

DevOps 範例 Jenkins 和 Selenium

投資程設科 吳秉豫

2018.2.27

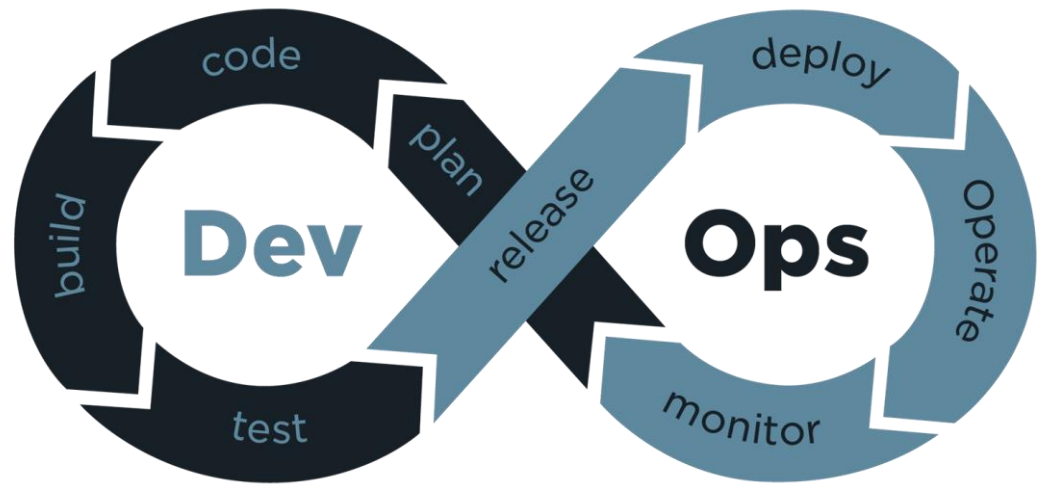


國泰人壽

國泰金控

DevOps

- **DevOps**是能讓營運單位和資訊開發維運單位之間更密切合作的一種方式。
- 實務上要能透過自動化來達成**持續整合**、**持續交付**及**持續監控**應用程式。



JENKINS



國泰人壽

國泰金控

持續整合和持續交付工具-Jenkins

- **Jenkins**是一套開源的自動化伺服器
- 主要用做持續整合工具，2.0版增加持續交付支援
- 支援Java、Node.js、python、iOS等應用程式
- 支援大多數版本控制系統
- 支援測試報表產出和信件通知
- 有豐富的plugins做延伸使用
- 安裝簡單且設定方便
- 目前最新穩定版：2.89.3 (2018/02)



本地端測試Jenkins事前準備

安裝以下軟體

	使用Java 8	使用Java 7
Java	JDK 8	JDK 7
建置工具	Maven (3.5.2) / Ant (1.10.1)	Maven (3.5.2) / Ant (1.9.9)
版控工具	Git (2.16.1)	
持續整合工具	Jenkins (2.89.3)	Jenkins (2.53)

後續範例為使用Java 8



本地端測試Jenkins事前準備

建立測試用Java專案

- 開啟 Git ，建立本地端 Repository
- 建立 Java Maven 專案，Workspace指向本地端 Git Repository
- 建立Java模組和JUnit測試程式



JAVA軟體專案項目管理和建構工具

MAVEN



Java 建置工具發展

Java 開發過程：開發 > 編譯 > 測試 > 包裝等工作

1. 原生：javac、java、jar等指令

2. Ant：

透過XML檔([build.xml](#))定義要執行的工作(target)和流程(depends)。

3. Maven：


透過專案對象模型([pom.xml](#))管理Java專案，內建專案生命週期(Project Lifecycle)進行工作和相依管理(Dependency Management)管理第三方套件(存放於[Repository](#))

4. Gradle：

結合Ant和Maven優點，並改用更簡潔的Groovy來開發。



Jenkins首頁儀表板


 **Jenkins**


搜尋


admin | 登出


Jenkins ▶


開啟自動更新頁面


 新增作業


 使用者


 建置歷程


 專案關連


 檢查檔案指紋

 管理 Jenkins

 我的視景

 Credentials

 New View

 新增描述

歡迎使用Jenkins!

