

# Introduciton

Evolution : Ant -> Maven -> Gradle

## Groovy

build.gradle 中用 groovy(DSL) 寫 task 應該是用來擴充 gradle tasks 的 task 項目的

## Create Project

根據[官方文件](#)說明，gradle 預設你的 Java 專案結構長這樣：

```
├─ build.gradle
├─ src
│   ├── main
│   │   ├── java
│   │   └── resources
│   ├── sourceSet
│   │   ├── java
│   │   └── resource
│   └── test
│       ├── java
│       └── resources
```

我先手動建立，Gradle 好像沒有自動化建立專案的功能(? 或是有工具? 應該也可用 maven 去建立?)，建立好以後，我們在 `src/main/java` 下建立 `HelloWorld.java`，簡單的加入 HelloWorld 的程式碼，並且在 `build.gradle` 中加入 `apply plugin: 'java'`，最後的目錄結構會長這樣：

```
~/Desktop/Gradle/first ->tree
.
├─ build.gradle
├─ src
│   ├── main
│   │   ├── java
│   │   │   └── HelloWorld.java
│   │   └── resources
│   ├── sourceSet
│   │   ├── java
│   │   └── resource
│   └── test
│       ├── java
│       └── resources
```

10 directories, 2 files

# Build Project

接著在用 `gradle build` :

```
~/Desktop/Gradle/first -->gradle build
:compileJava
:processResources UP-TO-DATE
:classes
:jar
:assemble
:compileTestJava UP-TO-DATE
:processTestResources UP-TO-DATE
:testClasses UP-TO-DATE
:test UP-TO-DATE
:check UP-TO-DATE
:build
```

BUILD SUCCESSFUL

Total time: 4.572 secs

```
~/Desktop/Gradle/first -->tree
.
├─ build
│   ├── classes
│   │   └─ main
│   │       └─ HelloWorld.class
│   ├── dependency-cache
│   ├── libs
│   │   └─ first.jar
│   └─ tmp
│       ├── compileJava
│       ├── jar
│       └─ MANIFEST.MF
├─ build.gradle
└─ src
    ├── main
    │   ├── java
    │   │   └─ HelloWorld.java
    │   └─ resources
    ├── sourceSet
    │   ├── java
    │   └─ resource
    └─ test
        ├── java
        └─ resources
```

18 directories, 5 files

## Execute Built Project

要直接用 gradle 的 task 執行的話，要在 build.gradle 中加入：

```
task runJar(dependsOn:jar) << {  
    javaexec { main="-jar"; args jar.archivePath }  
}
```

接著就可以輸入 `gradle runjar` 來執行了

## Gradle Dependencies Tool

這天正在安裝一個 node 套件，在使用 `npm install --save` 時發現，Maven 或 Gradle 不能像 npm 那樣，建立好 package.json 後只要寫 `{ "dependencies": {} }`，就可以輸入 `npm install --save *` 來安裝並自動加入 package.json 中，如果我要安裝 lucene，不能只要輸入 `gradle install lucene`；如果我要加入 dependencies，不能只要輸入 `gradle install --save lucene`，你都必须去 Maven Central 查相依套件在 maven 或 gradle 的設定內容後加入 build.gradle，所以我才覺得麻煩，搜尋一下，有人從 npm 過來的也有一樣的懶散習慣，所以就開發一個自動查詢和加入 build.gradle 的工具 gradleps，不過要用 npm install 裝，而且還沒有辦法像 npm install 會自動下載安裝後並加入 build.gradle (也許未來可以幫忙發展)

## Installation & Usage

---

```
$ npm install gradleps -g
```

After installation you can search through Maven Central

```
$ gradleps search guice --limit 5  
Found 296 results. Displaying first 5:  
com.google.inject:guice 4.0-beta  
com.jolira:guice 3.0.0  
org.jvnet.hudson:guice 3.0-rc1  
com.mycila.com.google.inject:guice 3.0-20100927  
org.mod4j.com.google.inject:guice 1.0-XTEXT-PATCHED
```

你要建立一個至少包含 dependencies{} 的 build.gradle，接著你就可以自動把套件安裝設定加入 build.gradle：

```
$ gradle install guice -f build.gradle
```

```
Possible options:
```

- [y] com.google.inject:guice 4.0-beta
- [1] com.jolira:guice 3.0.0
- [2] org.jvnet.hudson:guice 3.0-rc1
- [3] com.mycila.com.google.inject:guice 3.0-20100927
- [4] org.mod4j.com.google.inject:guice 1.0-XTEXT-PATCHED

```
Installing com.google.inject:guice@4.0-beta
```

```
Is it okay? [Y/n/1/2/3/4] choice Y
```

```
All done!
```