

# L'utilisation de frameworks pour le développement avec Java EE

## Module 2 - Gradle



# Objectifs

- Savoir installer un environnement Gradle
- Savoir créer un projet Gradle Java en ligne de commande
- Comprendre le fonctionnement des tâches Gradle
- Savoir utiliser Gradle dans un environnement Eclipse

# Moteur de production

- Définition

- Un moteur de production est un logiciel dont la fonction principale consiste à automatiser (ordonnancer et piloter) l'ensemble des actions (préprocessing, compilation, édition des liens, tests...) contribuant, à partir de données sources, à la production d'un ensemble logiciels opérationnels (Wikipédia)

- Exemples

- Ant
- Maven
- Gradle

# Pourquoi utiliser Gradle ?

- Gestionnaire de dépendances
  - Récupération des dépendances dans des repositories Maven, dans des repositories Apache Ivy, dans un système de fichiers, dans un gestionnaire de versions (Subversion)
- Basé sur Groovy
- Gère les cycles complets de création de projet
  - Compilation, tests, génération du fichier jar et déploiement

# Gradle

# Installation

- Téléchargement de l'archive à l'adresse <https://gradle.org/install/>
- Dernière version
  - Gradle 4.5 (released on 24 Jan. 2018)
- Version de Java nécessaire
  - Au minimum la version 7 du JDK ou du JRE
- Dézipper l'archive dans un répertoire
  - Ajouter le répertoire bin dans le PATH

# Gradle Installation

- Vérification : > `gradle -v`

```
:~ EFE$ gradle -v
```

```
-----  
Gradle 4.5  
-----
```

```
Build time: 2017-11-24 17:04:52 UTC  
Revision: 77d0ec90636f43669dc794ca17ef80dd65457bec  
  
Groovy: 2.4.12  
Ant: Apache Ant(TM) version 1.9.9 compiled on February 2 2017  
JVM: 1.8.0_121 (Oracle Corporation 25.121-b13)  
OS: Mac OS X 10.13.2 x86_64
```

# Gradle

# Installation

- Liste des tâches : > `gradle tasks`

```
> Task :tasks
```

```
-----  
All tasks runnable from root project  
-----
```

## Build Setup tasks

```
-----  
init - Initializes a new Gradle build.  
wrapper - Generates Gradle wrapper files.
```

## Help tasks

```
-----  
buildEnvironment - Displays all buildscript dependencies declared in root project 'bin'.  
components - Displays the components produced by root project 'bin'. [incubating]  
dependencies - Displays all dependencies declared in root project 'bin'.  
dependencyInsight - Displays the insight into a specific dependency in root project 'bin'.  
dependentComponents - Displays the dependent components of components in root project 'bin'. [incubating]  
help - Displays a help message.  
model - Displays the configuration model of root project 'bin'. [incubating]  
projects - Displays the sub-projects of root project 'bin'.  
properties - Displays the properties of root project 'bin'.  
tasks - Displays the tasks runnable from root project 'bin'.
```

Gradle

# Installation de Gradle

## Démonstration

Module 02 - 01

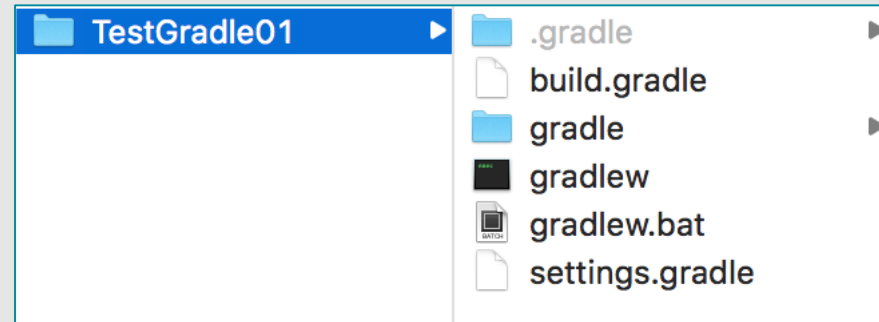




# Création d'un projet Gradle

- Se positionner dans le répertoire souhaité  
> `gradle init`

```
TestGradle01 EFE$ gradle init  
  
BUILD SUCCESSFUL in 0s  
2 actionable tasks: 2 executed  
TestGradle01 EFE$
```



