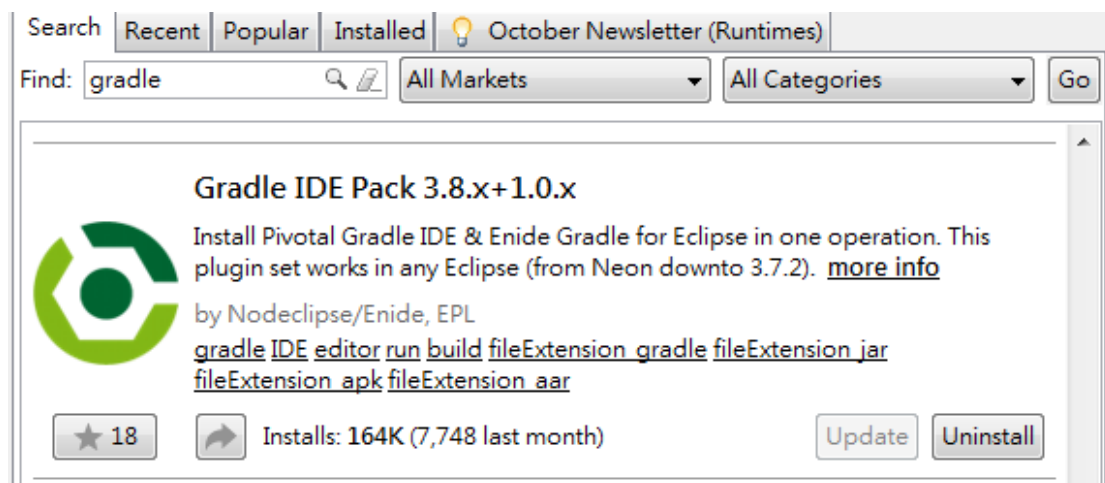
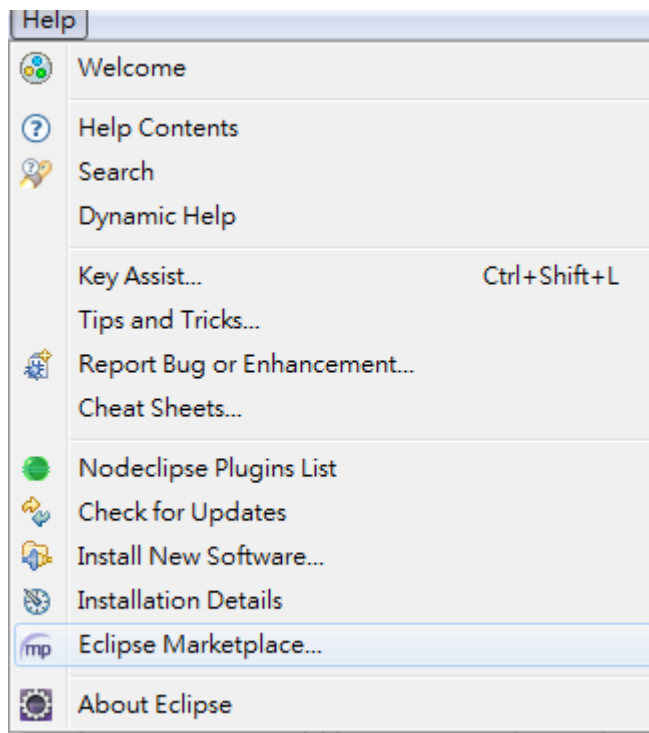