請[建立新工作](#)來開始使用

建置佇列


佇列中沒有建置作業。

建置執行程式狀態

- 閒置
- 閒置

頁面產生時間: 2018/2/6 上午10時56分46秒 [REST API](#) [Jenkins ver. 2.89.3](#)

建立Jenkins專案

 **Jenkins**

搜尋


admin | 登出

Jenkins > 全部 >

Enter an item name


First Jenkins

» Required field




建置 Free-Style 軟體專案

這是 Jenkins 的主要功能。Jenkins 能搭配各式程式碼管理、建置系統來建置您的專案，甚至還能做軟體建置以外的其他事情。




建置 Maven 2 或 3 專案

建置 Maven 2 或 3 的專案。Jenkins 會利用您已經寫在 POM 裡的資料，大大減少手動設定。



建立多重設定專案

適合需要大量不同設定的專案，例如在各式環境上進行測試、平台相依的建置...等作業。



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK



建立Jenkins專案

 **Jenkins**

搜尋

admin | 登出

Jenkins > First Jenkins >

General

原始碼管理

建置觸發程序

建置環境

建置

建置後動作

專案名稱

First Jenkins

描述

[Plain text] 預覽

☐ 參數化建置

☐ 忽略舊Builds

☐ 停用Build

☐ 必要時同時執行多個建構

?

?

?

?

進階...

原始碼管理

☒ 無

☐ Git

儲存

Apply

Jenkins 基本專案組成

- General
- 原始碼管理
- 建置觸發程序
- 建置環境
- 建置
- 建置後動作



Jenkins 原始碼管理

支援大多數版本控制系統

- ClearCase (ClearCase plugin)
- CVS (CVS plugin)
- Git (Git plugin) (安裝)
- GitHub (GitHub Branch Source plugin)
- 其他



Jenkins專案-原始碼管理

Jenkins > First Jenkins >

General 原始碼管理 建置觸發程序 建置環境 建置 建置後動作

原始碼管理

☐ 無

☒ Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

儲存庫瀏覽器

Additional Behaviours

遠端觸發建置 (例如: 透過 Script 腳本)



Jenkins 建置觸發程序

- 遠端觸發建置 (例如: 透過 Script 腳本)
- 在其他專案建置後建置
- 定期建置
- 其他



Jenkins專案-原始碼管理

Jenkins > First Jenkins >

General 原始碼管理 **建置觸發程序** 建置環境 建置 建置後動作

建置觸發程序

- ☐ 遠端觸發建置 (例如: 透過 Script 腳本) ?
- ☐ Build after other projects are built ?
- ☐ 定期建置 ?
- ☐ 輪詢 SCM ?

建置環境

- ☐ With Ant ?

建置

新增建置步驟 ▾

建置後動作

新增建置後動作 ▾

儲存 Apply



Jenkins 建置環境

建置環境設定

- With Ant (Ant Plugin)
- Node & npm (NodeJS Plugin)
- Config File (Config File Provider Plugin)
- 其他



General

原始碼管理

建置觸發程序

建置環境

建置

建置後動作

- ☐ 定期建置
- ☐ 輪詢 SCM



建置環境

- ☐ Provide Configuration files
- ☐ Provide Node & npm bin/ folder to PATH
- ☐ With Ant



建置

新增建置步驟 ▾

- Execute NodeJS script
- Invoke Gradle script
- Provide Configuration files
- 叫用 Ant
- 呼叫最上層 Maven 目標
- 執行 Shell
- 執行 Windows 批次指令

儲存

Apply

Jenkins 建置

組合多個建置步驟完成建置

- 呼叫最上層Maven目標 (pom.xml)
- 執行Shell
- 執行Windows批次指令
- 叫用Ant (Ant Plugin)
- 叫用Gradle script (Gradle Plugin)
- 執行NodeJS script (NodeJS Plugin)
- 其他



Jenkins專案-建置環境

Jenkins > First Jenkins >

General 原始碼管理 **建置觸發程序** 建置環境 建置 建置後動作

☐ 定期建置 ?
☐ 輪詢 SCM ?

建置環境

☐ Provide Configuration files ?
☐ Provide Node & npm bin/ folder to PATH
☐ With Ant ?

建置

新增建置步驟 ▾

- Execute NodeJS script
- Invoke Gradle script**
- Provide Configuration files
- 叫用 Ant
- 呼叫最上層 Maven 目標
- 執行 Shell
- 執行 Windows 批次指令

儲存 Apply

頁面產生時間: 2018/2/6 下午 01:40:10 秒 REST API Jenkins ver. 2.80.2



Jenkins專案-建置環境

Jenkins > First Jenkins >

General 原始碼管理 **建置觸發程序** 建置環境 建置 建置後動作

☐ 定期建置 ?
☐ 輪詢 SCM ?