# Fichiers et répertoires générés

- Fichier **build.gradle**
  - permet de configurer les plugins, les dépendances...
- Fichier **settings.gradle**
  - permet de configurer les organisations multiprojets
- Répertoire **.gradle**
  - est un répertoire permettant à Gradle de faire son travail
- Répertoire **gradle**
  - permet configurer gradlew
- Fichiers **gradlew** et **gradlew.bat**
  - commandes (Linux et Windows) utilisées à la place de gradle afin, notamment, de vérifier la version

Gradle

# Création d'un projet Gradle

## Démonstration



# Création d'un projet Java SE

- Ajout d'une référence vers le plugin Java
  - Dans le fichier `build.gradle` :

```
plugins {  
    id 'java-library'  
}
```

- Liste des tâches
  - > `gradle tasks`

> Task :tasks

-----  
All tasks runnable from root project  
-----

**Build tasks**

-----  
`assemble` - Assembles the outputs of this project.  
`build` - Assembles and tests this project.  
`buildDependents` - Assembles and tests this project and all projects that depend on it.  
`buildNeeded` - Assembles and tests this project and all projects it depends on.  
`classes` - Assembles main classes.  
`clean` - Deletes the build directory.  
`jar` - Assembles a jar archive containing the main classes.  
`testClasses` - Assembles test classes.

**Build Setup tasks**

-----  
...

**Documentation tasks**

-----  
`javadoc` - Generates Javadoc API documentation for the main source code.

**Help tasks**

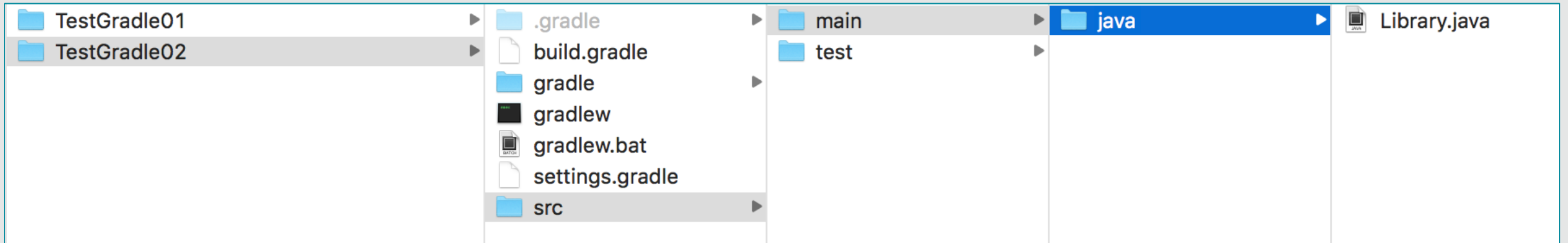
-----  
...

**Verification tasks**

-----  
`check` - Runs all checks.  
`test` - Runs the unit tests.

# Création d'un projet Java SE

- Création d'un nouveau projet Java
  - > `gradle init --type java-library`



# Création d'un projet Java SE

- Compilation des sources
  - > `./gradlew compileJava`

TestGradle01	.gradle	classes	java	main	Library.class
TestGradle02	build	tmp			
	build.gradle				
	gradle				
	gradlew				
	gradlew.bat				
	settings.gradle				
	src				

# Création d'un projet Java SE

- Génération de l'archive JAR
  - > `./gradlew jar`

TestGradle01	.gradle	classes	TestGradle02.jar
TestGradle02	build	libs	
	build.gradle	tmp	
	gradle		
	gradlew		
	gradlew.bat		
	settings.gradle		
	src		

