

Gradle Build Tool

Prof. Michele Loreti

Programmazione Avanzata

Corso di Laurea in Informatica (L31)

Scuola di Scienze e Tecnologie

Gradle Build System

<http://www.gradle.org>

Gradle is an advanced general purpose build management system based on **Groovy** and **Kotlin**.

Gradle Build System

<http://www.gradle.org>

Gradle is an advanced general purpose build management system based on **Groovy** and **Kotlin**.

Gradle supports the automatic download and configuration of dependencies or other libraries.

Gradle Build System

<http://www.gradle.org>

Gradle is an advanced general purpose build management system based on **Groovy** and **Kotlin**.

Gradle supports the automatic download and configuration of dependencies or other libraries.

It supports **Maven** and **Ivy** repositories for retrieving these dependencies.

Gradle Build System

<http://www.gradle.org>

Gradle is an advanced general purpose build management system based on **Groovy** and **Kotlin**.

Gradle supports the automatic download and configuration of dependencies or other libraries.

It supports **Maven** and **Ivy** repositories for retrieving these dependencies. This allows reusing the artifacts of existing build systems.

Gradle Build System

<http://www.gradle.org>

Gradle is an advanced general purpose build management system based on **Groovy** and **Kotlin**.

Gradle supports the automatic download and configuration of dependencies or other libraries.

It supports **Maven** and **Ivy** repositories for retrieving these dependencies. This allows reusing the artifacts of existing build systems.

Gradle supports multi-project and multi-artifact builds.

Gradle Build System

<http://www.gradle.org>

Gradle is an advanced general purpose build management system based on **Groovy** and **Kotlin**.

Gradle supports the automatic download and configuration of dependencies or other libraries.

It supports **Maven** and **Ivy** repositories for retrieving these dependencies. This allows reusing the artifacts of existing build systems.

Gradle supports multi-project and multi-artifact builds.

Gradle has the notion of projects and tasks.

Projects and Tasks

A **Gradle** build consists of one or more **projects**.

Projects and Tasks

A **Gradle** build consists of one or more **projects**.

Projects can be something which should be built or something that should be done.

Projects and Tasks

A **Gradle** build consists of one or more **projects**.

Projects can be something which should be built or something that should be done.

Each project consists of **tasks**.

Projects and Tasks

A **Gradle** build consists of one or more **projects**.

Projects can be something which should be built or something that should be done.

Each project consists of **tasks**.

A **task** represents a piece of work which a build performs, e.g., compile the source code or generate the Javadoc.

Projects and Tasks

A **Gradle** build consists of one or more **projects**.

Projects can be something which should be built or something that should be done.

Each project consists of **tasks**.

A **task** represents a piece of work which a build performs, e.g., compile the source code or generate the Javadoc.

Instructions **how to install** Gradle can be found at:

`https://gradle.org/install/`

Initialise a project

Create the directory where your project will go.

Initialise a project

Create the directory where your project will go.

Use the Gradle's `init` command to generate a simple project:

```
$ gradle init
```

Initialise a project

Create the directory where your project will go.

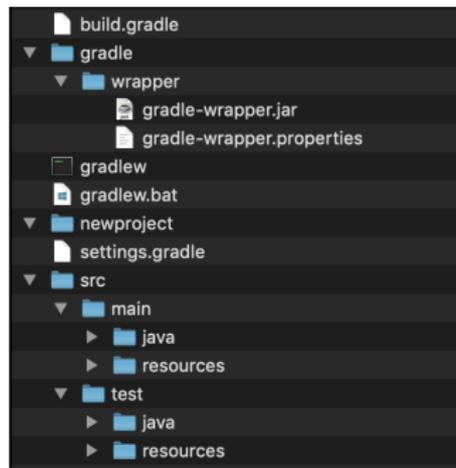
Use the Gradle's `init` command to generate a simple project:

```
$ gradle init
```

When you have interactively **configured** the project, the skeleton of a project is generated.