建置環境

☐ Provide Configuration files ?
☐ Provide Node & npm bin/ folder to PATH
☐ With Ant ?

建置

呼叫最上層 Maven 目標 ?

Maven 版本

Goal

進階...

新增建置步驟 ▾

建置後動作

儲存 Apply



Jenkins 建置後動作

- 封存成品
- 彙總下游測試結果
- 發佈JUnit測試結果報告
- 發佈Javadoc
- 電子郵件通知
- 其他



Jenkins專案-建置環境

Jenkins > First Jenkins >

General 原始碼管理 建置觸發程序 建置環境 **建置** 建置後動作

建置

呼叫最上層 Maven 目標

Maven 版本: Maven

新增建置後動作

- 封存成品
- 建置其他專案
- 彙總下游測試結果
- 發佈 JUnit 測試結果報告
- 發佈 Javadoc
- 記錄檔案指紋，以便追蹤使用狀況
- Git Publisher
- 電子郵件通知

進階...

儲存 Apply

頁面產生時間: 2018/2/6 下午01時18分11秒 [REST API](#) [Jenkins ver. 2.89.3](#)



 [回到專案](#) [狀態](#) [變更](#) **Console Output** [View as plain text](#) [編輯建構資訊](#) [刪除這次建構](#)

終端機輸出

由使用者 [admin](#) 啟動

建置中 工作區 C:\Program Files (x86)\Jenkins\workspace\First Jenkins

[First Jenkins] \$ cmd.exe /C "D:\Maven\bin\mvn.cmd compile && exit %%ERRORLEVEL%%"

[INFO] Scanning for projects...

[INFO]

[INFO] -----

[INFO] Building MavenProj 0.0.1-SNAPSHOT

[INFO] -----

[INFO]

[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ MavenProj ---

[WARNING] Using platform encoding (MS950 actually) to copy filtered resources, i.e. build is platform dependent!

[INFO] skip non existing resourceDirectory C:\Program Files (x86)\Jenkins\workspace\First Jenkins\src\main\resources

[INFO]