Gradle

# Création d'un projet Java

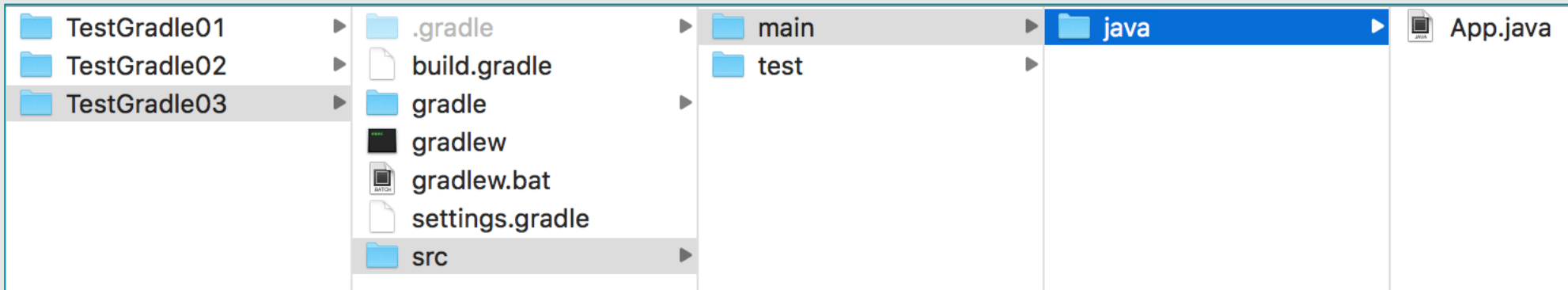
## Démonstration





# Création d'une application Java SE

- Création d'un projet de type application (classe main)
  - > `gradle init --type java-application`



Gradle

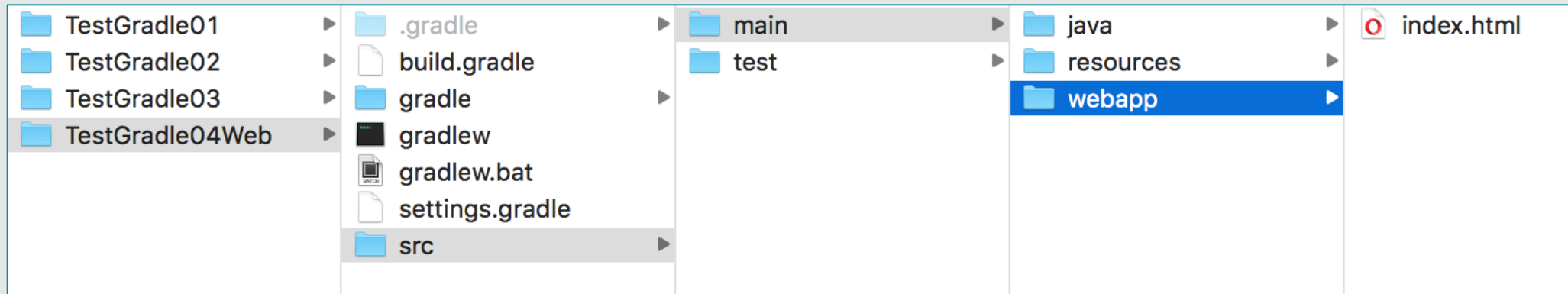
# Création d'une application Java

## Démonstration



# Création d'une application Java EE Web

- Création d'un nouveau projet Java
  - > `gradle init --type java-library`
- Création du répertoire pour le contenu web



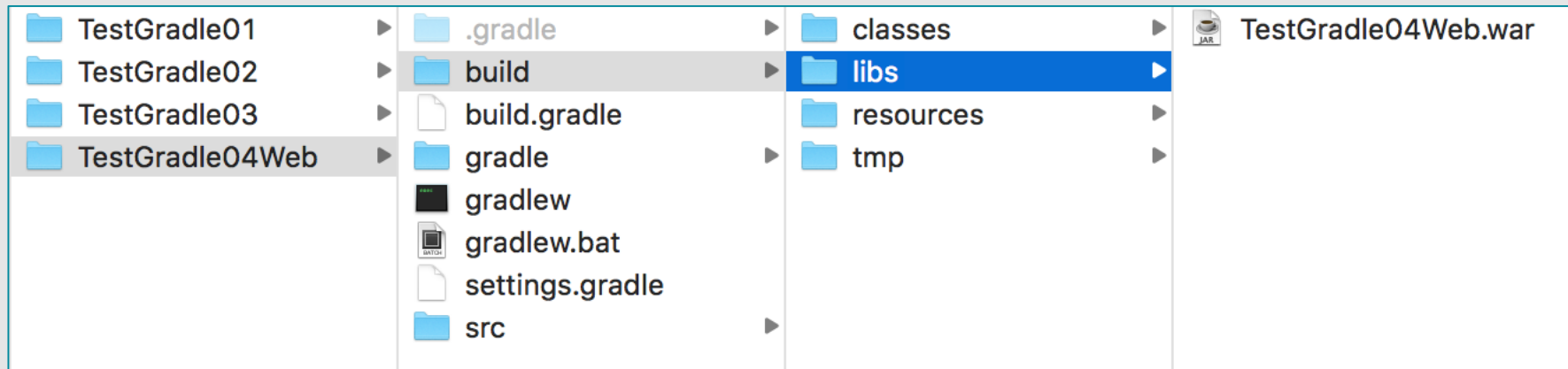
# Création d'une application Java EE Web