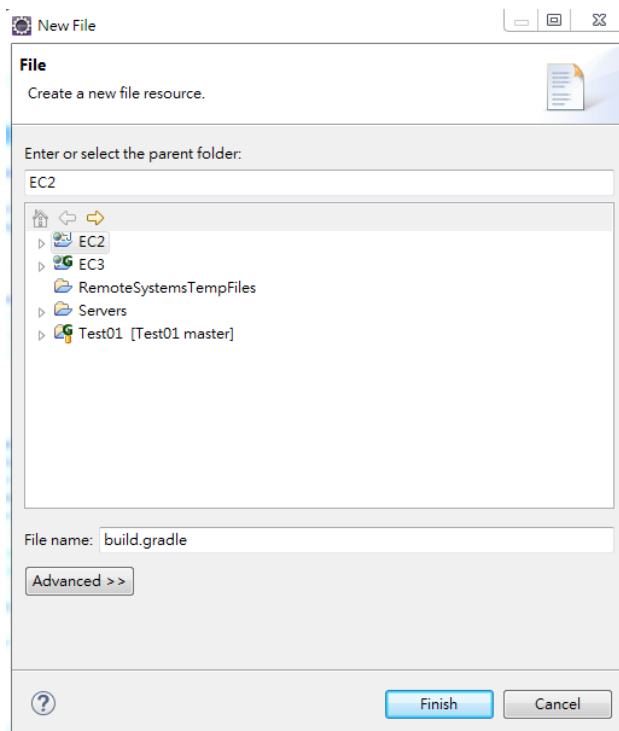
# EC2 Gradle 專案與 Jenkins 結合

## 一. Eclipse 與 Gradle 環境

### 1、安裝 gradle plugin



## 2、新增 build.gradle 檔案



## 3、編輯 build.gradle

```
1 apply plugin: 'java'
2 apply plugin: 'war'
3 apply plugin: 'eclipse-wtp'
4
5 webAppDirName = 'WebContent'
6
7 repositories {
8
9     repositories {
10         maven {
11             url "http://repo1.maven.org/maven2/"
12         }
13         maven {
14             url "https://plugins.gradle.org/m2/"
15         }
16     }
17 }
18
19 dependencies {
20     testCompile 'junit:junit:4.12'
21     compile 'javax.servlet:javax.servlet-api:3.1.0'
22 }
23 }
```

因為在變成 gradle 專案之後，專案會變成原始的 java 專案，必須加入 `webAppDirName = 'WebContent'`，讓專案變成 Dynamic Web Project

➤ Gradle 為甚麼要使用 Maven ?

多數常用的 Java Library 可以在 MVNRepository 網站找到可用的版本。

Gradle 建立在這些基礎資源上：從 Maven Repository 獲取豐富的套件庫，並簡化定義及使用方式。

➤ compile 關鍵字

compile - 原始碼編譯時期 (Compile time) 就需要的函式庫

runtime - 程式執行時期 (Runtime) 才需要用到的函式庫

testCompile - 單元測試 (JUnit) 原始碼編譯時期

testRuntime - 單元測試執行階段

➤ 套件定義格式: Group:Module:Version

例如: org.apache.httpcomponents:httpclient:4.3




➤ plugin 定義

eclipse-wtp: EE 版的 eclipse 和標準版最大的差別在有 Web Tools platform (wtp)，所以使用 eclipse-wtp。

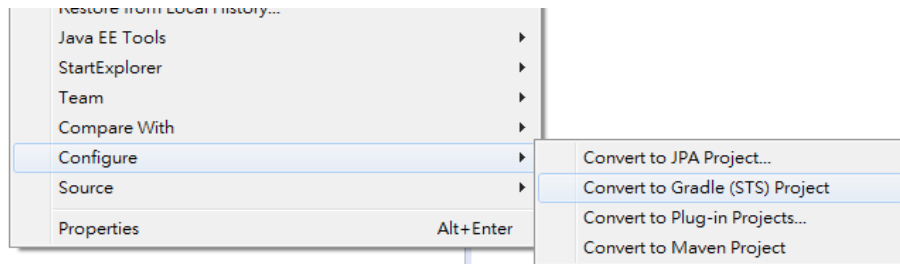
Java : build.gradle 擁有對 Java 專案建置的功能。

Table 64.1. Eclipse plugin behavior

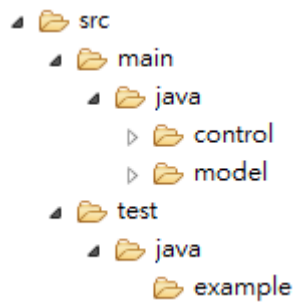
Plugin	Description
None	Generates minimal .project file.
Java	Adds Java configuration to .project. Generates .classpath and JDT settings file.
Groovy	Adds Groovy configuration to .project file.
Scala	Adds Scala support to .project and .classpath files.
War	Adds web application support to .project file.
Ear	Adds ear application support to .project file.

