# 測試腳本撰寫說明

## Selenium

### WebDriver

Chrome

ChromeDriverManager.getInstance().setup();  
WebDriver driver = new ChromeDriver();  
driver.manage().timeouts().pageLoadTimeout(5, TimeUnit.**SECONDS**);

IE

InternetExplorerDriverManager.getInstance().setup();  
// System.setProperty("webdriver.ie.driver", "D:\\IE\\IEDriverServer.exe");  
 DesiredCapabilities ieCapabilities = DesiredCapabilities.internetExplorer();  
  
 ieCapabilities.setCapability(InternetExplorerDriver.**INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS**, true);  
 ieCapabilities.setCapability("nativeEvents", false);  
 ieCapabilities.setCapability("unexpectedAlertBehaviour", "accept");  
 ieCapabilities.setCapability("disable-popup-blocking", true);  
 ieCapabilities.setCapability(InternetExplorerDriver.**ENABLE\_PERSISTENT\_HOVERING**, false);  
 ieCapabilities.setCapability(InternetExplorerDriver.**REQUIRE\_WINDOW\_FOCUS**, false);  
 ieCapabilities.setJavascriptEnabled(true);  
  
 WebDriver driver = new InternetExplorerDriver(ieCapabilities);  
 driver.manage().timeouts().pageLoadTimeout(5, TimeUnit.**SECONDS**);

需額外配置regedit

32位元

**HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE\_BFCACHE**

**64位元**

**HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE\_BFCACHE**

**都在上面路徑下添加「DWORD」iexplore.exe值為0**

參考https://blog.csdn.net/chengly0129/article/details/68482829

### 取得畫面元素

以下是取得欄位的text或value的方法

WebElement element = driver.findElement(By.id("username1"));

element.getText();

element.getAttribute("value")

### 對畫面欄位塞入資料

1. 使用套件方法去操作
2. 直接使用Script操作

WebElement element = driver.findElement(By.id("username1"));  
element.sendKeys("admin");  
  
JavascriptExecutor js = (JavascriptExecutor) driver;  
js.executeScript("$('#password1').val('12345');");

### 按鈕點擊

1. 直接對元件執行點擊，不建議是因為IE不支援此方法。
2. 直接使用Script點擊。

element.click();  
js.executeScript("$('#testBtn').click();");

### 彈出視窗

操作彈出視窗

註：若操作時會有彈出視窗，需要先結束彈出視窗才可動作，若在操作時彈出視窗剛好生成會報錯，需要做好這個控管

Alert alert\_box = driver.switchTo().alert();  
alert\_box.dismiss();