- Modification du fichier `build.gradle`

```
plugins {  
    id 'java-library'  
    id 'war'  
}  
  
dependencies {  
  
    // In this section you declare where to find the dependencies of your project  
    repositories {  
        // Use jcenter for resolving your dependencies.  
        // You can declare any Maven/Ivy/file repository here.  
        jcenter()  
    }  
}
```

# Création d'une application Java EE Web

- Génération de l'archive WAR  
    > `./gradlew war`



Gradle

# Création d'un projet Java EE Web

## Démonstration



# Test d'une application Java EE Web

- Possibilité de déployer l'application dans un conteneur web
- Gradle préconise l'utilisation du plugin externe **Gretty**

```
plugins {  
    id 'war'  
    id 'org.akhikh1.gretty' version '1.4.2'  
}
```

- Nouvelles tâches  
    > gradle tasks

```
jettyRestart - Sends 'restart' command to a running server.  
jettyRun - Starts web-app inplace, in interactive mode.  
jettyRunDebug - Starts web-app inplace, in debug and interactive mode.  
jettyRunWar - Starts web-app on WAR-file, in interactive mode.  
jettyRunWarDebug - Starts web-app on WAR-file, in debug and interactive mode.  
jettyStart - Starts web-app inplace (stopped by 'jettyStop').  
jettyStartDebug - Starts web-app inplace, in debug mode (stopped by 'jettyStop').  
jettyStartWar - Starts web-app on WAR-file (stopped by 'jettyStop').  
jettyStartWarDebug - Starts web-app on WAR-file, in debug mode (stopped by 'jettyStop').  
jettyStop - Sends 'stop' command to a running server.  
prepareArchiveWebApp - Prepares war web-app  
prepareInplaceWebApp - Prepares inplace web-app  
prepareInplaceWebAppClasses - Compiles classes of this web-app and all overlays (if any)  
prepareInplaceWebAppFolder - Copies webAppDir of this web-app and all overlays (if any) to ${buildDir}/inplaceWebapp  
showClassPath - Shows classpath information  
tomcatRestart - Sends 'restart' command to a running server.  
tomcatRun - Starts web-app inplace, in interactive mode.  
tomcatRunDebug - Starts web-app inplace, in debug and interactive mode.  
tomcatRunWar - Starts web-app on WAR-file, in interactive mode.  
tomcatRunWarDebug - Starts web-app on WAR-file, in debug and interactive mode.  
tomcatStart - Starts web-app inplace (stopped by 'tomcatStop').  
tomcatStartDebug - Starts web-app inplace, in debug mode (stopped by 'tomcatStop').  
tomcatStartWar - Starts web-app on WAR-file (stopped by 'tomcatStop').  
tomcatStartWarDebug - Starts web-app on WAR-file, in debug mode (stopped by 'tomcatStop').  
tomcatStop - Sends 'stop' command to a running server.
```

# Test d'une application Java EE Web

- `./gradlew tomcatRun`

```
TestGradle05Web EFE$ ./gradlew tomcatRun
17:54:21 INFO Tomcat 8.0.33 started and listening on port 8080
17:54:21 INFO TestGradle05Web runs at:
17:54:21 INFO http://localhost:8080/TestGradle05Web

> Task :tomcatRun
Press any key to stop the server.
<=====--> 87% EXECUTING [11s]
> :tomcatRun
```



Gradle

# Test d'un projet Java EE Web

## Démonstration



# Gestion des dépendances

**@Deprecated  
Pour les java-library**

- Les dépendances peuvent être nécessaires à :
  - La compilation uniquement
    - `compileOnly`
  - La compilation et l'exécution
    - `compile`
  - La compilation et l'exécution des tests
    - `testCompile`

```
dependencies {  
    compile group: 'mysql', name: 'mysql-connector-java', version: '5.1.38'  
    compileOnly group: 'javax.servlet', name: 'javax.servlet-api', version: '3.1.0'  
    testCompile 'junit:junit:4.12'  
}
```