 .classpath	2016/11/18 下午 ...	CLASSPATH 檔案	1 KB
	2016/11/9 下午 0...	文字文件	1 KB
 .project	2016/11/18 下午 ...	PROJECT 檔案	1 KB

#### 4、將專案變成 gradle 專案



#### 5、改變資料夾路徑

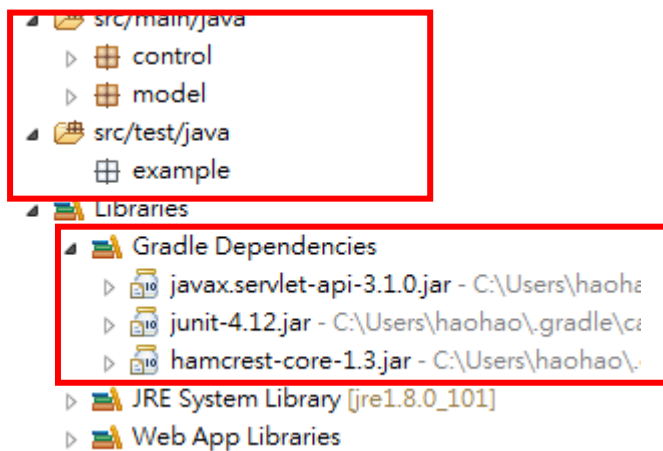
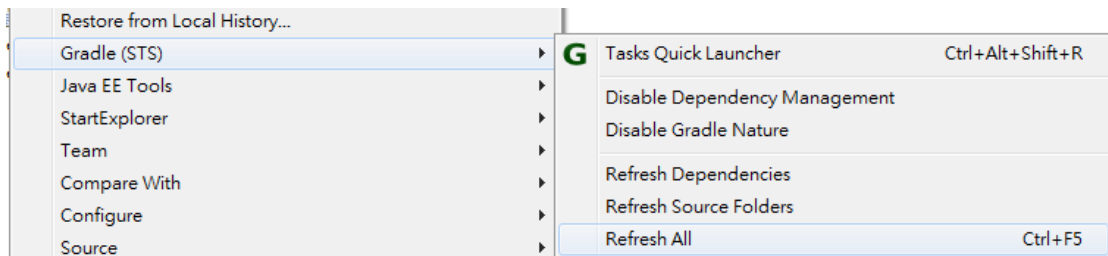


