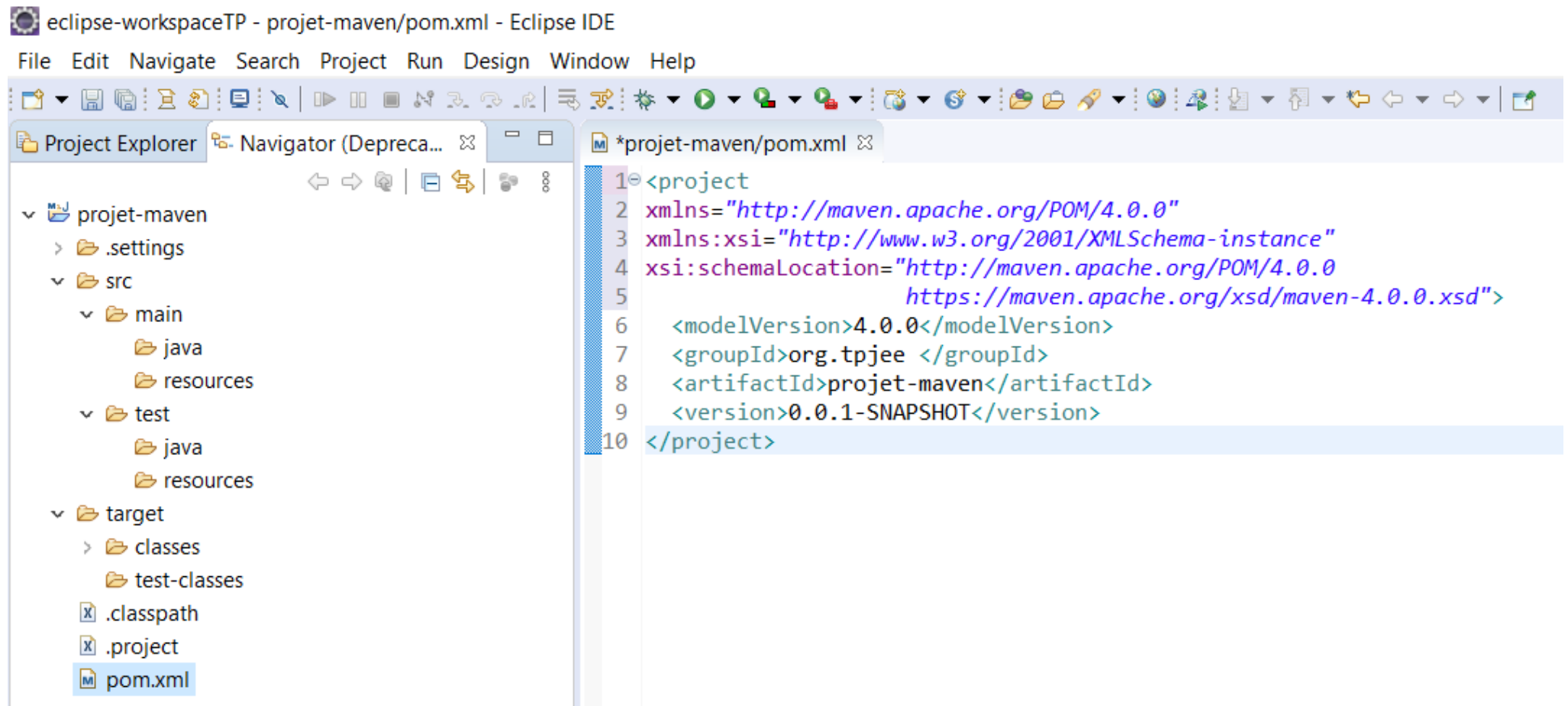
PREMIER PROJET MAVEN DANS ECLIPSE

Remplissons donc ce panneau avec nos valeurs afin de terminer la création de notre projet. On ne touche pas à l'élément packaging, qui est JAR pour les projets Java classiques.

Artifact	
Group Id:	<input type="text" value="org.tpjee"/>
Artifact Id:	<input type="text" value="projet-maven"/>
Version:	<input type="text" value="0.0.1-SNAPSHOT"/>
Packaging:	<input type="text" value="jar"/>
Name:	<input type="text"/>
Description:	<input type="text"/>
Parent Project	
Group Id:	<input type="text"/>
Artifact Id:	<input type="text"/>
Version:	<input type="text"/>
<input type="button" value="Browse..."/> <input type="button" value="Clear"/>	
Advanced	

PREMIER PROJET MAVEN DANS ECLIPSE

Notre projet créé apparaît donc dans la vue Package navigator d'Eclipse (Window/ShowView/Navigator) sous la forme suivante.

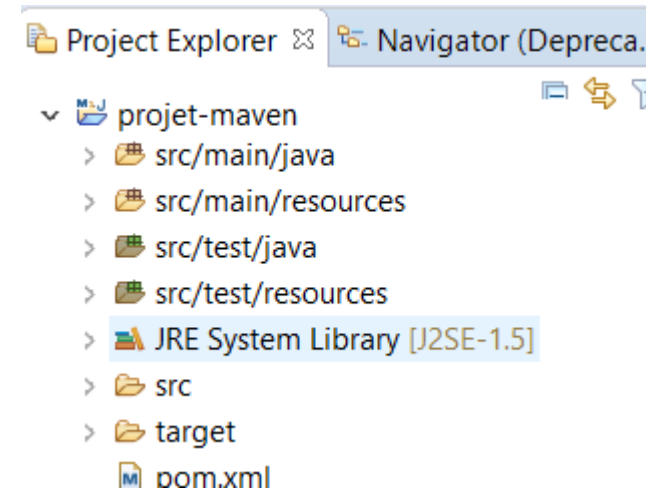


PREMIER PROJET MAVEN DANS ECLIPSE

Les remarques :

- Maven crée des projets Java avec une structure particulière.
- Il crée `main/java` et `src/test/java`, pour le code source proprement dit de notre projet et des tests. Les deux autres répertoires créés sont `src/main/resources` et `src/test/resources`. Ce sont des répertoires dans lequel on range est fichiers de ressources, différents du code.
- Le code source de notre application doit être créé dans `src/main/java`.
- Maven nous a ensuite créé un répertoire `target`. C'est le répertoire qu'il utilise pour produire tous les fichiers de notre projet. Ce sont les fichiers des classes compilées, les rapports de test, les fichiers JAR, WAR, etc...
- Maven nous a créé un fichier `pom.xml`. Ce fichier, appelé fichier POM (Project Object Model) est le fichier central de notre projet Maven, dans lequel Maven enregistre toutes les informations dont il a besoin. La structure complète d'un fichier POM peut être très complexe, nous n'en verrons que quelques aspects.

On peut aussi remarquer que Maven a configuré notre projet comme un projet Java 5. C'est le premier point que nous allons revoir, passer en Java 8



PREMIER PROJET MAVEN DANS ECLIPSE

La première chose à configurer est la version de Java que nous voulons utiliser, ainsi qu'un point de configuration.

On va ajouté les lignes suivantes à notre fichier POM.

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

PREMIER PROJET MAVEN DANS ECLIPSE

La première chose à configurer est la version de Java que nous voulons utiliser, ainsi qu'un point de configuration.

On va ajouté les lignes suivantes à notre fichier POM.

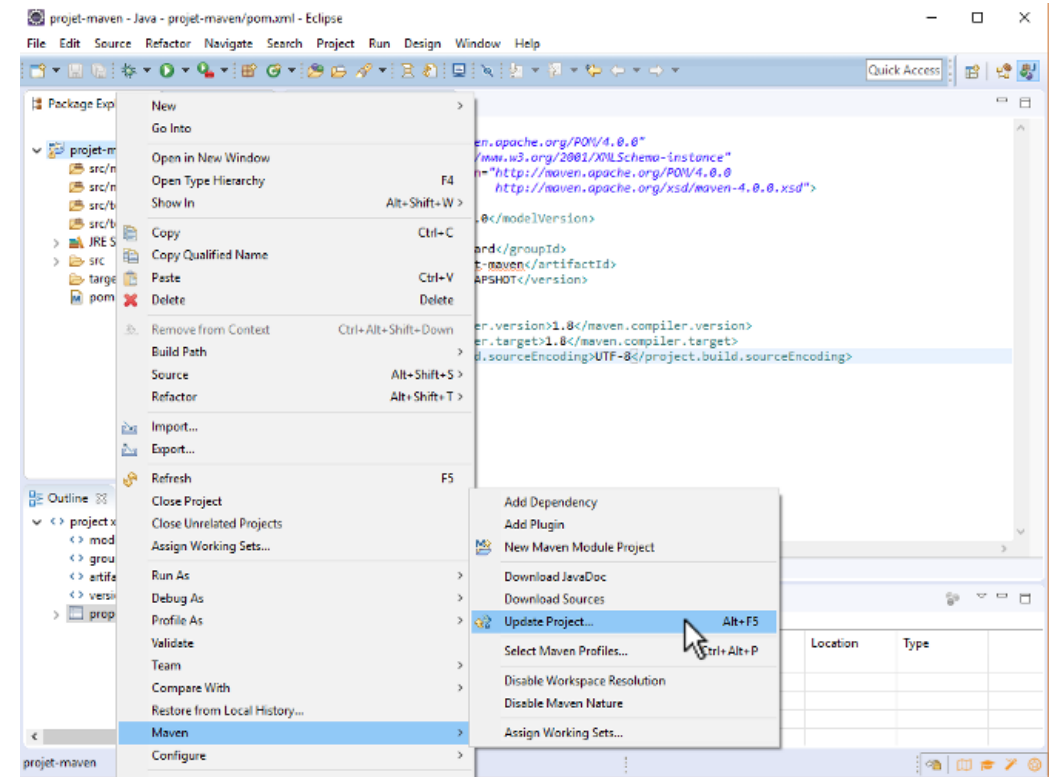
<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>



LES PHASES MAVEN

Maven associe à tout projet MAVEN des phases . A chacune de ces phases est associé/déclenché par défaut des tâches mises en oeuvre par des plugins.