## Pepeline

### 基本腳本

node {  
 stage('step1'){

…  
 }

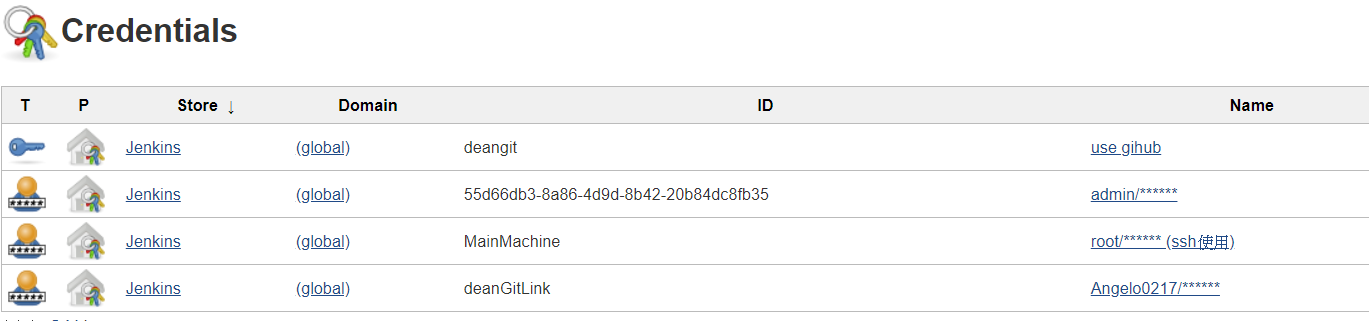
stage('step2'){

…  
 }

}

### Git操作

操作Jenkins配置的CredentialsId來執行commond來操作git即可



withCredentials([usernamePassword(credentialsId: 'deanGitLink', passwordVariable: 'GIT\_PASSWORD', usernameVariable: 'GIT\_USERNAME')]) {  
 withEnv(['PATH+EXTRA=/usr/sbin:/usr/bin:/sbin:/bin']) {  
 sh 'echo ${GIT\_USERNAME}'  
 sh 'git clone --single-branch --branch master https://${GIT\_USERNAME}:${GIT\_PASSWORD}@github.com/angelo0217/Springboot2Demo.git'  
 //sh('git push https://${GIT\_USERNAME}:${GIT\_PASSWORD}@<REPO> --tags')  
 }  
 }  
 }

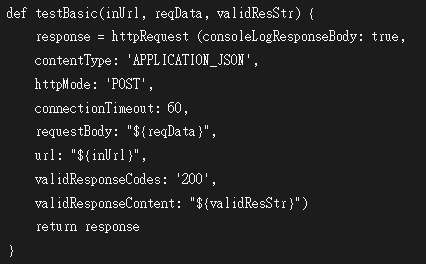
執行腳本時不用擔心console印出帳密



### HttpRequest

宣告一個HttpRequest的方法讓整個腳本呼叫

設定傳入參數



在stage間呼叫，傳入呼叫url、ReqData、回傳時比對字串

stage('TestWeb'){  
 testBasic("${testUrl}",  
 """{  
 \"serverIp\" : \"${serverIp}\",  
 \"range\":\"${range}\",  
 \"name\":\"${name}\",  
 \"age\":${age}  
 }""",  
 "\"msg\":\"success\"")  
}