[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ MavenProj ---

[INFO] Nothing to compile - all classes are up to date

[INFO] -----

[INFO] BUILD SUCCESS

[INFO] -----

[INFO] Total time: 1.012 s

[INFO] Finished at: 2018-02-06T13:09:15+08:00

[INFO] Final Memory: 8M/245M

[INFO] -----

Finished: SUCCESS

建置JENKINS專案



Jenkins專案-建置環境

 **Jenkins**

搜尋

admin | 登出

Jenkins > First Jenkins >

開啟自動更新頁面

 回到儀表板

 狀態

 變更

 工作目錄

 馬上建置

 刪除專案

 組態

專案 First Jenkins

 新增描述

停用專案

 工作區

 最近變更

建置歷程

趨勢

#3	2018/2/6 下午 1:22
#2	2018/2/6 下午 1:21
#1	2018/2/6 下午 1:09

 RSS 摘要: 全部

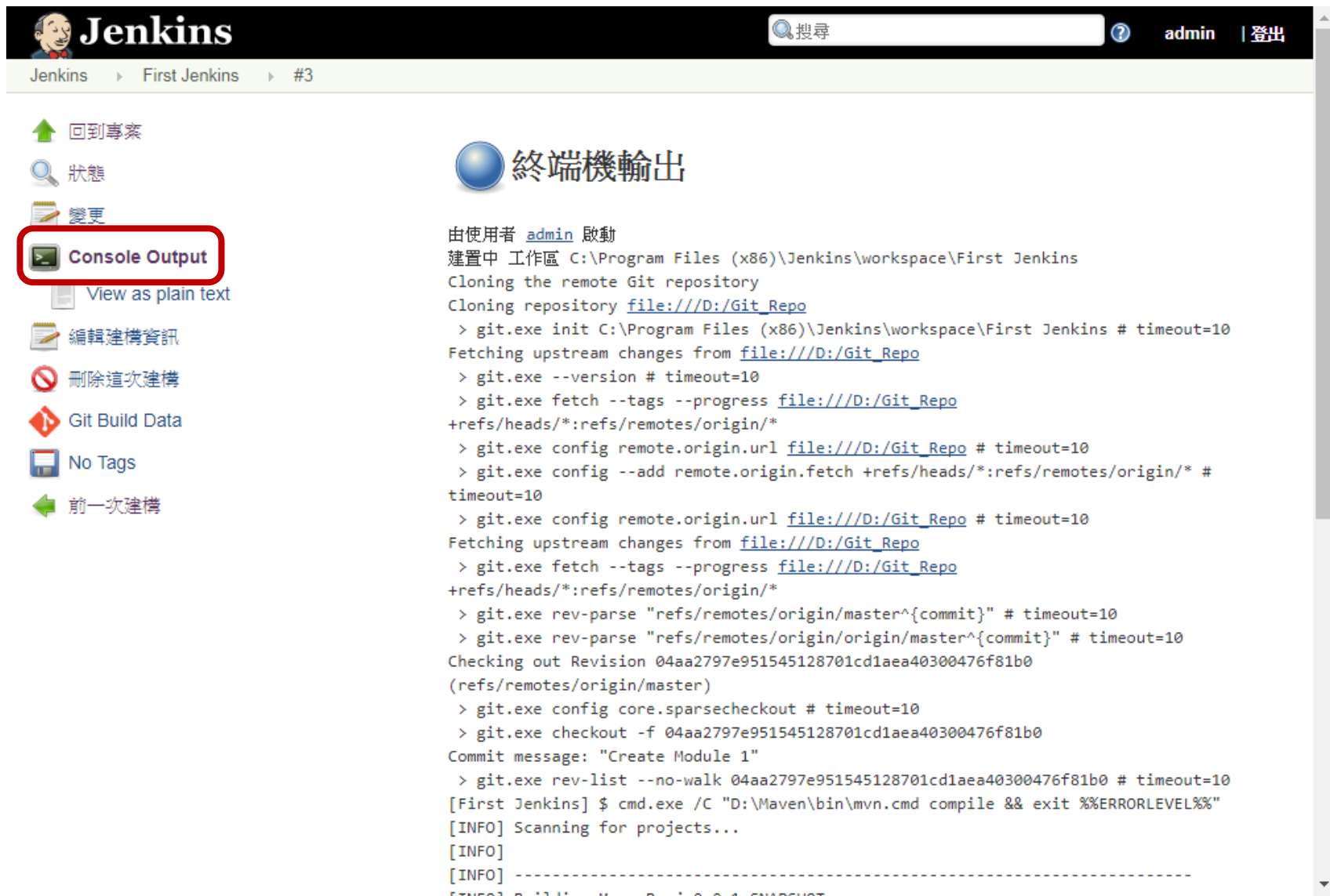
 RSS 摘要: 失敗

永久連結

- [最新建置 \(#3\). 4 分 39 秒 以前](#)
- [最新穩定建置 \(#3\). 4 分 39 秒 以前](#)
- [最新成功建置 \(#3\). 4 分 39 秒 以前](#)
- [Last completed build \(#3\). 4 分 39 秒 以前](#)

頁面產生時間: 2018/2/6 下午 01 時 27 分 16 秒 [REST API](#) [Jenkins ver. 2.89.3](#)

Jenkins專案-建置環境



The screenshot shows the Jenkins web interface. At the top, there's a header with the Jenkins logo, a search bar, and the user 'admin' with a '登出' (Logout) link. Below the header, the breadcrumb path is 'Jenkins > First Jenkins > #3'. The left sidebar contains several links: '回到專案' (Return to project), '狀態' (Status), '變更' (Change), 'Console Output' (highlighted with a red box), 'View as plain text', '編輯建構資訊' (Edit build information), '刪除這次建構' (Delete this build), 'Git Build Data', 'No Tags', and '前一次建構' (Previous build). The main content area is titled '終端機輸出' (Terminal Output) and displays the console output of the build. The output shows the build being started by 'admin', cloning a Git repository from 'file:///D:/Git_Repo', and running a series of Git commands to initialize, fetch, and checkout the repository. The build is currently in the 'Building' state.