必須要有指定的資料夾路徑，gradle 才能讀到，

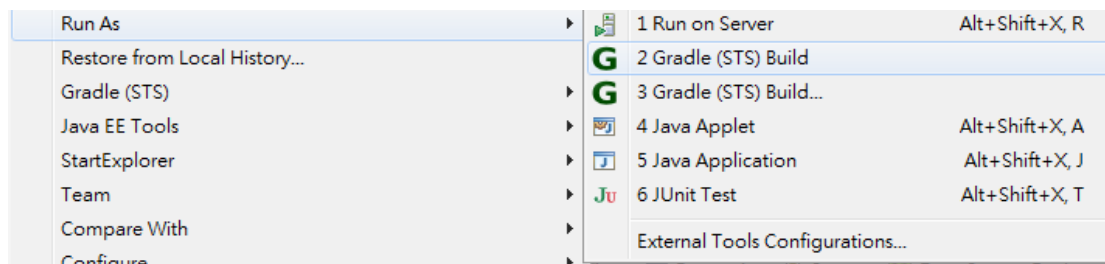
檔案必須放在 src / main / java 底下

測試檔案則是放在 src / test / java 底下

#### 6、Refresh gradle



## 7、Run as gradle build 專案

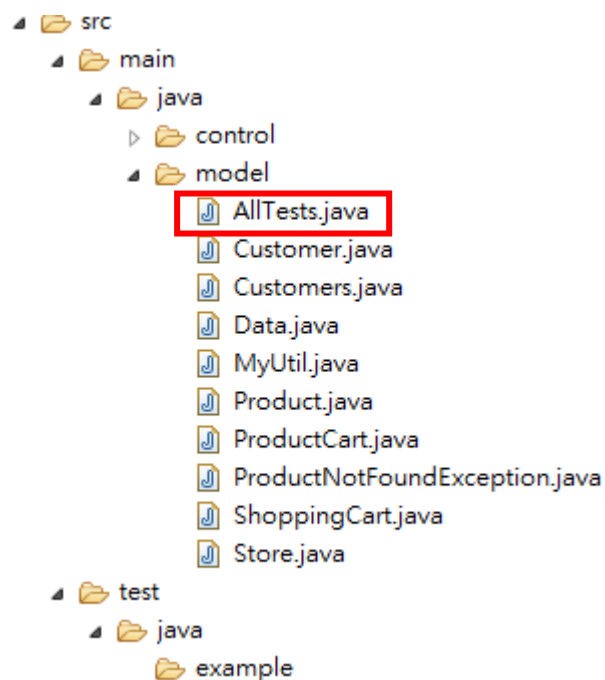


## 8、會出現 build 錯誤訊息

```
<terminated> EC2 [Gradle (STS) Build] Gradle Build on EC2
[sts] -----
:compileJavaC:\tmp\workspace\EC2\src\main\java\model\AllTests.java:3: error: package org.junit.runner does not exist
import org.junit.runner.RunWith;
^
C:\tmp\workspace\EC2\src\main\java\model\AllTests.java:4: error: package org.junit.runners does not exist
import org.junit.runners.Suite;
^
C:\tmp\workspace\EC2\src\main\java\model\AllTests.java:5: error: package org.junit.runners.Suite does not exist
import org.junit.runners.Suite.SuiteClasses;
^
C:\tmp\workspace\EC2\src\main\java\model\AllTests.java:7: error: cannot find symbol
@RunWith(Suite.class)
^
symbol: class RunWith
```

因為在 src / main / java / model 裡的 AllTests.java 是測試檔，不是一般檔案

## 9、將 AllTests.java 丟到 test / java / example 裡



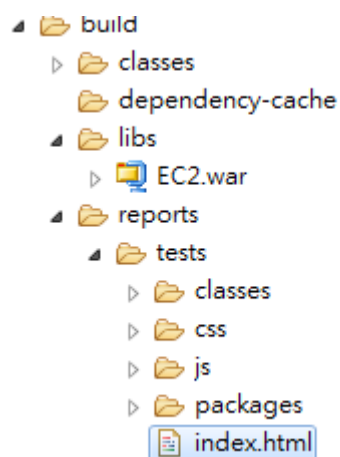
## 10、更改 Alltests.java 套件，並新增一個 ProductTest 檔

```
1 package example;
2
3+ import org.junit.runner.RunWith;
6
7 @RunWith(Suite.class)
8 @SuiteClasses({})
9 public class AllTests {
10
11 }
12
```

---

```
1 package example;
2
3- import static org.junit.Assert.*;
4 import model.Product;
5
6 import org.junit.After;
7 import org.junit.Before;
8 import org.junit.Test;
9 public class ProductTest {
10 private Product p;
11 private int expectedPrice;
12- @Before
13 public void setUp() throws Exception {
14 p = new Product("Book", "UML", 200);
15 expectedPrice = 200;
16 }
17- @After
18 public void tearDown() throws Exception {
19 p = null;
20 }
21- @Test
22 public void testGetPrice() {
23 assertEquals(expectedPrice, p.getPrice());
24 }
25 }
```