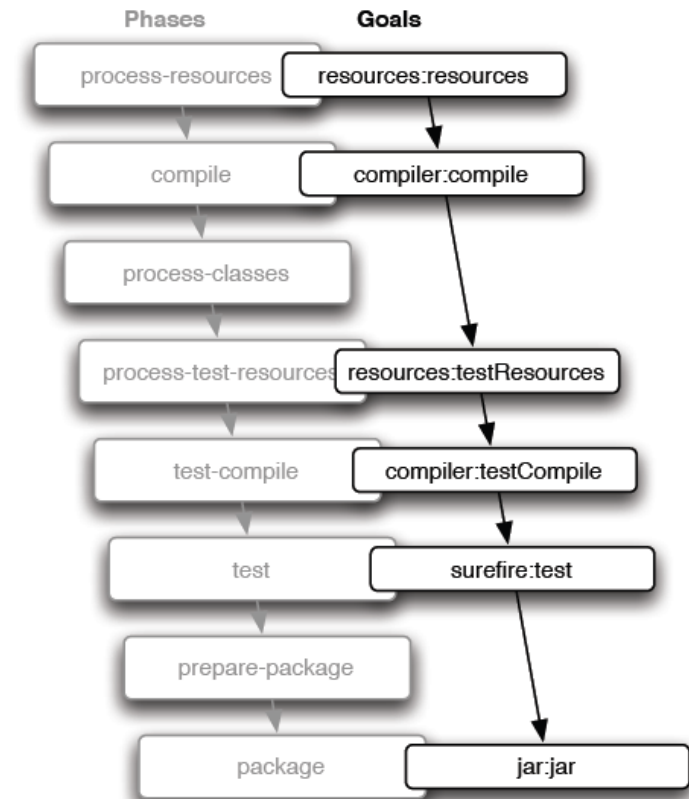
Dans le cycle de vie 'par défaut' d'un projet Maven, les phases les plus utilisées sont :

- **validate** : vérifie les prérequis d'un projet maven
- **compile** : compilation du code source
- **test** : lancement des tests unitaires
- **package** : assemble le code compilé en un livrable
- **install** : partage le livrable pour d'autres projets sur le même ordinateur
- **deploy** : publie le livrable pour d'autres projets dans un 'repository' distant

Les phases s'exécutent de façon séquentielle de façon à ce qu'une phase dépende de la phase précédente.

Nettoyage

mvn clean permet de vider le répertoire 'target'



AJOUT D'UNE DEPENDANCE A NOTRE PROJET

Une des premières utilités de Maven est de gérer automatiquement pour nous les dépendances de nos projets Java. Un projet industriel peut dépendre de dizaines, voire de centaines d'autres projets, certains développés en interne, d'autre en externe, qu'ils soient Open-source ou non.

Maven propose un mécanisme très pratique de déclaration de ces dépendances. Il gère le téléchargement automatique de ces dépendances dans un cache sur notre disque local, et la déclaration de ces dépendances dans Eclipse.

AJOUT D'UNE DEPENDANCE A NOTRE PROJET

Ajoutons le code suivant dans notre POM afin de configurer de la bibliothèque qu'on va utiliser:

```
<dependencies>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.3.2</version>
  </dependency>
</dependencies>
```

Cette déclaration indique à Maven que nous souhaitons mettre commons-lang3 en version 3.3.2 en dépendance de notre projet. Maven va commencer par regarder dans son cache local si cette dépendance s'y trouve. Si ce n'est pas le cas, il va interroger un repository de dépendances sur Internet (en général il s'agira de Maven Central : <http://search.maven.org/>, on peut également redéfinir ce point) afin de savoir si cette dépendance existe. Si tel est le cas, elle sera ajoutée au cache local.

AJOUT D'UNE DEPENDANCE A NOTRE PROJET

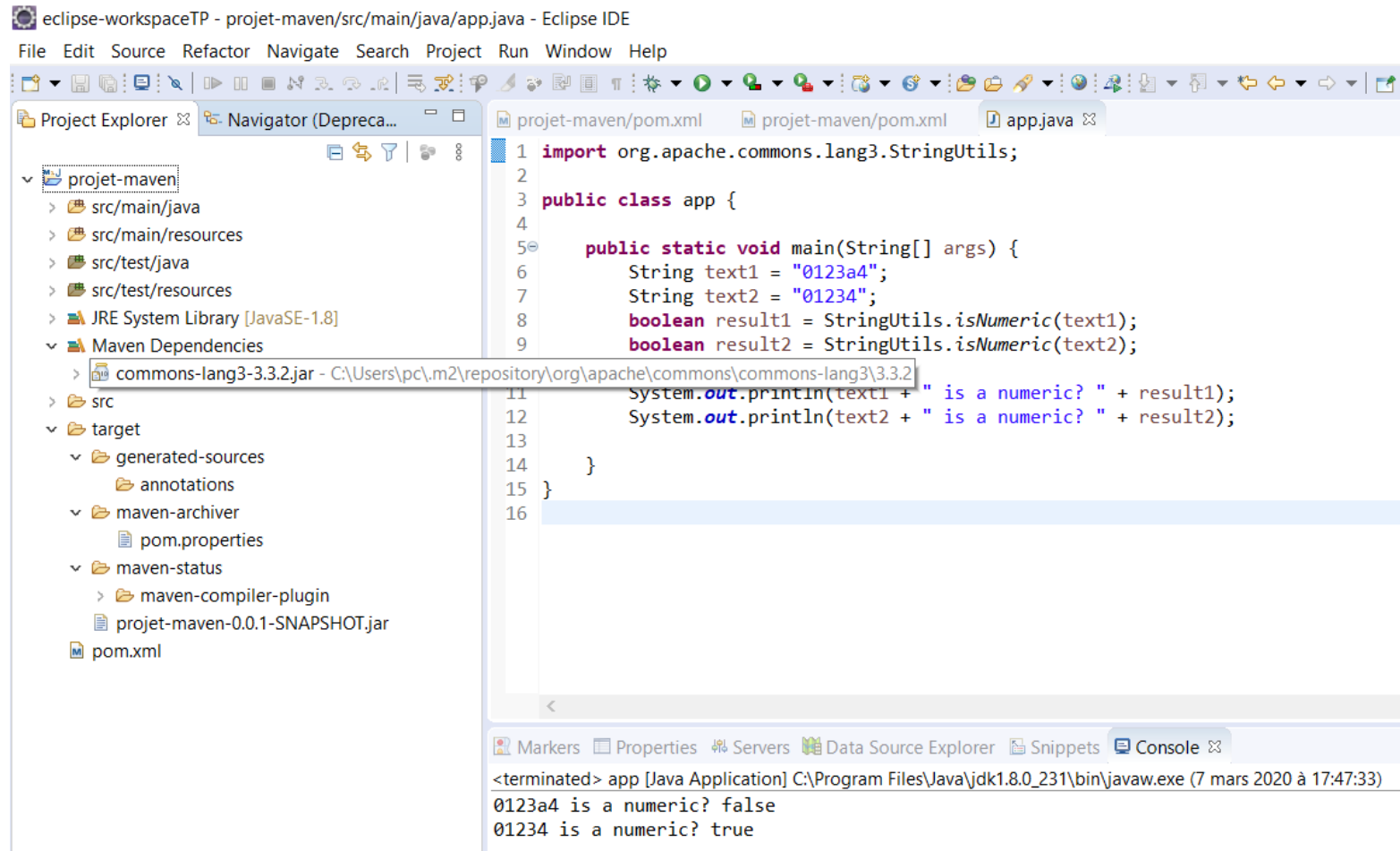
```
import org.apache.commons.lang3.StringUtils;

public class CheckNumeric {

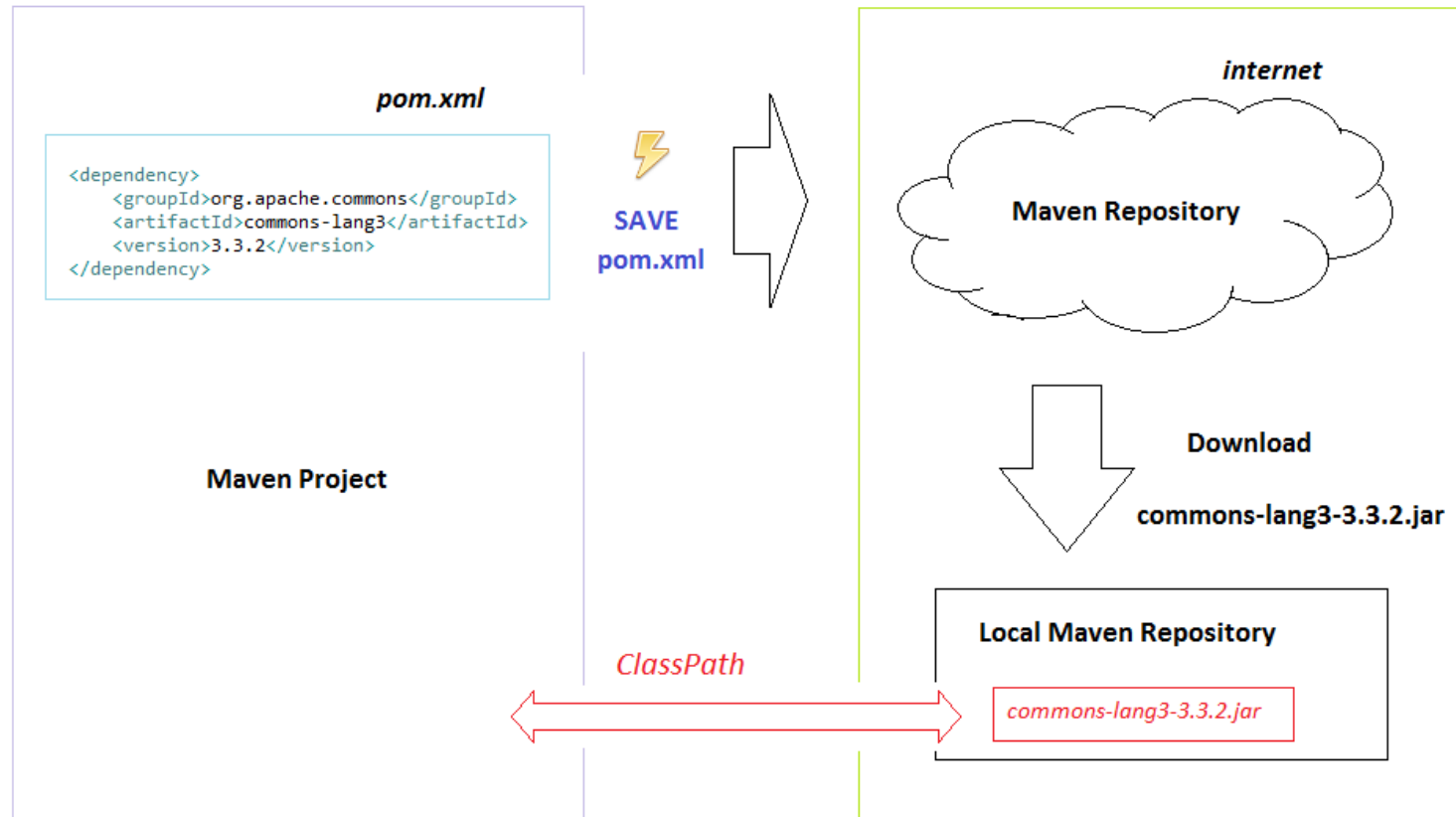
    public static void main(String[] args) {
        String text1 = "0123a4";
        String text2 = "01234";
        boolean result1 = StringUtils.isNumeric(text1);
        boolean result2 = StringUtils.isNumeric(text2);

        System.out.println(text1 + " is a numeric? " + result1);
        System.out.println(text2 + " is a numeric? " + result2);
    }
}
```