# Gestion des dépendances

- Depuis la version 3.0 de Gradle et pour les java-library:

<b>api</b>	Declaring API dependencies
<b>implementation</b>	Declaring implementation dependencies
<b>compileOnly</b>	Declaring compile only dependencies
runtimeOnly	Declaring runtime dependencies
<b>testImplementation</b>	Test dependencies
testCompileOnly	Declaring test compile only dependencies
testRuntimeOnly	Declaring test runtime dependencies

Gradle

# Gestion des dépendances

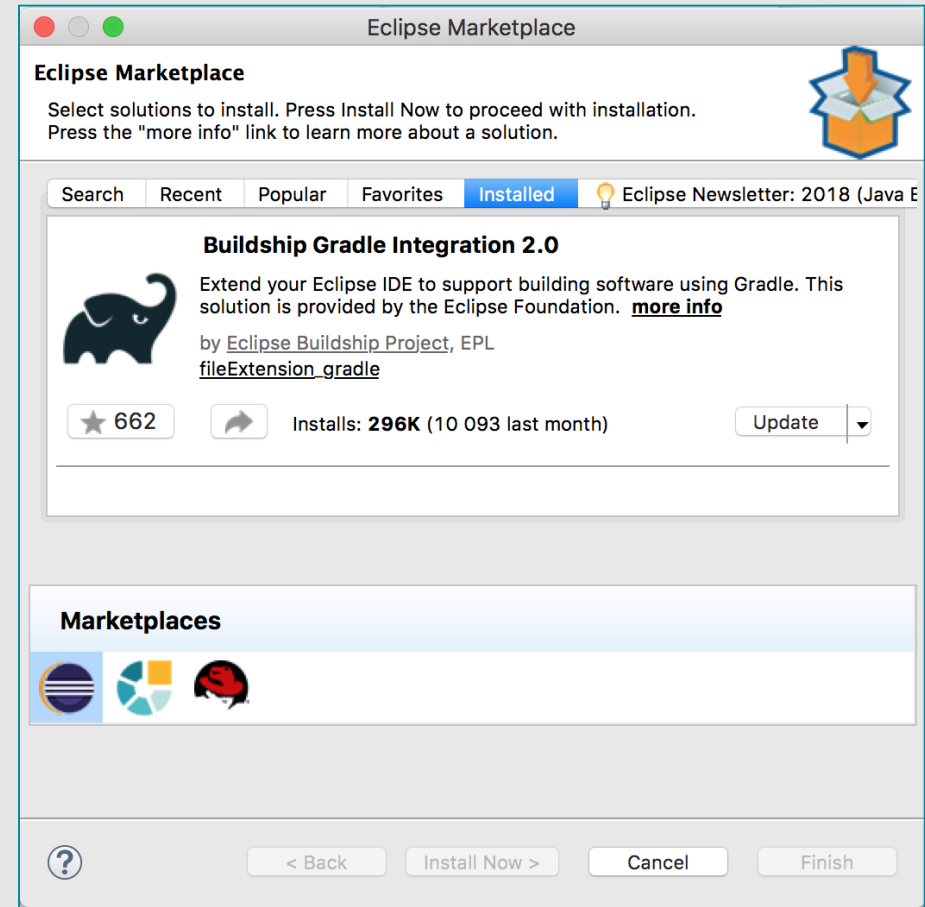
## Démonstration



# Gradle

## Création d'un projet Eclipse

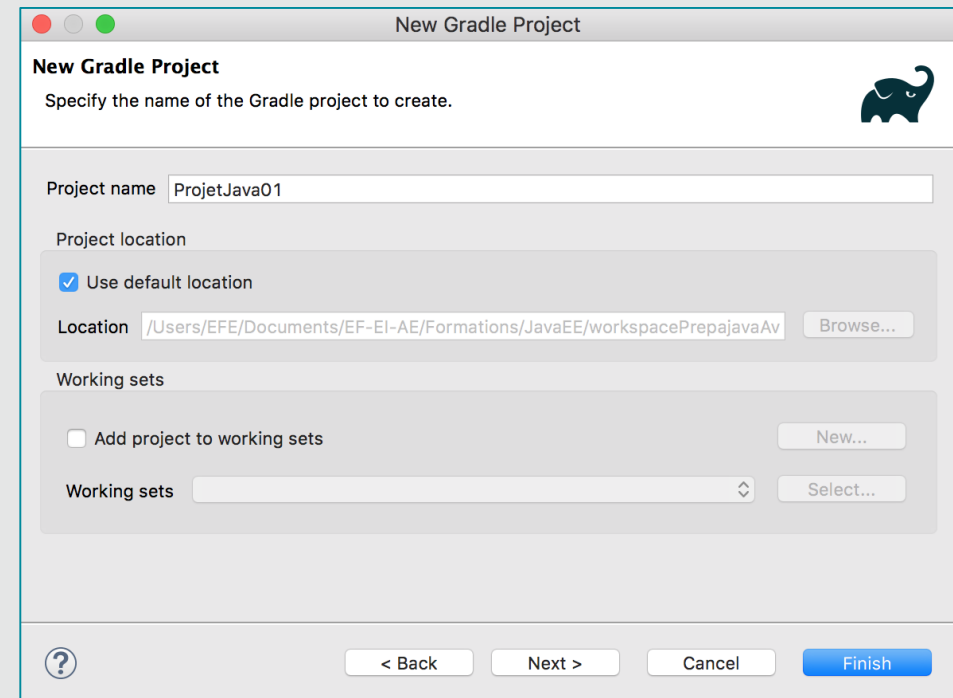
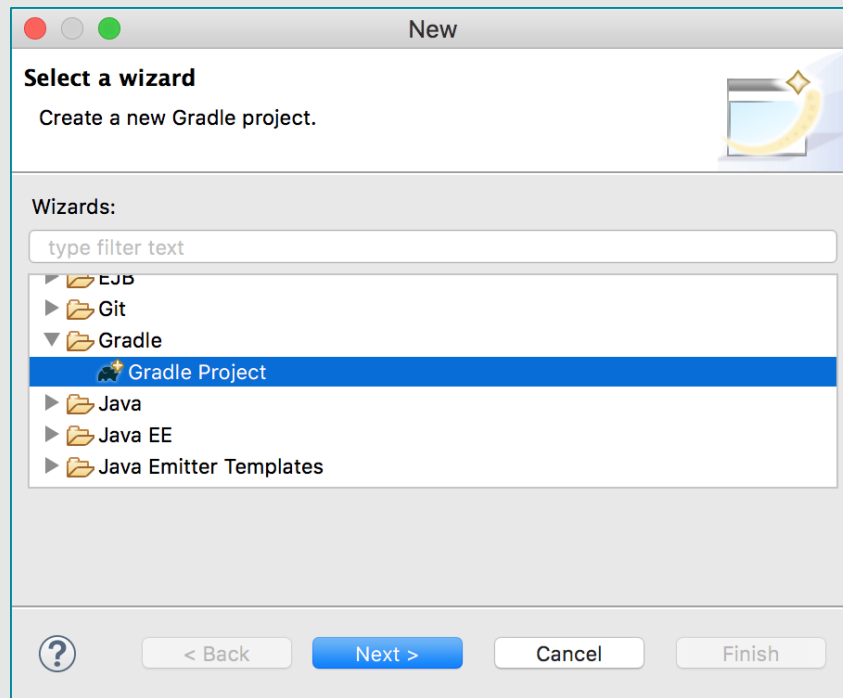
- Installation du plugin Gradle