由使用者 [admin](#) 啟動
建置中 工作區 C:\Program Files (x86)\Jenkins\workspace\First Jenkins
Cloning the remote Git repository
Cloning repository [file:///D:/Git_Repo](#)
> git.exe init C:\Program Files (x86)\Jenkins\workspace\First Jenkins # timeout=10
Fetching upstream changes from [file:///D:/Git_Repo](#)
> git.exe --version # timeout=10
> git.exe fetch --tags --progress [file:///D:/Git_Repo](#)
+refs/heads/*:refs/remotes/origin/*
> git.exe config remote.origin.url [file:///D:/Git_Repo](#) # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git.exe config remote.origin.url [file:///D:/Git_Repo](#) # timeout=10
Fetching upstream changes from [file:///D:/Git_Repo](#)
> git.exe fetch --tags --progress [file:///D:/Git_Repo](#)
+refs/heads/*:refs/remotes/origin/*
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
> git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10
Checking out Revision 04aa2797e951545128701cd1aea40300476f81b0
(refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 04aa2797e951545128701cd1aea40300476f81b0
Commit message: "Create Module 1"
> git.exe rev-list --no-walk 04aa2797e951545128701cd1aea40300476f81b0 # timeout=10
[First Jenkins] \$ cmd.exe /C "D:\Maven\bin\mvn.cmd compile && exit %ERRORLEVEL%"
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS



JENKINS結合JUNIT單元測試 (JUNIT PLUGIN)



Jenkins專案-JUnit測試

Jenkins > First Jenkins >

General 原始碼管理 建置觸發程序 建置環境 **建置** 建置後動作

建置

呼叫最上層 Maven 目標 X ?

Maven 版本

Goal ▼

進階...

新增建置步驟 ▼

建置後動作

發佈 JUnit 測試結果報告 X ?

測試報告 XML

用來指定產生的 XML 報告原始檔的 [Fileset 'includes'](#) 設定，例如 'myproject/target/test-reports/*.xml'。檔案集的基底目錄就是[工作區目錄](#)。

☐ 保留完整的標準輸出及標準錯誤內容 ?

Health report amplification factor ?

1% failing tests scores as 99% health. 5% failing tests scores as 95% health

☐ Do not fail the build on empty test results ?

儲存 Apply



Jenkins專案-JUnit測試

Jenkins > First Jenkins >

General 原始碼管理 建置觸發程序 建置環境 建置 建置後動作

新增建置步驟 ▾

建置後動作

發佈 JUnit 測試結果報告 ✕ ?

測試報告 XML

用來指定產生的 XML 報告原始檔的 [Fileset 'includes'](#) 設定，例如 'myproject/target/test-reports/*.xml'。檔案集的基底目錄就是 [工作區目錄](#)。

☐ 保留完整的標準輸出及標準錯誤內容 ?

Health report amplification factor ?

1% failing tests scores as 99% health. 5% failing tests scores as 95% health

Allow empty results ☐ Do not fail the build on empty test results ?

新增建置後動作 ▾

儲存 Apply

頁面產生時間: 2018/2/6 下午02時43分53秒 [REST API](#) [Jenkins ver. 2.89.3](#)



Jenkins專案-JUnit測試結果



Jenkins專案-JUnit測試結果

 **Jenkins**

搜尋

admin | 登出

Jenkins > First Jenkins > #16 > 測試結果 > (root)

開啟自動更新頁面

 回到專案

 狀態

 變更

 Console Output

 編輯建構資訊

 歷程

 Git Build Data

 No Tags

 測試結果

 上一次建置

測試結果: (root)

1 項失敗 (+1)

3 項測試 (+2)
花費 33 毫秒。
[新增描述](#)

所有失敗的測試

測試名稱	花費時間	累積次數
+ Module_1_Test.testMinus	0 毫秒	1

所有測試

類別	花費時間	失敗 (差異)	略過 (差異)	Pass (差異)	總計 (差異)
Module_1_Test	33 毫秒	1 +1	0	1	2 +1
Module_2_Test	0 毫秒	0	0	1 +1	1 +1

頁面產生時間: 2018/2/6 下午02時55分34秒 [REST API](#) [Jenkins ver. 2.89.3](#)