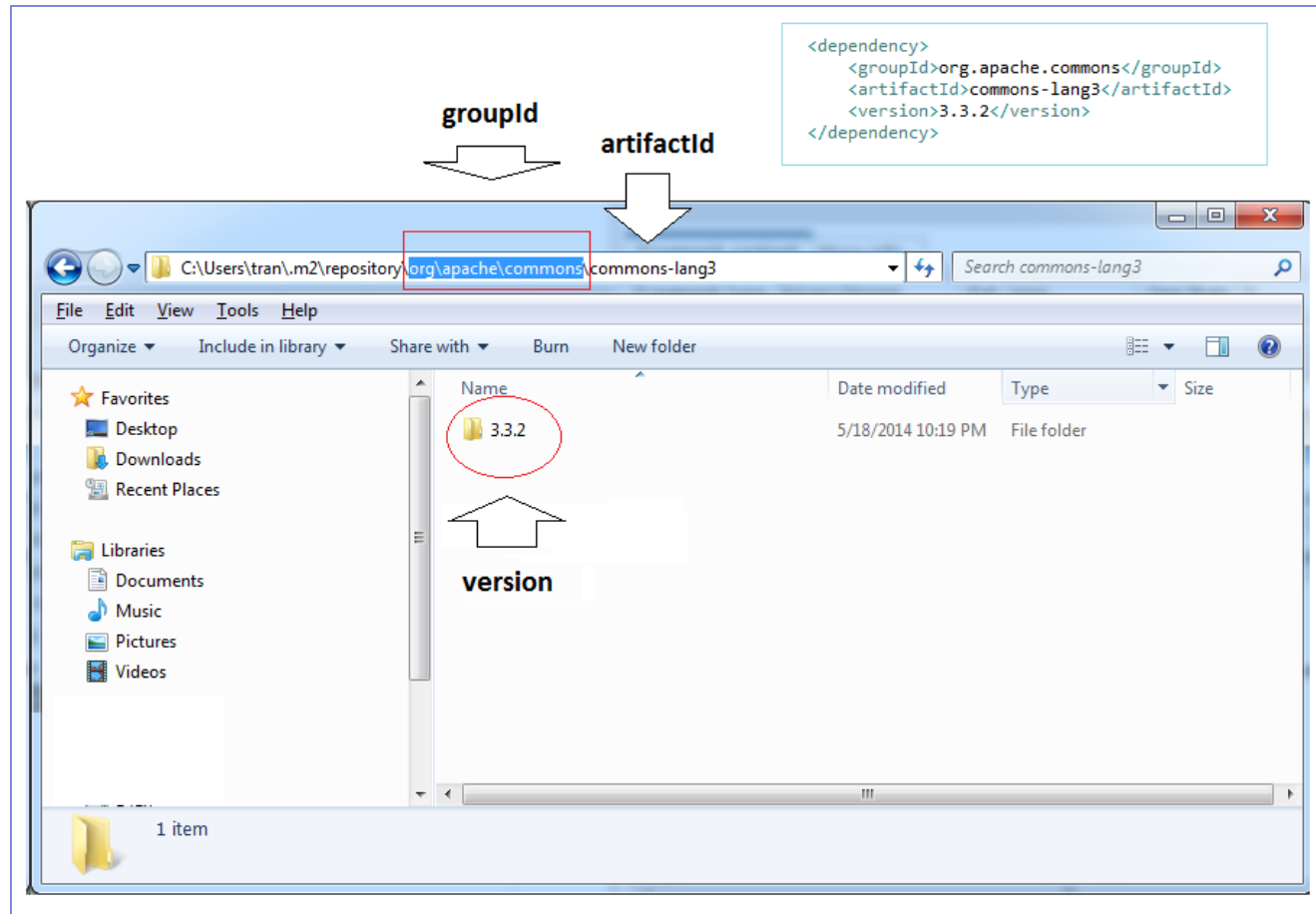
AJOUT D'UNE DEPENDANCE A NOTRE PROJET



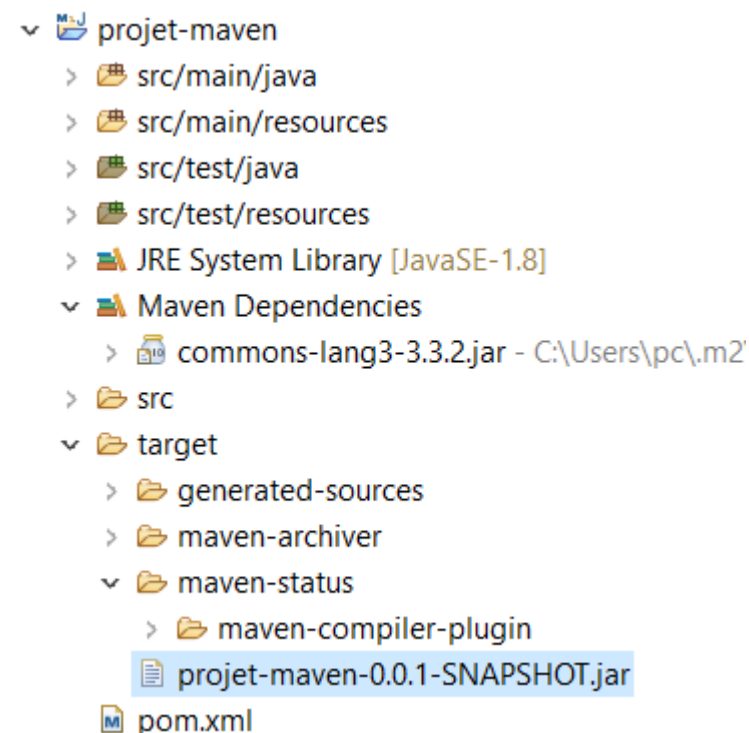
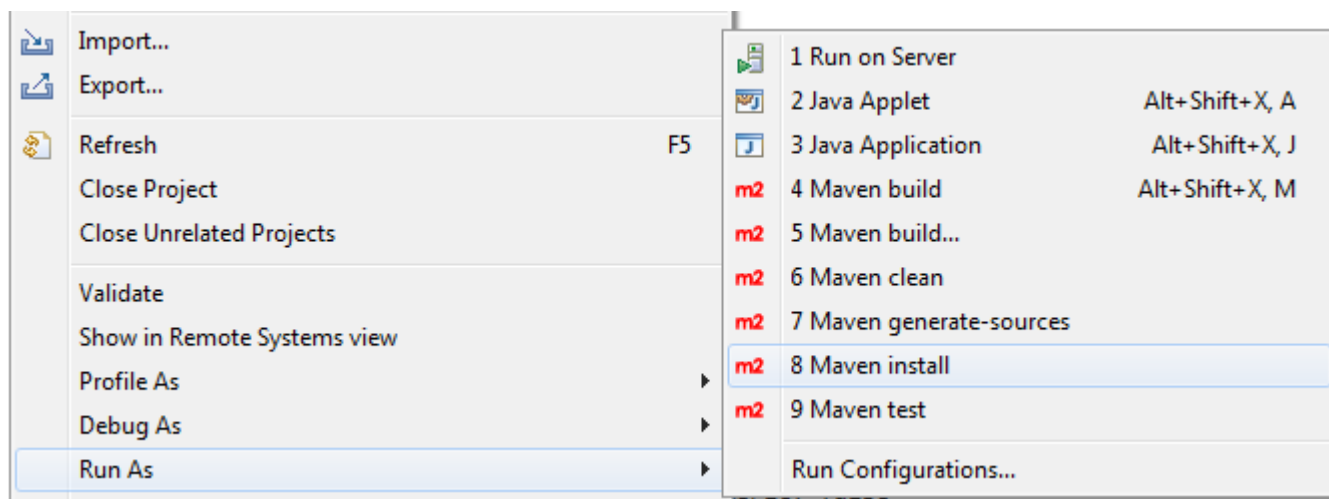
AJOUT D'UNE DEPENDANCE A NOTRE PROJET



AJOUT D'UNE DEPENDANCE A NOTRE PROJET



L'INSTALLATION LE PROJET AVEC MAVEN



Voir le projet que vous avez emballé sur le **Local Repository**:

> .m2 > repository > org > tpjee > projet-maven > 0.0.1-SNAPSHOT

Nom	Modifié le	Type	Taille
_remote.repositories	07/03/2020 17:50	Fichier REPOSITOR...	1 Ko
maven-metadata-local	07/03/2020 17:50	Document XML	1 Ko
projet-maven-0.0.1-SNAPSHOT	07/03/2020 15:58	Executable Jar File	3 Ko
pro	07/03/2020 15:27	Fichier POM	1 Ko

Type : Executable Jar File
Taille : 2,99 Ko
Modifié le : 07/03/2020 15:58

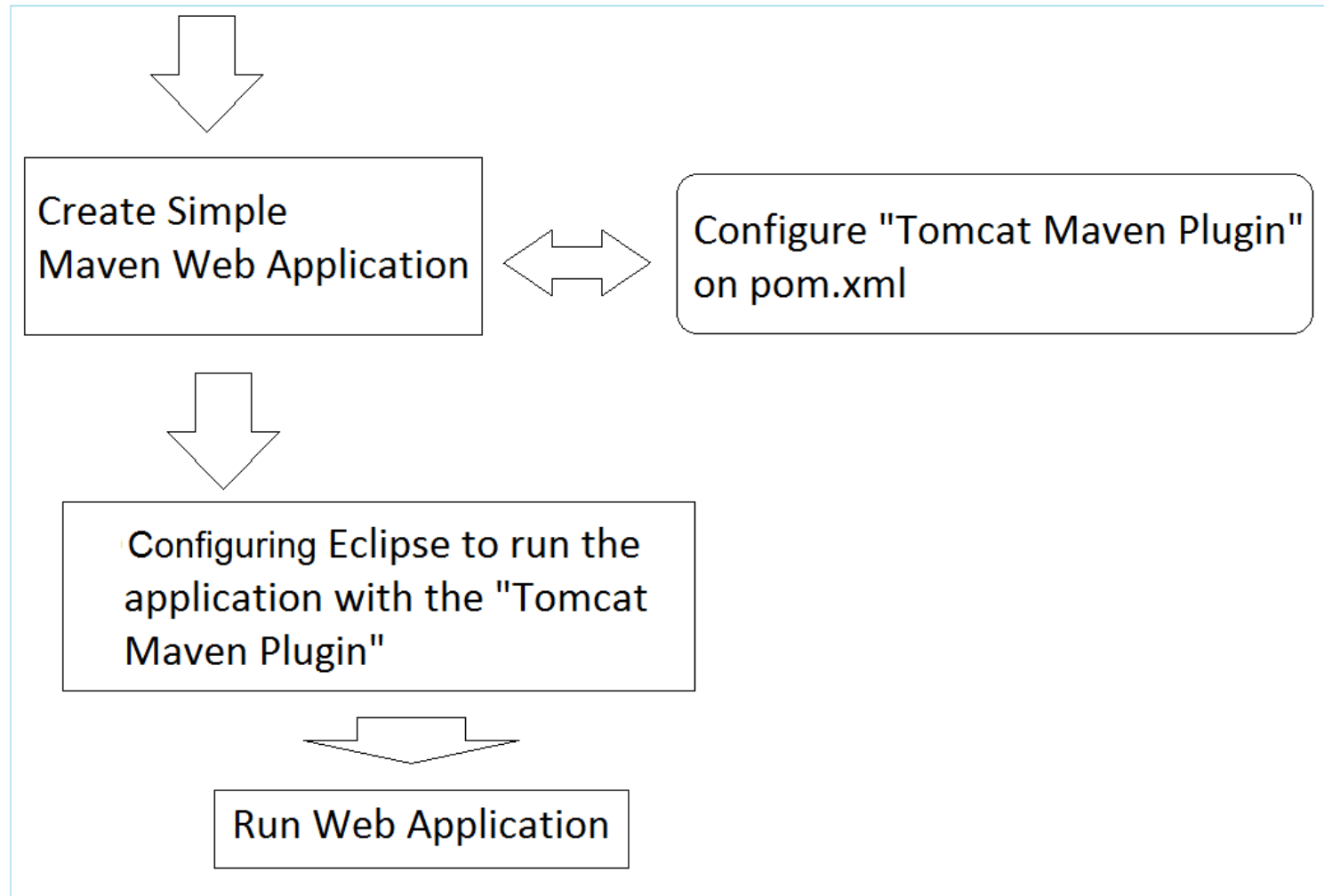
RÉPERTOIRE MAVEN SUR INTERNET

La question est de savoir où rechercher les informations groupId, artifactId, version.

Vous pouvez consulter un des deux sites:

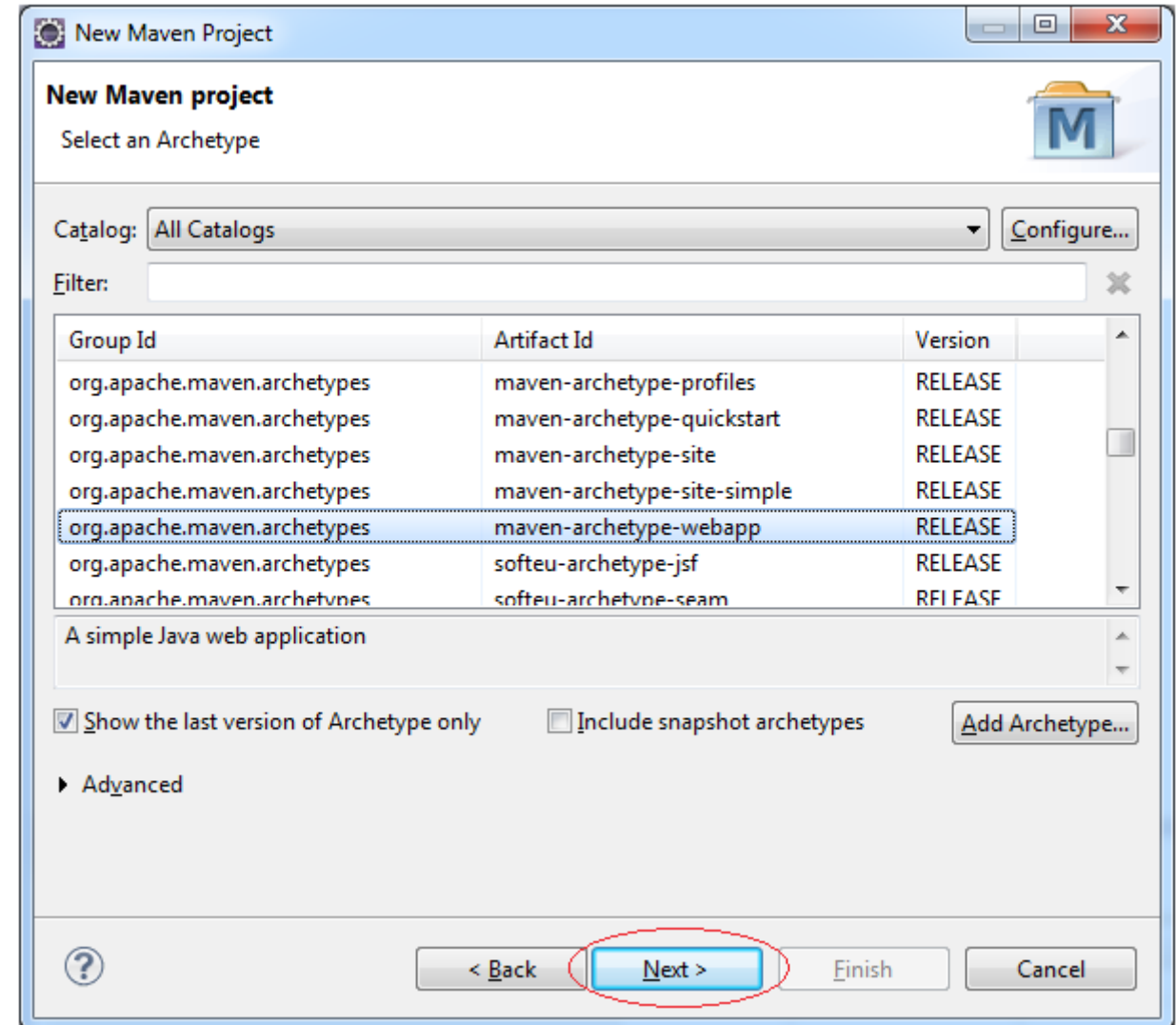
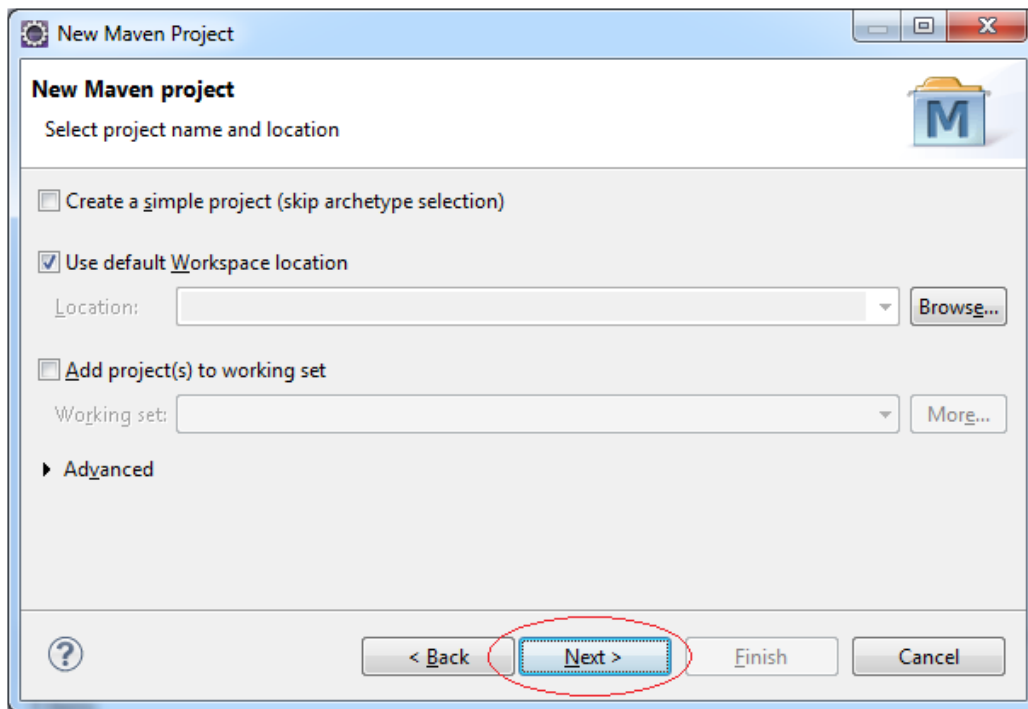
- <http://search.maven.org>
- <http://mvnrepository.com>

EXÉCUTEZ L'APPLICATION WEB JAVA MAVEN DANS TOMCAT MAVEN PLUGIN



EXÉCUTEZ L'APPLICATION WEB JAVA MAVEN DANS TOMCAT MAVEN PLUGIN

Créer Maven Webapp Project



EXÉCUTEZ L'APPLICATION WEB JAVA MAVEN DANS TOMCAT MAVEN PLUGIN

Pour **importing maven remote archetype catalogs in eclipse**

Cliquez sur Windows -> Preferences -> Maven -> Archetypes

Click on Add Remote Catalog button.

Remplissez catalog file : "<https://repo1.maven.org/maven2/archetype-catalog.xml>"

description: **archetype catalogs**

EXÉCUTEZ L'APPLICATION WEB JAVA MAVEN DANS TOMCAT MAVEN PLUGIN

Créer Maven Webapp Project

New Maven Project

New Maven project

Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

Properties available from archetype:

Name	Value

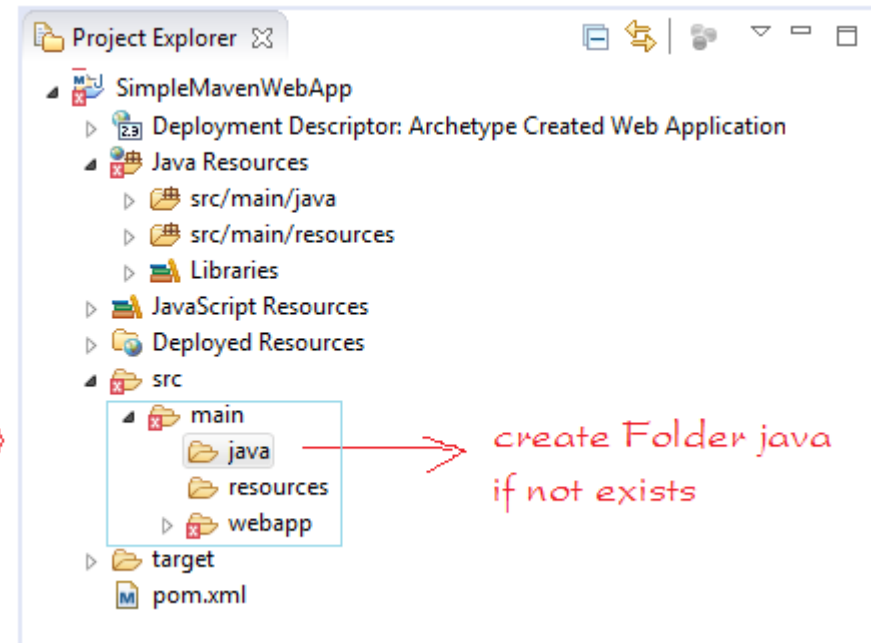
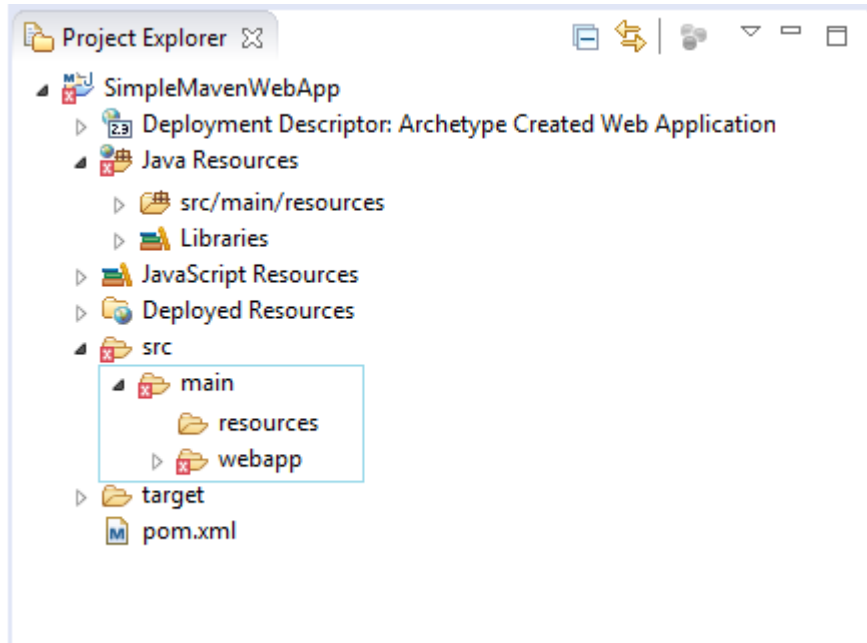
► **Advanced**

EXÉCUTEZ L'APPLICATION WEB JAVA MAVEN DANS TOMCAT MAVEN PLUGIN

Ne vous inquiétez pas du message d'erreur lorsque le projet a été créé. La raison en est que vous ne déclarez pas cette bibliothèque Servlet.

Eclipse crée la structure du projet Maven qui peut être fausse. Vous devez le fixer.