## 11、再 run 一次 build gradle

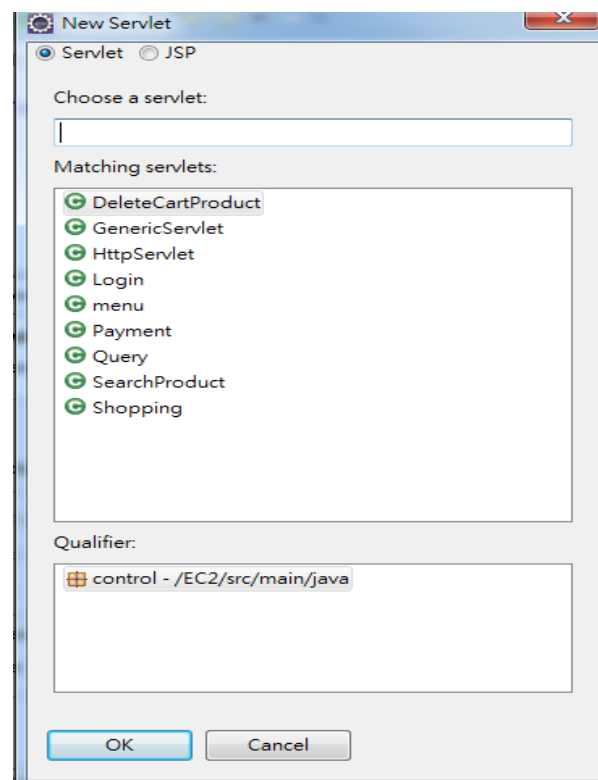
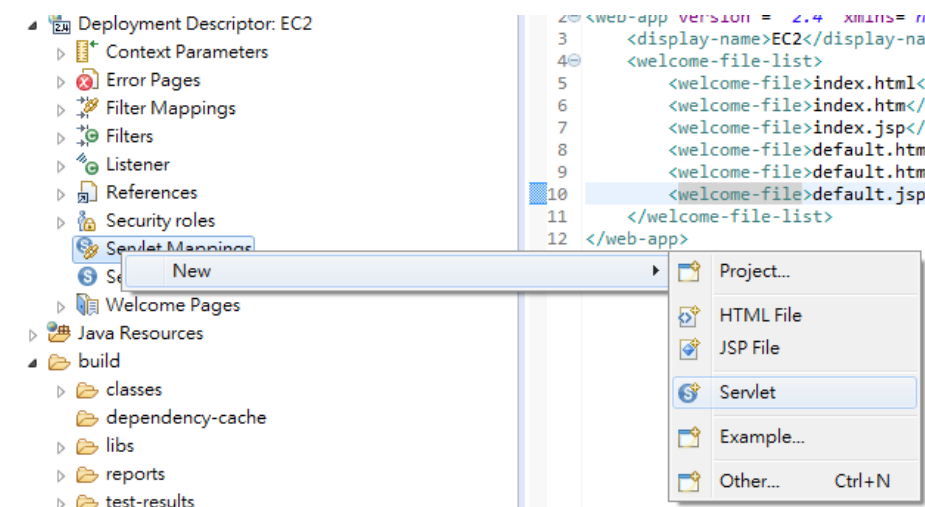


build / reports / tests / index.html ，有測試的結果網頁

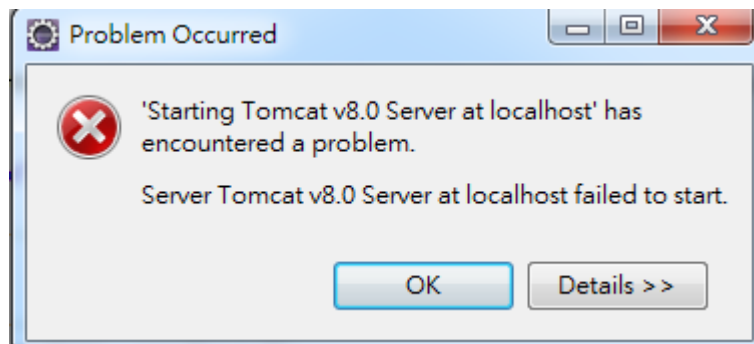
## 12、在 src / main / java / control / Login.java Run on server



