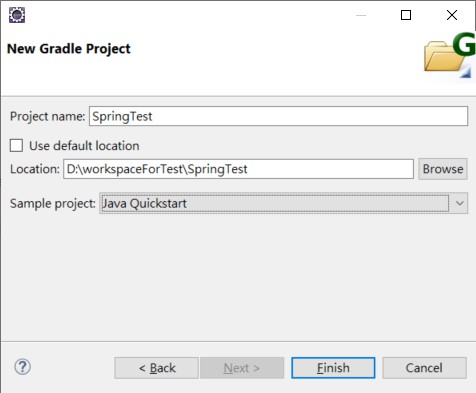
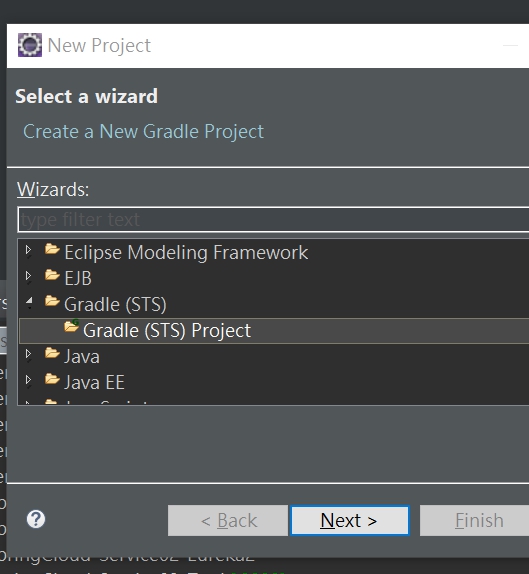
建立Spring專案

**配置**

1.在eclipse建立一個gradle專案



2.把src/main/java裡面的class package都砍掉

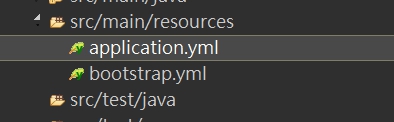
3.build.gradle 改內容

|  |
| --- |
| buildscript {  ext {  springBootVersion = '1.5.18.RELEASE'  }  repositories {  mavenCentral()  }  dependencies {  classpath("org.springframework.boot:spring-boot-gradle-plugin:${springBootVersion}")  }  }  apply plugin: 'java'  apply plugin: 'eclipse'  apply plugin: 'org.springframework.boot'    // JVM 版本號要求  sourceCompatibility = 1.8  targetCompatibility = 1.8  repositories {  mavenCentral()  }  // 取得spring-cloud-starter-config (取得clinet端的依賴)  dependencies {  compile('org.springframework.boot:spring-boot-starter-web')  //test  testCompile('org.springframework.boot:spring-boot-starter-test')    }  dependencyManagement {  imports {  mavenBom "org.springframework.cloud:spring-cloud-dependencies:Dalston.SR5"  }  } |

4.gradle(STS)->Refresh All

下載spring 需要使用的jar包

5.建立bootstrap.yml與 application.yml

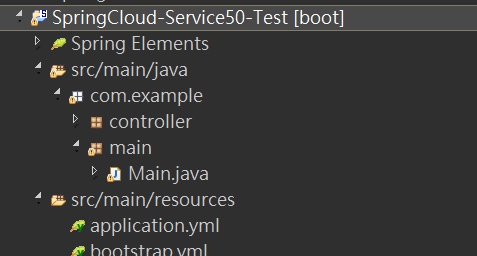


new->other->general->File

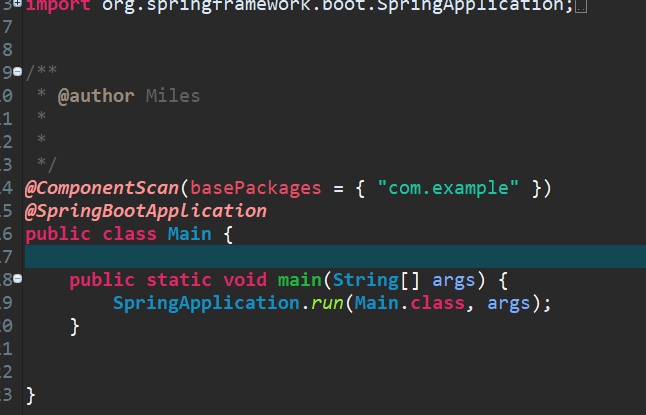
|  |
| --- |
| server:  port: 40990 |

預設值會是8080

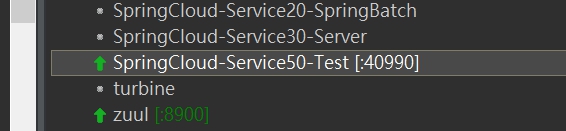
6.建立一個main的主程式，這是啟動spring boot的關鍵



這個Application類便是Spring Boot程序的入口。

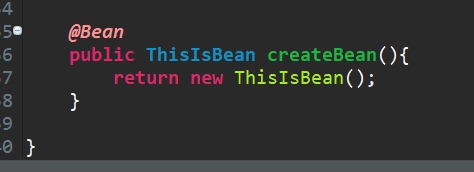


啟動server



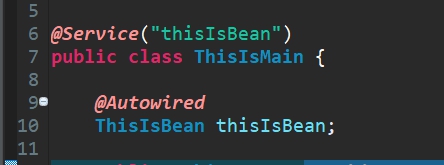
開發

ThisIsBean並沒有綁任何Spring 的Annotation 所以Spring 不知道他是什麼東西，所以要在某個地方建立一個Spring Bean給他。spring 容器裡面就有一個bean 叫做thisIsBean物件



這樣就可以注入了。

因為ThisIsMain有綁Service Annotation 所以在自動掃描的時候，他已經是一個spring bean。



進入程式