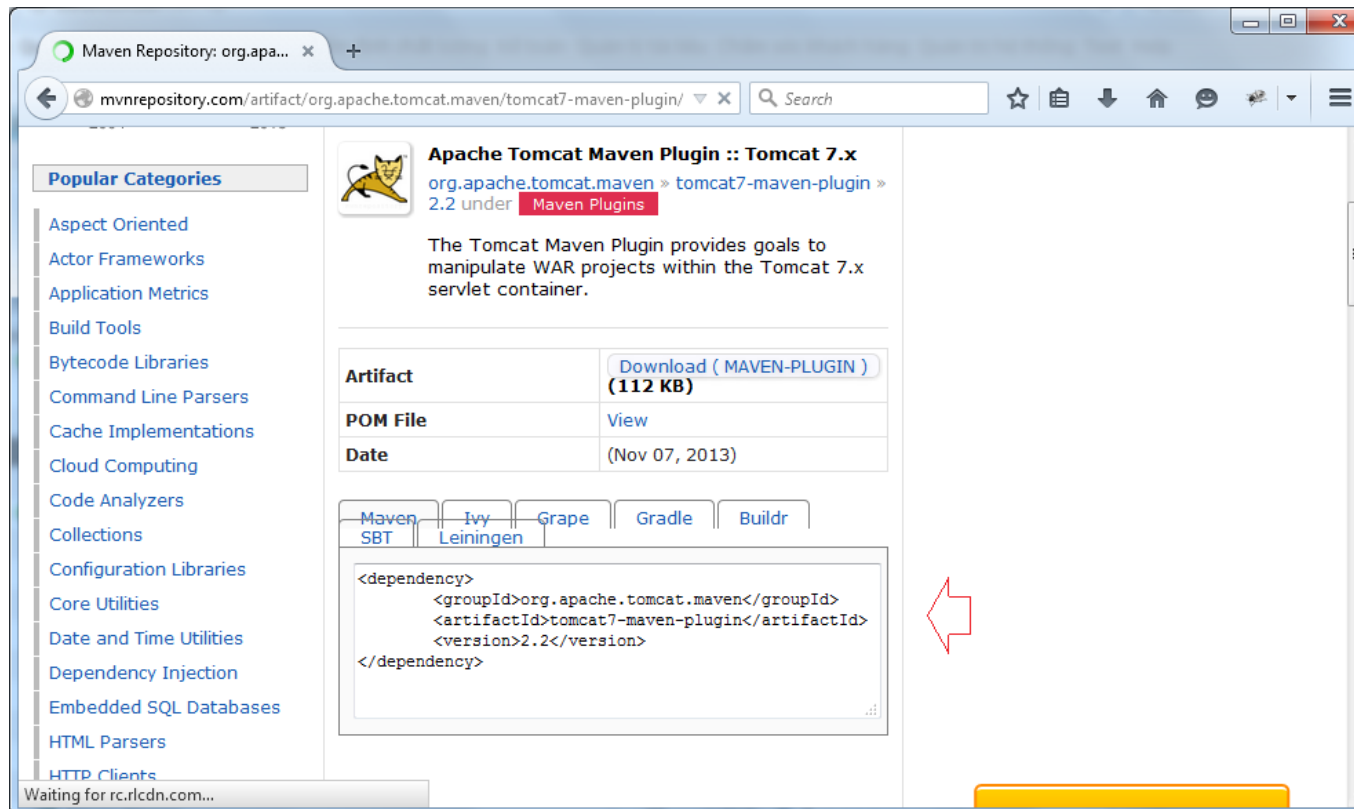
- SimpleMavenWebApp
 - Deployment Descriptor: Archetype
 - Java Resources
 - JavaScript Resources
 - Deployed Resources
 - src
 - target
 - pom.xml



CHERCHER TOMCAT MAVEN PLUGIN POUR UTILISER

Ensuite, nous chercherons une version de "Tomcat Maven Plugin" (convenante ou plus récente) :

<http://mvnrepository.com/artifact/org.apache.tomcat.maven/tomcat7-maven-plugin>



The screenshot shows the Maven Repository website for the Apache Tomcat Maven Plugin. The browser address bar displays the URL: mvnrepository.com/artifact/org.apache.tomcat.maven/tomcat7-maven-plugin. The page features a sidebar with "Popular Categories" and a main content area with the following details:

- Artifact:** [Download \(MAVEN-PLUGIN \) \(112 KB\)](#)
- POM File:** [View](#)
- Date:** (Nov 07, 2013)

Below the artifact information, there are tabs for different build systems: Maven, Ivy, Grape, Gradle, and Buildr. The Maven tab is selected, showing the following XML snippet:

```
<dependency>
  <groupId>org.apache.tomcat.maven</groupId>
  <artifactId>tomcat7-maven-plugin</artifactId>
  <version>2.2</version>
</dependency>
```

A red arrow points to the XML snippet, indicating the dependency to be added to the project's POM file.

CONFIGUREZ MAVEN

Configurez de la bibliothèque Servlet de Maven:

```
<!-- Servlet Library -->
```

```
<dependency>
```

```
  <groupId>javax.servlet</groupId>
```

```
  <artifactId>javax.servlet-api</artifactId>
```

```
  <version>3.1.0</version>
```

```
  <scope>provided</scope>
```

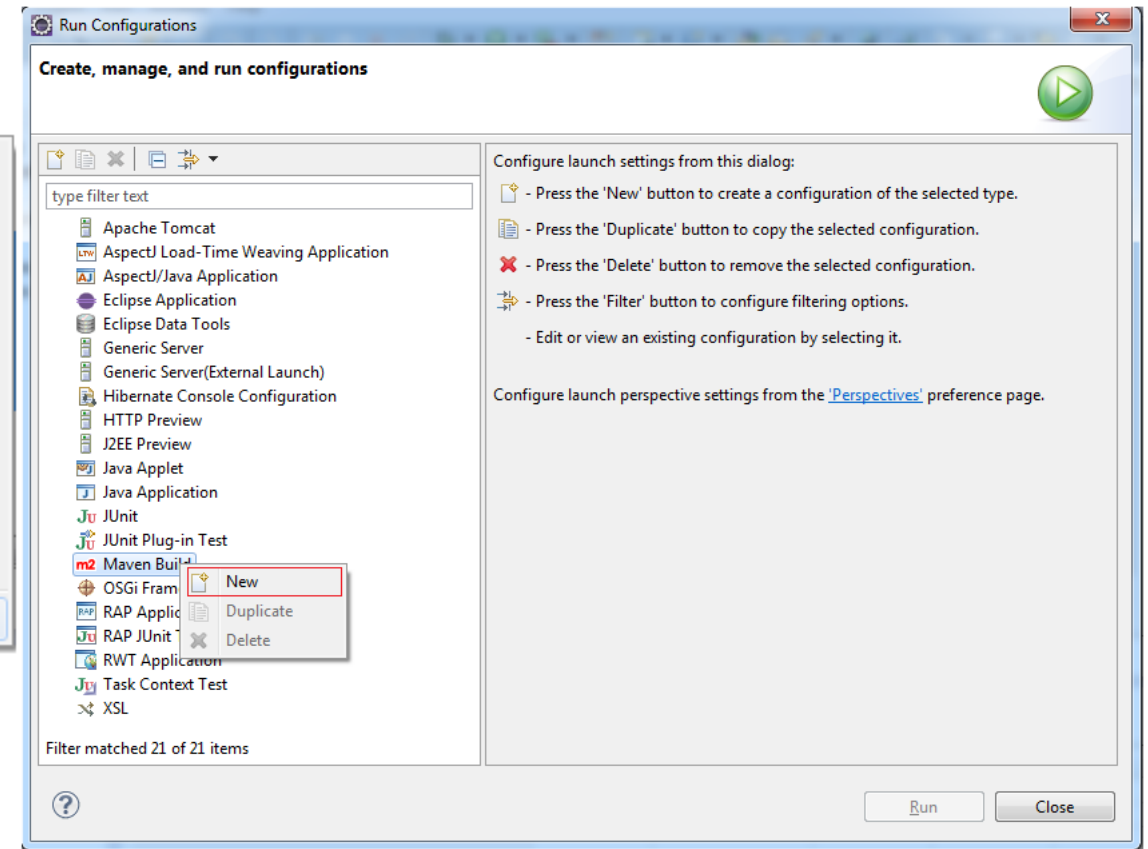
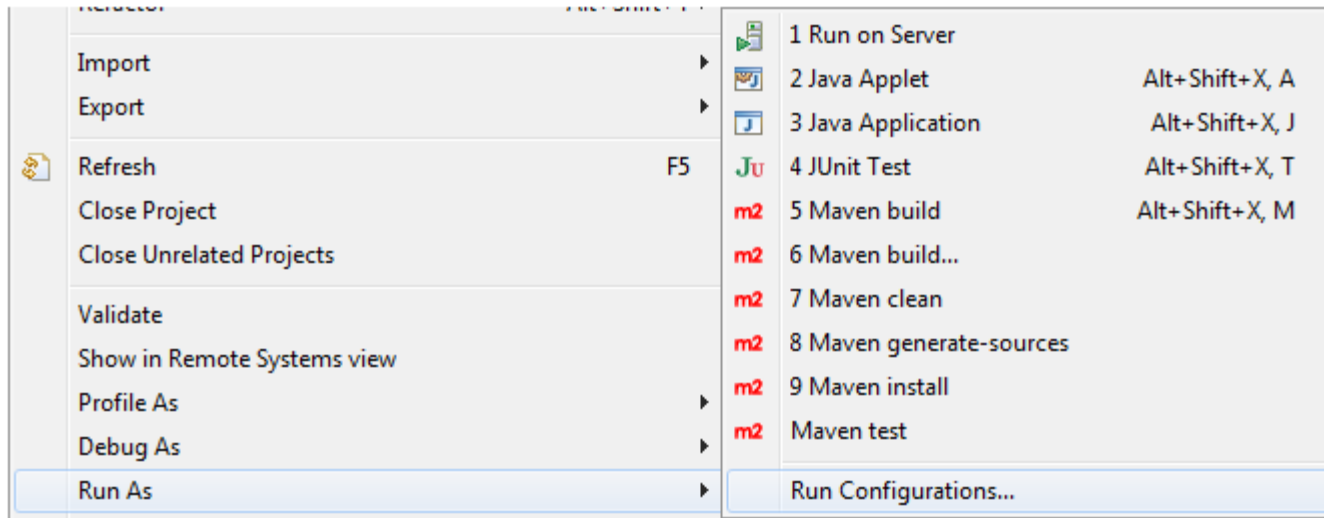
```
</dependency>
```