因為 gradle 改變了檔案路徑，必須重新對 servlet 做 mapping



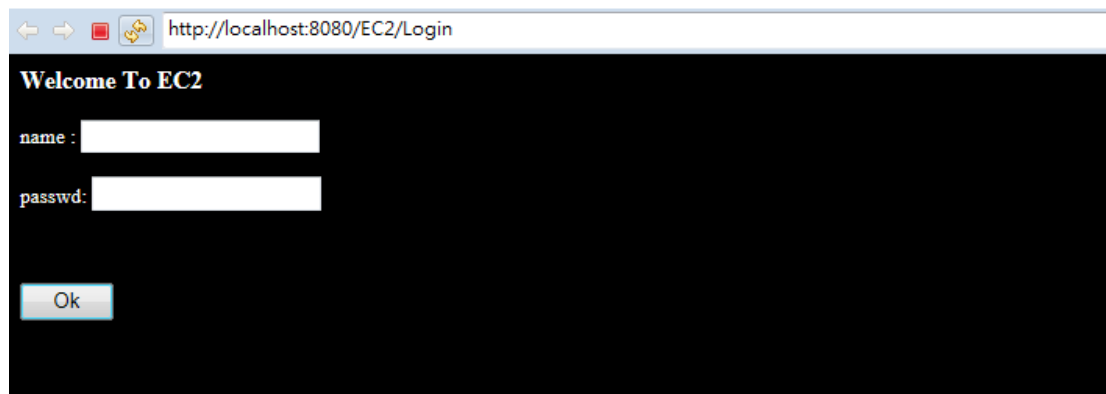
再執行一次，出現 server 無法執行的錯誤



因為必須在 Login.java 找到相對應 mapping 的 urlpattern

```
13 /** Servlet implementation class Login */
14 @WebServlet(name="Login", urlPatterns={"/Login"}, loadOnStartup=1)
15 public class Login extends HttpServlet {
16     private static final long serialVersionUID = 1L
```

然後再執行一次





## 二. 專案連接 git server 與 git server 設定

1、在虛擬機 /usr / share / gitrepo / ec2 創立一個 git server

```
jykuo@ubuntu:/usr/share/gitrepo/ec2$ git init --bare  
Initialized empty Git repository in /usr/share/gitrepo/ec2/  
jykuo@ubuntu:/usr/share/gitrepo/ec2$
```

2、將專案 push 到 git server

Push to Another Repository

**Destination Git Repository**  
Enter the location of the destination repository.

**Location**

URI:  Local File...

Host:

Repository path:

**Connection**

Protocol:

Port:

**Authentication**

User:

Password:

Store in Secure Store ☐

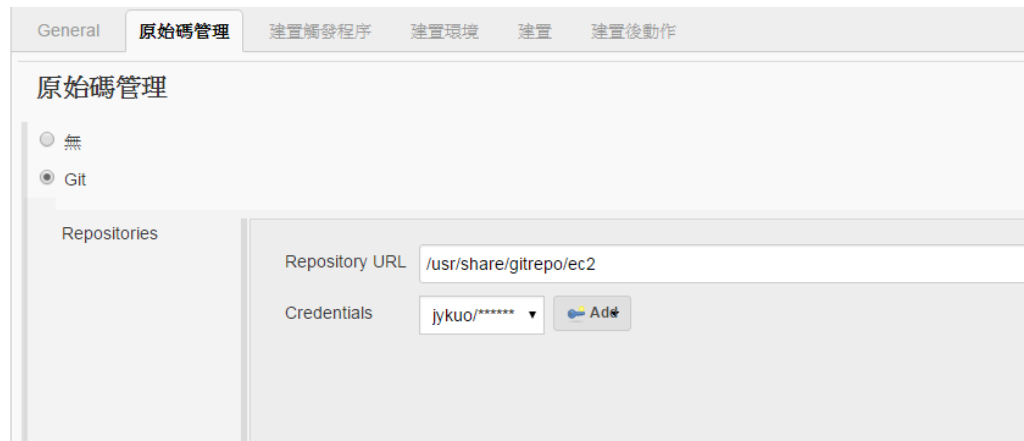
192.168.200.128 為虛擬機的 ip，以 ssh 方式連接

### 三. Jenkins 設定

#### 1、 在 Jenkins 創立一個專案

在原始碼管理增加 git repositories: `/usr/share/gitrepo/ec2`

當 git server 被 push 一份新版本，jenkins 就會自動建置



#### 2、 設定 gradle build



#### 3、 執行完成果

### Workspace of EC2 on master



測試網頁結果

Test Summary

1

tests

0

failures

0

ignored

0.002s

duration

100%

successful

PackagesClasses

Packages

Package	Tests	Failures	Ignored	Duration	Success rate
<a href="#">example</a>	1	0	0	0.002s	100%

Classes

Class	Tests	Failures	Ignored	Duration	Success rate
<a href="#">example.ProductTest</a>	1	0	0	0.002s	100%

1

Generated by Gradle 3.0 at Nov 9, 2016 12:59:56 AM

Jenkins 每五分鐘建置一次的結果

Git 輪詢記錄

Started on Nov 9, 2016 1:01:00 AM  
Using strategy: Default  
[poll] Last Built Revision: Revision 267f19fab7f528677f17c5dee5d2a104e54b0ec0  
using GIT\_ASKPASS to set credentials  
> git ls-remote -h /usr/share/gitrepo/ec2 # timeout=10  
Found 1 remote heads on /usr/share/gitrepo/ec2  
[poll] Latest remote head revision on refs/heads/master is: 267f19fab7f528677f  
Done. Took 52 ms  
No changes