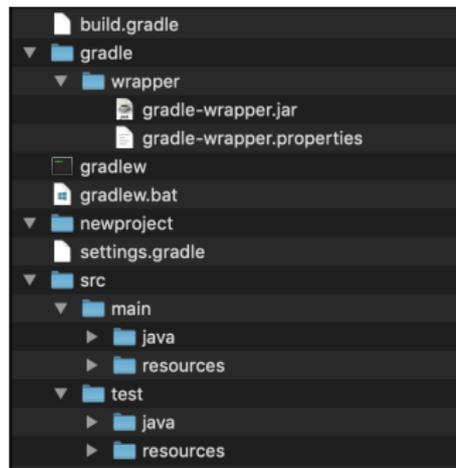
Initialise a project

- `build.gradle`, build script for configuring the current project;



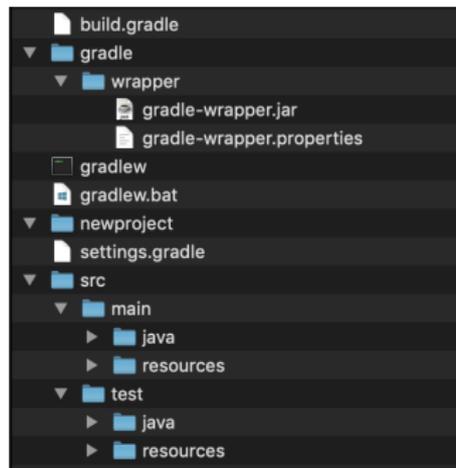
Initialise a project

- `build.gradle`, build script for configuring the current project;
- `gradle-wrapper.jar`, **Gradle Wrapper** executable JAR that invokes a declared version of Gradle, downloading it beforehand if necessary;



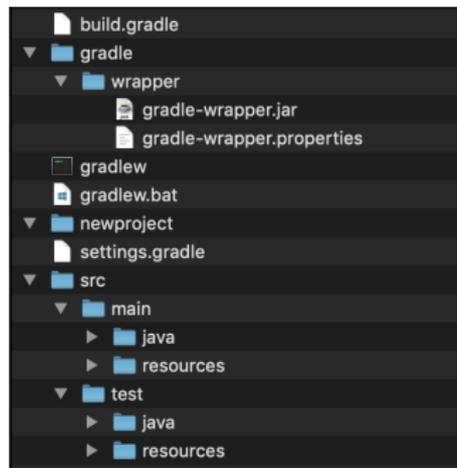
Initialise a project

- `build.gradle`, build script for configuring the current project;
- `gradle-wrapper.jar`, **Gradle Wrapper** executable JAR that invokes a declared version of Gradle, downloading it beforehand if necessary;
- `gradle-wrapper.properties`, Gradle Wrapper configuration properties



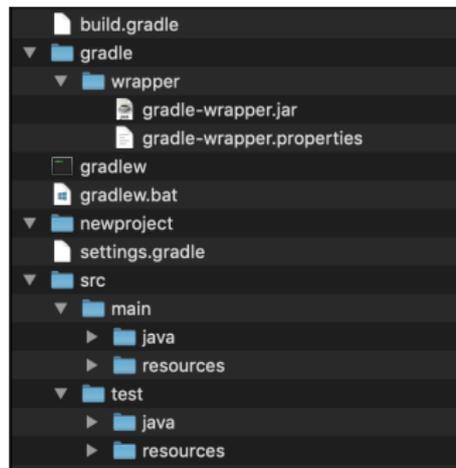
Initialise a project

- `build.gradle`, build script for configuring the current project;
- `gradle-wrapper.jar`, **Gradle Wrapper** executable JAR that invokes a declared version of Gradle, downloading it beforehand if necessary;
- `gradle-wrapper.properties`, Gradle Wrapper configuration properties
- `gradlew`, script for Unix-based systems