CONFIGUREZ MAVEN

Copiez et collez le code ci-dessous au pom.xml pour configurer le plugin Tomcat

```
<build>
  <finalName>SimpleMavenWebApp</finalName>
  <plugins>
    <!-- Config: Maven Tomcat Plugin -->
    <!-- http://mvnrepository.com/artifact/org.apache.tomcat.maven/tomcat7-maven-plugin -->
    <plugin>
      <groupId>org.apache.tomcat.maven</groupId>
      <artifactId>tomcat7-maven-plugin</artifactId>
      <version>2.2</version>
      <!-- Config: contextPath and Port (Default - /SimpleMavenWebApp : 8080) -->
      <!-- <configuration>
        <path>/</path>
        <port>8899</port>
      </configuration> -->
    </plugin>
  </plugins>
</build>
```

CRÉEZ UNE NOUVELLE CONFIGURATION D'EXÉCUTION



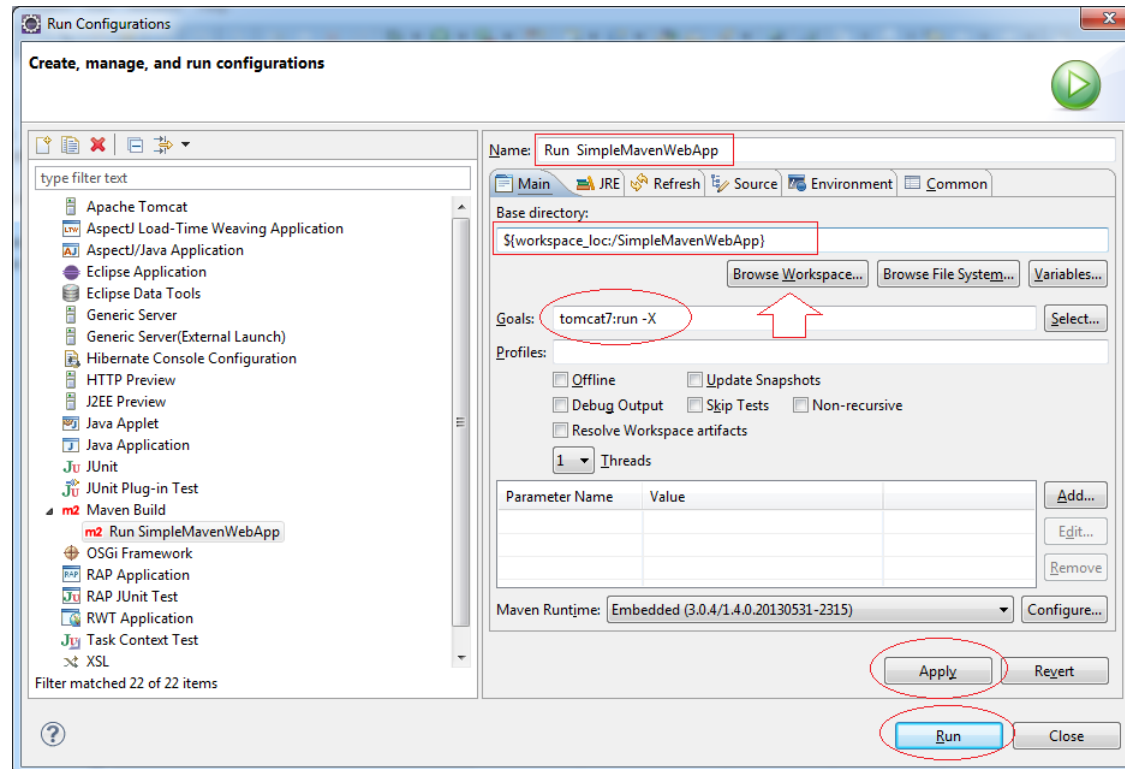
CRÉEZ UNE NOUVELLE CONFIGURATION D'EXÉCUTION

Saisissez les informations comme l'illustration montrée ci-dessous, cliquez sur Apply et Run.