## 四. Selenium + Junit

### 1、 Selenium 錄製測試腳本，產生 Junit

Firefox 版本 : 33.0, selenium 版本 : 2.53.0

新增一個 ShoppingTest.java 檔 放到 src / test / java / example 下，  
將錄製完產生的 java code 複製貼上

```
package example;

import java.util.regex.Pattern;

public class ShoppingTest {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://localhost:8080/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testShopping() throws Exception {
        driver.get(baseUrl + "EC2/Login");
        driver.findElement(By.name("name")).clear();
        ...
    }
}
```

### 2、 新增 Selenium 的 套件

```
dependencies {
    testCompile 'junit:junit:4.12'
    compile 'javax.servlet:javax.servlet-api:3.1.0'
    compile "org.seleniumhq.selenium:selenium-java:2.53.0"
}
```

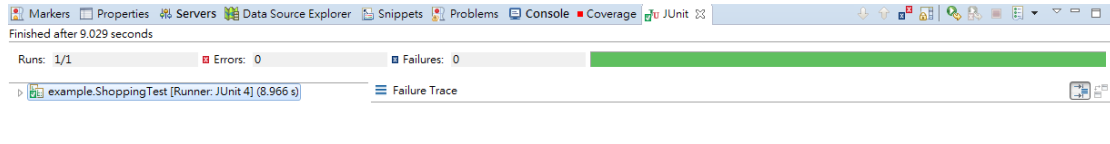
Refresh gradle

- Libraries
  - Gradle Dependencies
    - cglib-nodep-2.1.3.jar - C:\Users\haohao\gra
    - gson-2.3.1.jar - C:\Users\haohao\gradle\cach
    - guava-19.0.jar - C:\Users\haohao\gradle\cac
    - commons-codec-1.9.jar - C:\Users\haohao\g
    - commons-io-2.4.jar - C:\Users\haohao\gradle
    - commons-logging-1.2.jar - C:\Users\haohao\
    - netty-3.5.7.Final.jar - C:\Users\haohao\gradle
    - javax.servlet-api-3.1.0.jar - C:\Users\haohao\
    - junit-4.12.jar - C:\Users\haohao\gradle\cache
    - jna-platform-4.1.0.jar - C:\Users\haohao\grac
    - jna-4.1.0.jar - C:\Users\haohao\gradle\cache:
    - commons-exec-1.3.jar - C:\Users\haohao\gra
    - httpclient-4.5.1.jar - C:\Users\haohao\gradle\
    - httpcore-4.4.3.jar - C:\Users\haohao\gradle\c
    - hamcrest-core-1.3.jar - C:\Users\haohao\grac
    - selenium-api-2.53.0.jar - C:\Users\haohao\gr
    - selenium-chrome-driver-2.53.0.jar - C:\Users\h
    - selenium-edge-driver-2.53.0.jar - C:\Users\ha

### 3、對 ShoppingTest.java 作 Junit test



會自動開啟一個 firefox 瀏覽器，執行剛剛錄製的畫面，測試成功如下圖



### 4、將專案 push 到 Git server，並在 Jenkins 建置

出現 Failed to connect to binary FirefoxBinary(/usr/bin/firefox) on port 7055

測試錯誤

	<a href="#">#6</a>	2016/11/17 上午 2:53
	<a href="#">#5</a>	2016/11/17 上午 1:23
	<a href="#">#4</a>	2016/11/17 上午 1:11
	<a href="#">#3</a>	2016/11/17 上午 1:08
	<a href="#">#2</a>	2016/11/9 上午 1:01
	<a href="#">#1</a>	2016/11/9 上午 12:59