# Gradle

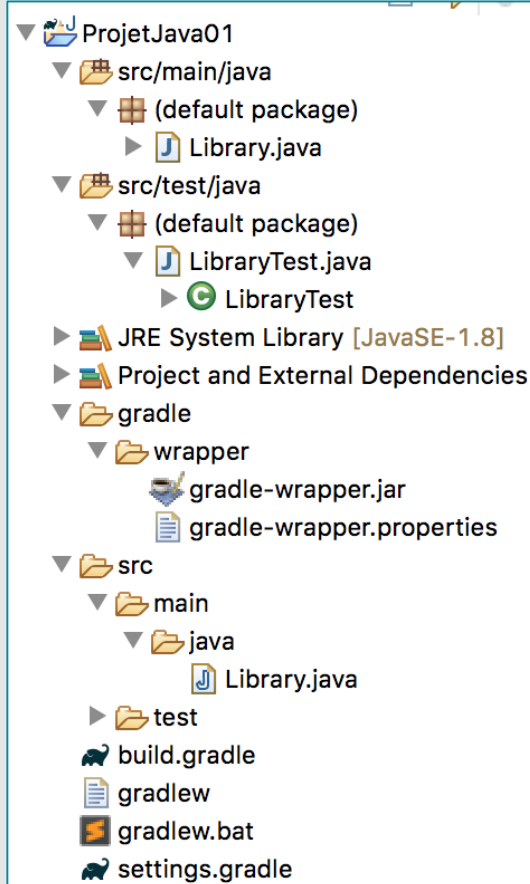
## Création d'un projet Eclipse

- New / Other



# Création d'un projet Eclipse

- Projet



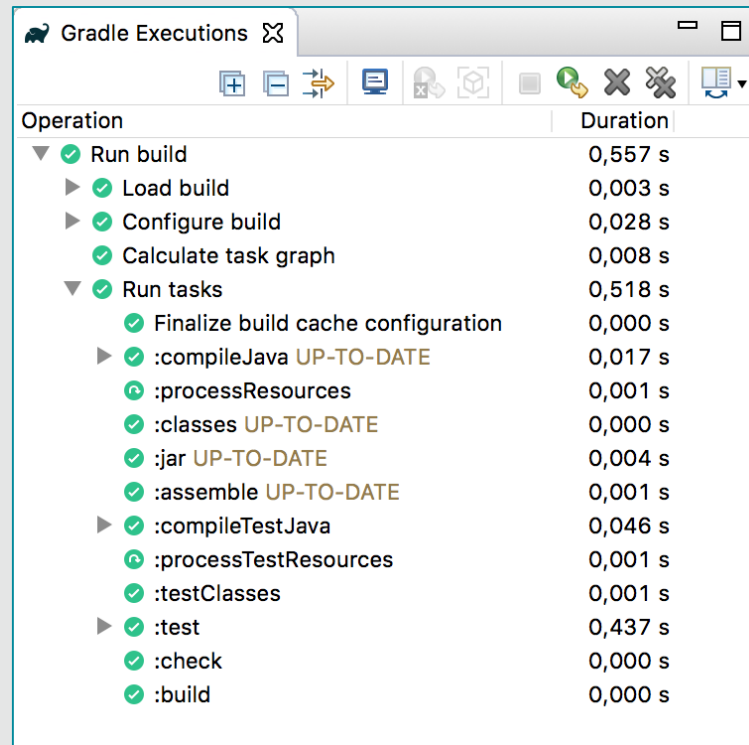
- build.gradle

```
plugins {  
    // Apply the java-library plugin to add support for Java Library  
    id 'java-library'  
}  
  
dependencies {  
    // Use JUnit test framework  
    testCompile 'junit:junit:4.12'  
}  
  
// In this section you declare where to find the dependencies of your project  
repositories {  
    // Use jcenter for resolving your dependencies.  
    // You can declare any Maven/Ivy/file repository here.  
    jcenter()  
}
```