Run SimpleMavenWebApp

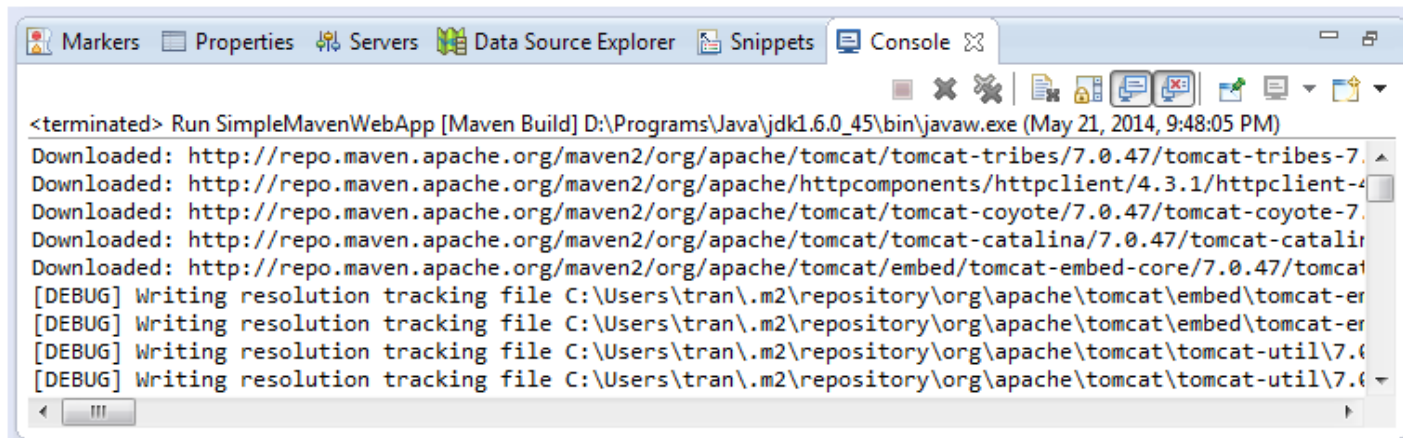
`${workspace_loc:/SimpleMavenWebApp}`

`tomcat7:run -X`

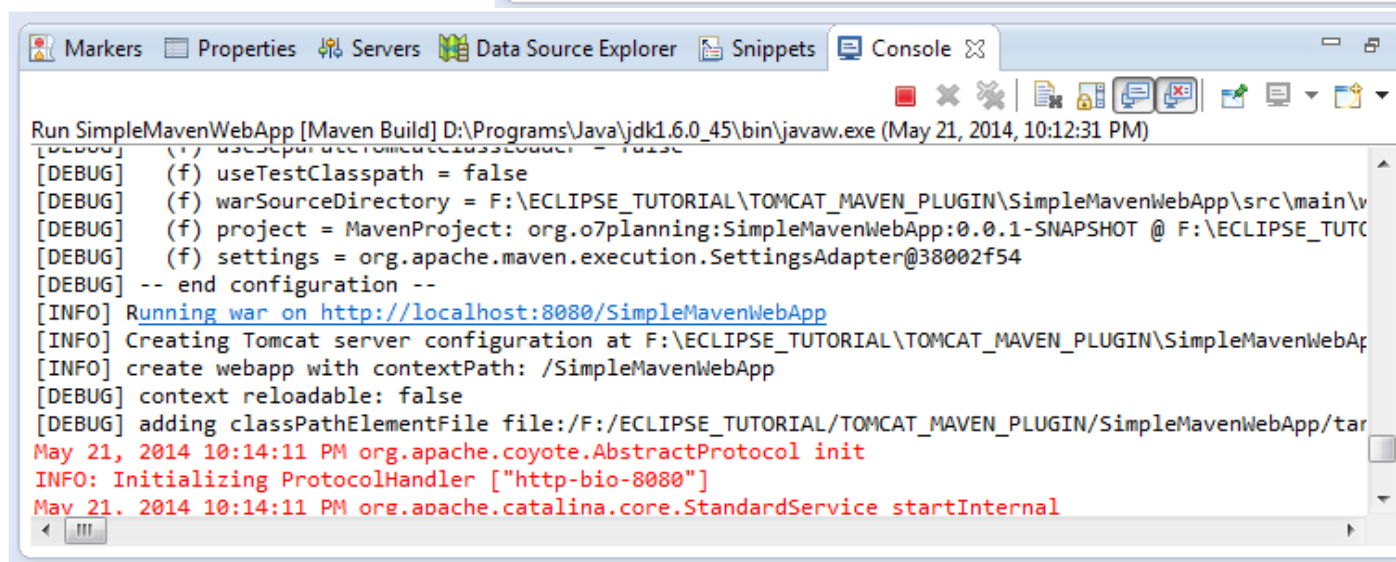


CRÉEZ UNE NOUVELLE CONFIGURATION D'EXÉCUTION

Dans la première exécution, Eclipse téléchargera "Tomcat Maven Plugin", donc vous devez attendre jusqu'à quand le téléchargement sera terminé.



```
<terminated> Run SimpleMavenWebApp [Maven Build] D:\Programs\Java\jdk1.6.0_45\bin\javaw.exe (May 21, 2014, 9:48:05 PM)
Downloaded: http://repo.maven.apache.org/maven2/org/apache/tomcat/tomcat-tribes/7.0.47/tomcat-tribes-7.0.47.jar
Downloaded: http://repo.maven.apache.org/maven2/org/apache/httpcomponents/httpclient/4.3.1/httpclient-4.3.1.jar
Downloaded: http://repo.maven.apache.org/maven2/org/apache/tomcat/tomcat-coyote/7.0.47/tomcat-coyote-7.0.47.jar
Downloaded: http://repo.maven.apache.org/maven2/org/apache/tomcat/tomcat-catalina/7.0.47/tomcat-catalina-7.0.47.jar
Downloaded: http://repo.maven.apache.org/maven2/org/apache/tomcat/embed/tomcat-embed-core/7.0.47/tomcat-embed-core-7.0.47.jar
[DEBUG] Writing resolution tracking file C:\Users\tran\.m2\repository\org\apache\tomcat\embed\tomcat-embed-core\7.0.47\tomcat-embed-core-7.0.47.jar
[DEBUG] Writing resolution tracking file C:\Users\tran\.m2\repository\org\apache\tomcat\embed\tomcat-embed-core\7.0.47\tomcat-embed-core-7.0.47.jar
[DEBUG] Writing resolution tracking file C:\Users\tran\.m2\repository\org\apache\tomcat\tomcat-util\7.0.47\tomcat-util-7.0.47.jar