#### Class example.ShoppingTest

all > example > ShoppingTest

1	1	0	52.901s	0%
tests	failures	ignored	duration	successful

Failed tests Tests Standard error

#### Failed tests

##### testShopping

```
org.openqa.selenium.WebDriverException: Failed to connect to binary FirefoxBinary(/usr/bin/firefox) on port 7055; process output follows:
Error: no display specified

Build info: version: '2.53.0', revision: '35ae25b1534ae328c771e0856c93e187490ca824', time: '2016-03-15 10:43:46'
System info: host: 'ubuntu', ip: '127.0.1.1', os.name: 'Linux', os.arch: 'amd64', os.version: '4.2.0-27-generic', java.version: '1.8.0_101'
Driver info: driver.version: FirefoxDriver
    at org.openqa.selenium.firefox.internal.NewProfileExtensionConnection.start(NewProfileExtensionConnection.java:125)
    at org.openqa.selenium.firefox.FirefoxDriver.startClient(FirefoxDriver.java:271)
    at org.openqa.selenium.remote.RemoteWebDriver.<init>(RemoteWebDriver.java:119)
    at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:216)
    at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:211)
    at org.openqa.selenium.firefox.FirefoxDriver.<init>(FirefoxDriver.java:207)
```

因為我們必須在 Jenkins 也建置 Selenium 環境

## 5、建置 Selenium 環境

管理 Jenkins → 外掛程式管理 → 安裝 Xvfb plugin → 重啟 Jenkins

☒ [Xvfb plugin](#)

This plugin starts before build, and stops after the build the [Xvfb](#) virtual framebuffer X11 server.

為甚麼要裝 Xvfb plugin ?

Xvfb 可以在 server 啟動虛擬 X window，讓 Selenium 可以在瀏覽器上跑測試。

管理 Jenkins → 組態設定 → 全域屬性

全域屬性

☐ 工具位置

☒ 環境變數

Key-Value 對應清單

名稱	DISPLAY
值	:99

管理 Jenkins → Global Tool Configuration → Xvfb installation

Xvfb installation

Xvfb installation 安裝

Xvfb installation  
Name

default

Directory in which to find Xvfb executable

☐ 自動安裝

新增 Xvfb installation

本系統上的 Xvfb installation 安裝清單

EC2 專案 → 組態 → 建置環境

建置環境

☒ Start Xvfb before the build, and shut it down after.

Xvfb installation	default
Let Xvfb choose display name	<input type="checkbox"/>
Xvfb specific display name	99
Start only on nodes labeled	
I'm running this job in parallel on same node	<input checked="" type="checkbox"/>
Timeout in seconds	0
Xvfb screen	1024x768x8
Xvfb display name offset	1

## 6、再建置一次專案

測試部分跑了兩分多鐘，出現一直找不到 Selenium element 的錯誤

### Class example.ShoppingTest

all > example > ShoppingTest

1 tests	1 failures	0 ignored	2m42.98s duration	0% successful
------------	---------------	--------------	----------------------	------------------

Failed tests Tests

#### Failed tests

##### testShopping



```
org.openqa.selenium.NoSuchElementException: Unable to locate element: {"method":"name","selector":"name"}
Command duration or timeout: 94.07 seconds
Build info: version: '2.53.0', revision: '35ae25b1534ae328c771e0856c93e187490ca824', time: '2016-03-15 10:43:46'
System info: host: 'ubuntu', ip: '127.0.1.1', os.name: 'Linux', os.arch: 'amd64', os.version: '4.2.0-27-generic'
Driver info: org.openqa.selenium.firefox.FirefoxDriver
Capabilities [{applicationCacheEnabled=true, rotatable=false, handlesAlerts=true, databaseEnabled=true, version=42.0.1, platform=Linux}]
Session ID: 7a4c2d73-312a-416e-8160-705be6aca209
```

必須把 ShoppingTest.java 的 localhost 改成實體 ip


因為 ubuntu server 的 localhost 並不是 EC2 的專案，所以 Selenium 一直抓不到，無法測試。

`baseUrl = "http://localhost:8080/";` ➡ `baseUrl = "http://140.124.181.9:8080/";`


改完後 push 到 Git server，再建置一次

 建置歷程  測試


X

 **#17**

 2016/11/18 上午 2:10

 **#16**

 2016/11/18 上午 12:40

 **#15**

 2016/11/17 下午 10:38

2 tests	0 failures	0 ignored	51.609s duration	100% successful
------------	---------------	--------------	---------------------	--------------------

Packages Classes

Packages

Package	Tests	Failures	Ignored	Duration	Success rate
<a href="#">example</a>	2	0	0	51.609s	100%

Classes

Class	Tests	Failures	Ignored	Duration	Success rate
<a href="#">example.ProductTest</a>	1	0	0	0.003s	100%
<a href="#">example.ShoppingTest</a>	1	0	0	51.606s	100%