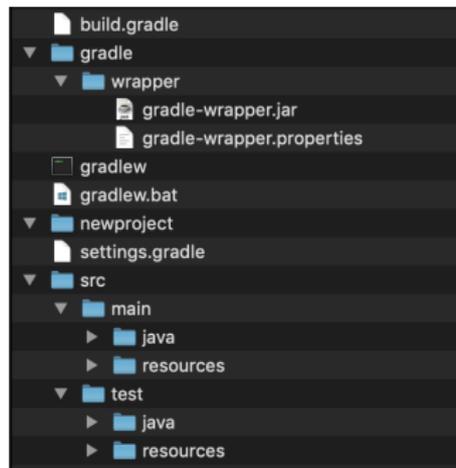
Initialise a project

- `build.gradle`, build script for configuring the current project;
- `gradle-wrapper.jar`, **Gradle Wrapper** executable JAR that invokes a declared version of Gradle, downloading it beforehand if necessary;
- `gradle-wrapper.properties`, Gradle Wrapper configuration properties
- `gradlew`, script for Unix-based systems
- `gradlew.bat`, script for Windows



Initialise a project

- `build.gradle`, build script for configuring the current project;
- `gradle-wrapper.jar`, **Gradle Wrapper** executable JAR that invokes a declared version of Gradle, downloading it beforehand if necessary;
- `gradle-wrapper.properties`, Gradle Wrapper configuration properties
- `gradlew`, script for Unix-based systems
- `gradlew.bat`, script for Windows
- `settings.gradle`, Gradle settings script for configuring the Gradle build



Inspect generated files. . .



Building our application. . .

To build the project, run the build task. You can use either the regular gradle command or use the included wrapper script (the second choice is the best).

Building our application. . .

To build the project, run the build task. You can use either the regular gradle command or use the included wrapper script (the second choice is the best).

```
$ ./gradlew build
```

Building our application. . .

To build the project, run the build task. You can use either the regular `gradle` command or use the included wrapper script (the second choice is the best).

```
$ ./gradlew build
```

The first time you run the build, Gradle will check whether or not you already have the Guava and JUnit libraries in your cache under your `~/.gradle` directory.

Building our application. . .

To build the project, run the build task. You can use either the regular `gradle` command or use the included wrapper script (the second choice is the best).

```
$ ./gradlew build
```

The first time you run the build, Gradle will check whether or not you already have the Guava and JUnit libraries in your cache under your `~/.gradle` directory.

The build task compiles the classes, runs the tests, and generates a test report.

Building our application. . .

To build the project, run the build task. You can use either the regular `gradle` command or use the included wrapper script (the second choice is the best).

```
$ ./gradlew build
```

The first time you run the build, Gradle will check whether or not you already have the Guava and JUnit libraries in your cache under your `~/.gradle` directory.

The build task compiles the classes, runs the tests, and generates a test report.

You can view the test report by opening the HTML output file, located at `build/reports/tests/test/index.html`.

Gradle tasks...

The list of available tasks can be obtained via the option task.

```
$ ./gradlew tasks
```

Gradle tasks. . .

The list of available tasks can be obtained via the option task.

```
$ ./gradlew tasks
```

The list of available tasks can be obtained via the option task.

```
$ ./gradlew tasks
```

Gradle tasks. . .

The list of available tasks can be obtained via the option task.

```
$ ./gradlew tasks
```

The list of available tasks can be obtained via the option task.

```
$ ./gradlew tasks
```

Since we have generated a **Java Application** we can use the task `run` that tell Gradle to execute the main method in the class assigned to the `mainClassName` property.

```
$ ./gradlew run
```

Distributing your Java Application. . .

Gradle can be used to generate zip and tar files containing what you need to run you application.

Distributing your Java Application. . .

Gradle can be used to generate zip and tar files containing what you need to run your application.

This is done by the **Distribution** Gradle plugin.

Distributing your Java Application. . .

Gradle can be used to generate zip and tar files containing what you need to run your application.

This is done by the **Distribution** Gradle plugin.

Some configuration are needed to customise the generation process.

Distributing your Java Application...

```
distTar.dependsOn javadoc
distZip.dependsOn javadoc
```

```
distributions {
    main {
        baseName = 'someName'
        contents {
            from 'src/readme'

            into('src') {
                from 'src/main/java/'
            }

            into('docs') {
                from 'build/docs/javadoc'
            }
        }
    }
}
```

To be continued...