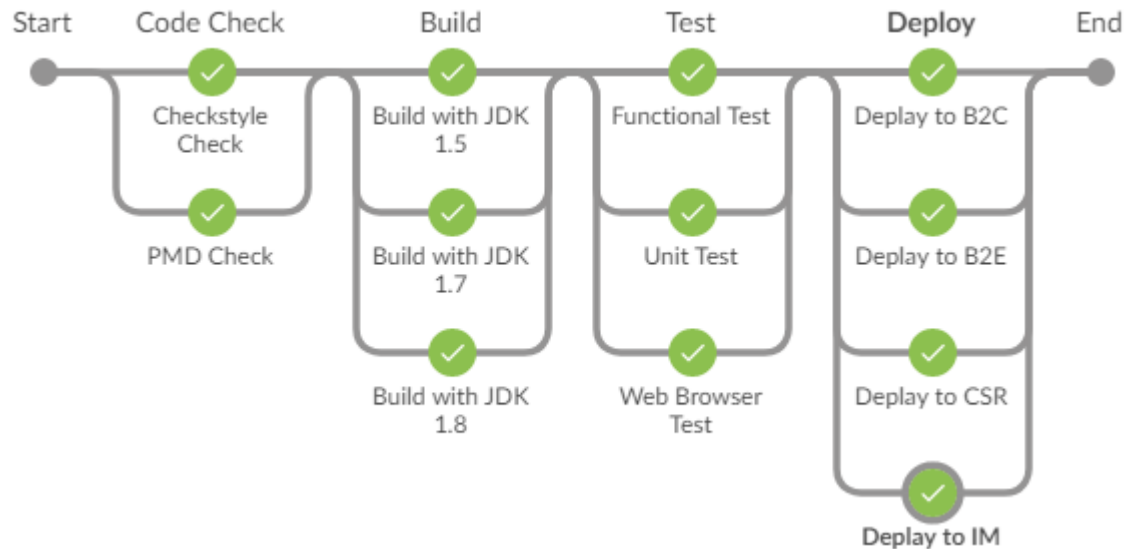


JENKINS PIPELINE



Jenkins Pipeline

- Jenkins工作流程框架 (Jenkins Pipeline Plugin)
- 用來實現平行複雜的建置流程
- 使用Groovy DSL撰寫於文字檔(Jenkinsfile)
- 配合Blue Ocean可讓Pipeline更視覺化來作業



網頁自動化框架 SELENIUM



網頁功能測試目的

- 測試使用者介面的可用性
- 對網頁功能流程進行黑箱測試
- 達到開發的持續整合和持續發佈(DevOps)



開源第三方網頁自動化套件 - Selenium

- 瀏覽器自動化測試框架和工具集
- 主要元件

Selenium IDE :

Firefox 外掛，可用來錄製、重播和匯出腳本，並可進行測試

Selenium WebDriver :

驅動瀏覽器，支援各式瀏覽器和行動平台

Selenium WebDriver API :

撰寫執行動作腳本，支援包含Java、Python、JavaScript等程式語言

Selenium Server :

透過 Selenium RC Server 讓測試腳本能在遠端多台機器上同時執行



Selenium 版本

	Selenium 2	Selenium 3
目前最新版本	2.53.1 (不再更新)	3.9.1
Java版本	2.46 Java 6 2.47 Java 7	Java 8
WebDriver	第三方開發	各瀏覽器開發商提供
WebDriver API		W3C 規格
升級優點		API幾乎可以完全套用 修正錯誤和增加測試穩定度



Selenium IDE

- 圖型化介面的測試案例建立和測試工具
- 支援瀏覽器：Firefox
- 可匯出成Java等測試程式碼



Selenium IDE

播放測試案例

The screenshot shows the Selenium IDE 2.9.1 interface. The main window is titled 'Untitled (untitled suite) - Selenium IDE 2.9.1 *'. The menu bar includes '檔案 (F)', '編輯 (E)', 'Actions (A)', 'Options (O)', and '說明'. The 'Base URL' is set to 'https://tw3.cathaylife.com.tw/eai/EHWeb/login?sys=EH'. The toolbar has buttons for 'Fast', 'Slow', 'Run', 'Stop', 'Refresh', and 'Help'. The 'Test Case' list on the left shows 'Untitled *'. The main area displays a table of commands:

Command	Target	Value
open	https://tw3.cathaylife.com.tw/eai/EHWeb...	
type	id=userID	E28817861D
type	id=passWD	E28817861D
clickAndWait	id=btn_submit	
selectWindow	name=leftFrame	
click	css=div.open_folder	
click	link=請購單輸入	
selectWindow	name=popupwin_1481785235754	
select	id=BUD_DIV_NO_I	請選擇