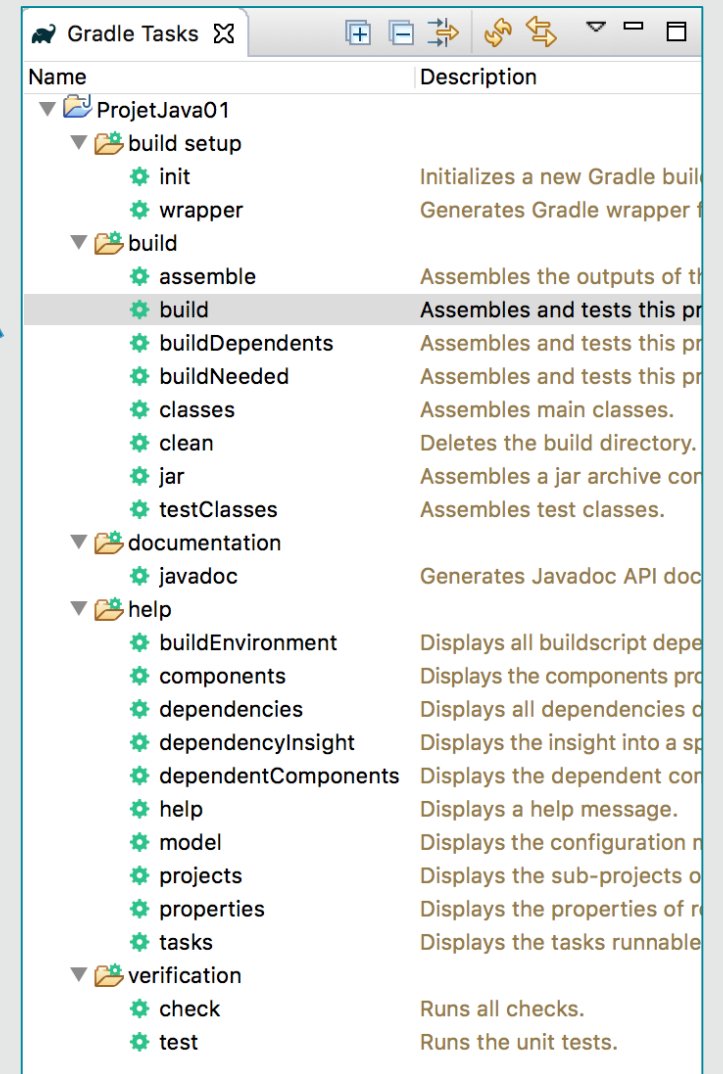
# Gradle

## Création d'un projet Eclipse

- Deux nouvelles vues



Operation	Duration
▼ Run build	0,557 s
▶ Load build	0,003 s
▶ Configure build	0,028 s
Calculate task graph	0,008 s
▼ Run tasks	0,518 s
Finalize build cache configuration	0,000 s
▶ :compileJava UP-TO-DATE	0,017 s
:processResources	0,001 s
:classes UP-TO-DATE	0,000 s
:jar UP-TO-DATE	0,004 s
:assemble UP-TO-DATE	0,001 s
▶ :compileTestJava	0,046 s
:processTestResources	0,001 s
:testClasses	0,001 s
▶ :test	0,437 s
:check	0,000 s
:build	0,000 s



Name	Description
▼ ProjetJava01	
▼ build setup	
init	Initializes a new Gradle build.
wrapper	Generates Gradle wrapper files.
▼ build	
assemble	Assembles the outputs of this project.
build	Assembles and tests this project.
buildDependents	Assembles and tests this project and all projects that depend on it.
buildNeeded	Assembles and tests this project and all projects it depends on.
classes	Assembles main classes.
clean	Deletes the build directory.
jar	Assembles a jar archive containing the classes of this project.
testClasses	Assembles test classes.
▼ documentation	
javadoc	Generates Javadoc API documentation for this project.
▼ help	
buildEnvironment	Displays all buildscript dependencies.
components	Displays the components project.
dependencies	Displays all dependencies of the project.
dependencyInsight	Displays the insight into a specific dependency in this project.
dependentComponents	Displays the dependent components of the project.
help	Displays a help message.
model	Displays the configuration model of the project.
projects	Displays the sub-projects of the project.
properties	Displays the properties of the project.
tasks	Displays the tasks runnable on the project.
▼ verification	
check	Runs all checks.
test	Runs the unit tests.



Gradle

# Création d'un projet Eclipse

## Démonstration



# Gradle

## Complément

- Configuration d'un proxy
  - Ajout d'un fichier `gradle.properties` à la racine du projet

```
# Pour le proxy
# Fichier à déposer à la racine de chaque application
# Ceci est un exemple

systemProp.http.proxyHost=192.168.1.17
systemProp.http.proxyPort=8080
systemProp.https.proxyHost=192.168.1.17
systemProp.https.proxyPort=8080
```

# Gradle

## Complément

- Configuration d'un serveur Gradle d'entreprise
  - Modification du fichier `build.gradle`

```
repositories {  
    jcenter{  
  
        // Utilisation du repository local de l'entreprise  
        // Exemple ici d'un serveur Archiva  
  
        url "http://archiva:8080/repository/JcenterLocal/"  
    }  
}
```