[DEBUG] Writing resolution tracking file C:\Users\tran\.m2\repository\org\apache\tomcat\tomcat-util\7.0.47\tomcat-util-7.0.47.jar
```

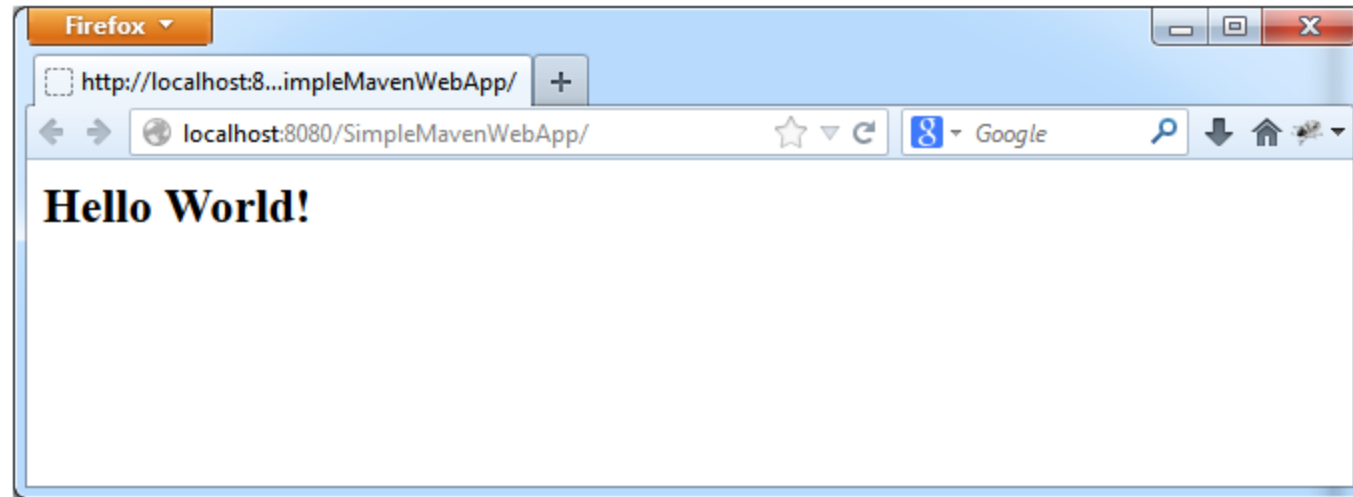


```
Run SimpleMavenWebApp [Maven Build] D:\Programs\Java\jdk1.6.0_45\bin\javaw.exe (May 21, 2014, 10:12:31 PM)
[DEBUG] (f) useSeparateTomcatClassLoader = false
[DEBUG] (f) useTestClasspath = false
[DEBUG] (f) warSourceDirectory = F:\ECLIPSE_TUTORIAL\TOMCAT_MAVEN_PLUGIN\SimpleMavenWebApp\src\main\war
[DEBUG] (f) project = MavenProject: org.o7planning:SimpleMavenWebApp:0.0.1-SNAPSHOT @ F:\ECLIPSE_TUTORIAL\TOMCAT_MAVEN_PLUGIN\SimpleMavenWebApp
[DEBUG] (f) settings = org.apache.maven.execution.SettingsAdapter@38002f54
[DEBUG] -- end configuration --
[INFO] Running war on http://localhost:8080/SimpleMavenWebApp
[INFO] Creating Tomcat server configuration at F:\ECLIPSE_TUTORIAL\TOMCAT_MAVEN_PLUGIN\SimpleMavenWebApp
[INFO] create webapp with contextPath: /SimpleMavenWebApp
[DEBUG] context reloadable: false
[DEBUG] adding classPathElement file:/F:/ECLIPSE_TUTORIAL/TOMCAT_MAVEN_PLUGIN/SimpleMavenWebApp/target/classes/
May 21, 2014 10:14:11 PM org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["http-bio-8080"]
May 21, 2014 10:14:11 PM org.apache.catalina.core.StandardService startInternal
```

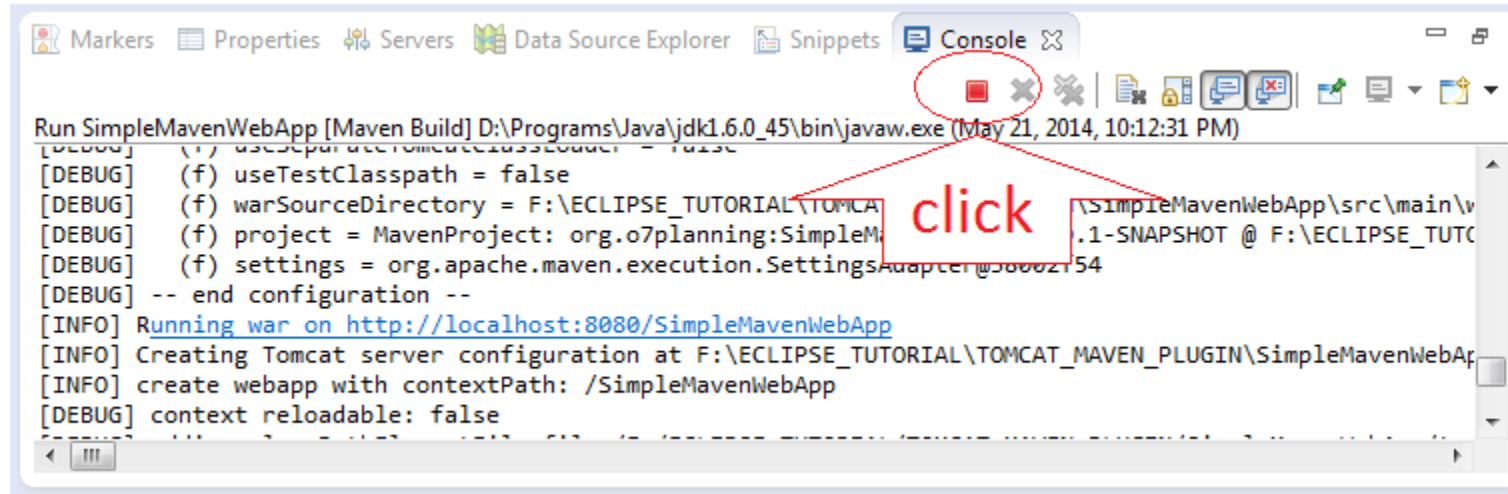
CRÉEZ UNE NOUVELLE CONFIGURATION D'EXÉCUTION

Copiez le lien ci-dessous pour exécuter sur votre navigateur

<http://localhost:8080/SimpleMavenWebApp>



CRÉEZ UNE NOUVELLE CONFIGURATION D'EXÉCUTION



Réexécutez de l'application. Vous pouvez l'exécuter comme montre l'image ci-dessous.