Below the table is a 'Command' field with 'open' selected, and 'Target' and 'Value' fields. The 'Target' field contains 'https://tw3.cathaylife.com.tw/eai/EHWeb/login?sys=EH'. The 'Value' field is empty. To the right of the interface, there are three red annotations: '錄製腳本' (Record Script) pointing to the 'Run' button, '測試腳本' (Test Script) pointing to the 'Test Case' list, and '腳本編輯' (Script Edit) pointing to the 'Command' field. At the bottom, the 'Log' tab is selected, showing the following log entries:

```
[info] Playing test case Untitled
[info] Executing: |open | https://tw3.cathaylife.com.tw/eai/EHWeb/login?sys=EH | |
[info] Executing: |type | id=userID | E28817861D |
[info] Executing: |type | id=passWD | 12345678 |
[info] Executing: |clickAndWait | id=btn_submit | |
```

On the left side of the screenshot, there is a red annotation '測試Log' (Test Log) pointing to the log tab.

Selenium WebDriver

- Selenium 提供各種瀏覽器驅動執行檔

程式範例

```
// 指定 Selenium IE WebDriver 執行檔位置
System.setProperty("webdriver.ie.driver", "D:/Selenium/IEDriverServer.exe");

// 指定瀏覽器相關參數
DesiredCapabilities capability = DesiredCapabilities.internetExplorer();
capability.setPlatform(Platform.WINDOWS);

// 建立 IE WebDriver 物件
WebDriver driver = new InternetExplorerDriver(capability);
```



Selenium WebDriver API

- Selenium 提供各種主流程式開發語言 API與瀏覽器互動
- 包含Java、C#、Ruby、Python、Javascript (Node.js)

收貨單位輸入						
收貨單位名稱	收貨人姓名	收貨人聯絡地址	收貨人聯絡電話	收貨人聯絡手機	應收貨數量	希望交貨日期
9300600 投資程設科 索引 1	吳o豫	台北市內湖區陽光街288號8樓	02-0227551399#2521	0911123456	2	2016-12-31
<div>加入 修改 刪除</div>						

程式範例

// 索引按鈕

1 driver.findElement(By.id("BTN_INDEX")).click();



Selenium WebDriver API

2

單位索引

公司別	總公司	區部別	投資資訊部	單位別	投資程設科
			確認		

程式範例

2

```
driver.switchTo().frame(0);  
Thread.sleep(2000);
```

3

```
Select select1 = new Select(driver.findElement(By.id("DEPT_KIND")));  
select1.selectByVisibleText("投資資訊部");  
Thread.sleep(2000);
```

4

```
Select select2 = new Select(driver.findElement(By.id("DIV_KIND")));  
select2.selectByVisibleText("投資程設科");
```

5

```
driver.findElement(By.id("btn_confirm")).click();  
Thread.sleep(2000);
```



Selenium WebDriver API

6

收貨單位輸入

收貨單位名稱	收貨人姓名	收貨人聯絡地址	收貨人聯絡電話	收貨人聯絡手機	應收貨數量	希望交貨日期
9300600 投資程設科 索引	吳o豫 7	台北市內湖區陽光街288號8樓	02-0227551399#2521	0911123456	2 8	2016-12-31 9

加入 修改 刪除 10

程式範例

6

7

8

9

10

```
// 切回原本畫面
driver.switchTo().parentFrame();
driver.switchTo().frame(driver.findElement(By.name("LeftFrame")));
driver.switchTo().frame(driver.findElement(By.name("mainFrame")));

driver.switchTo().frame(1);
Select select3 = new Select(driver.findElement(By.id("RCV_EMP_ID")));
select3.selectByVisibleText("吳o豫");
driver.findElement(By.id("RCVABL_CNT")).sendKeys("2");
driver.findElement(By.id("EXPT_DLVRY_DT")).sendKeys("2016-12-31");
driver.findElement(By.id("BTN_ADD")).click();
```



國泰人壽

國泰金控

Selenium 方法

1. id : `findElement(By.id())`
2. name : `findElement(By.name())`
3. class : `findElement(By.className())`
4. tag : `findElement(By.tagName())`
5. Link (超連結文字模糊搜尋) : `findElement(By.linkText())`
6. partialLink (超連結文字模糊搜尋) :
`findElement(By.partialLinkText())`
7. xpath : `findElement(By.xpath())`
8. css : `findElement(By.cssSelector())`



Selenium Demo

1. CathaySelenium
2. CathaySelenium3