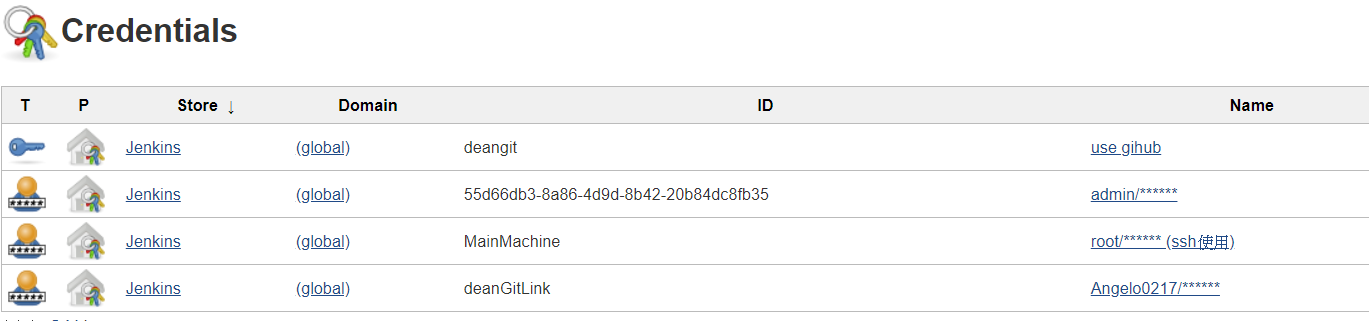
### 測試API

若要測試API，善用HttpRequest即可，可設定一個基本宣告。

在每個stage內去傳入不同參數，即可達到測試不同API的目的。

### SSH Step

配合Credentials來保護帳密資訊



需宣告連線主機使用的Credentials

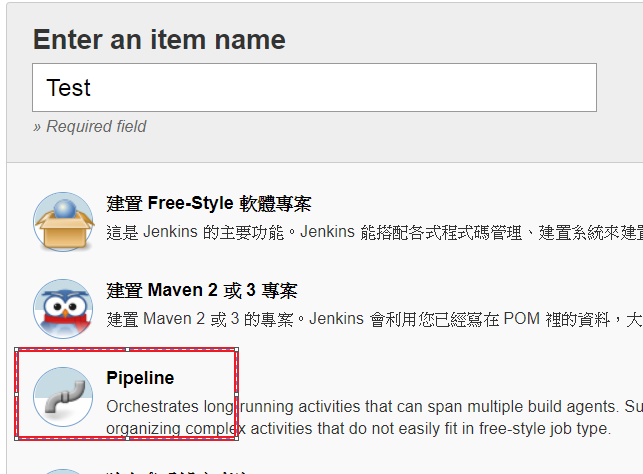
有多種方法可用，參考<https://engineering.cerner.com/blog/ssh-steps-for-jenkins-pipeline/>

def remote = [:]  
 remote.name = "node-1"  
 remote.host = "${dockerIp}"  
 remote.allowAnyHosts = true

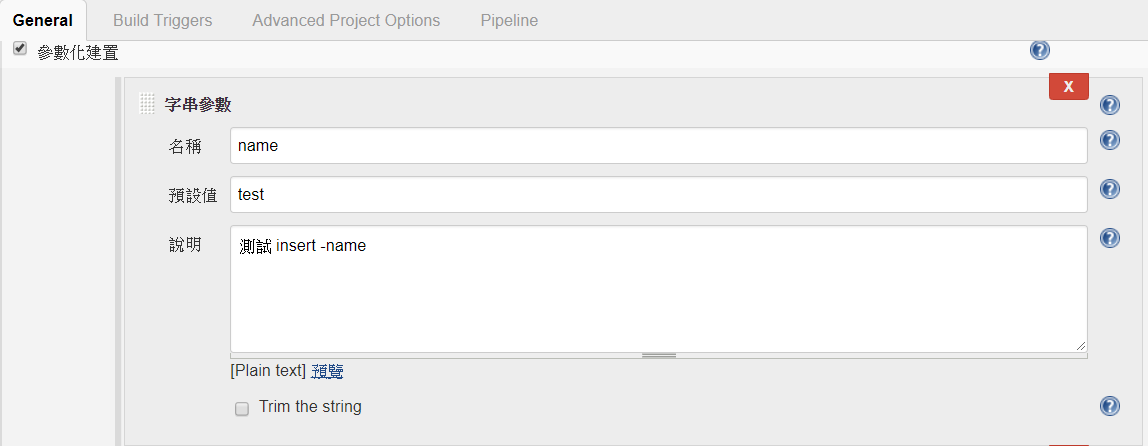
withCredentials([usernamePassword(credentialsId: 'MainMachine', passwordVariable: 'password', usernameVariable: 'user')]) {  
 remote.user = user  
 remote.password = password  
 stage('shutdown docker'){  
 try {  
 sshCommand remote: remote, command: 'docker stop web && docker rm web && docker rmi demo/web && cd /opt/vms/ && rm -rf \*';  
 }catch (exc) {  
 echo 'shutdown docker error'  
 }  
 }  
 stage('move jar'){  
 //sshCommand remote: remote, command: 'for i in {1..5}; do echo -n \"Loop \$i \"; date ; sleep 1; done'  
 sshPut remote: remote, from: '/var/jenkins\_home/workspace/testPPLhttp/Springboot2Demo/target/demo-0.0.1-SNAPSHOT.war', into: '/opt/vms'  
 sshPut remote: remote, from: '/var/jenkins\_home/workspace/testPPLhttp/Springboot2Demo/doc/Dockerfile', into: '/opt/vms'  
 }  
 stage('build docker'){  
 sshCommand remote: remote, command: 'cd /opt/vms/ && docker build -t="demo/web" .'  
 sshCommand remote: remote, command: 'docker run -itd -v /opt/vms/log:/workspace/logs --name web -d -p 12345:8080 demo/web'  
 }  
}

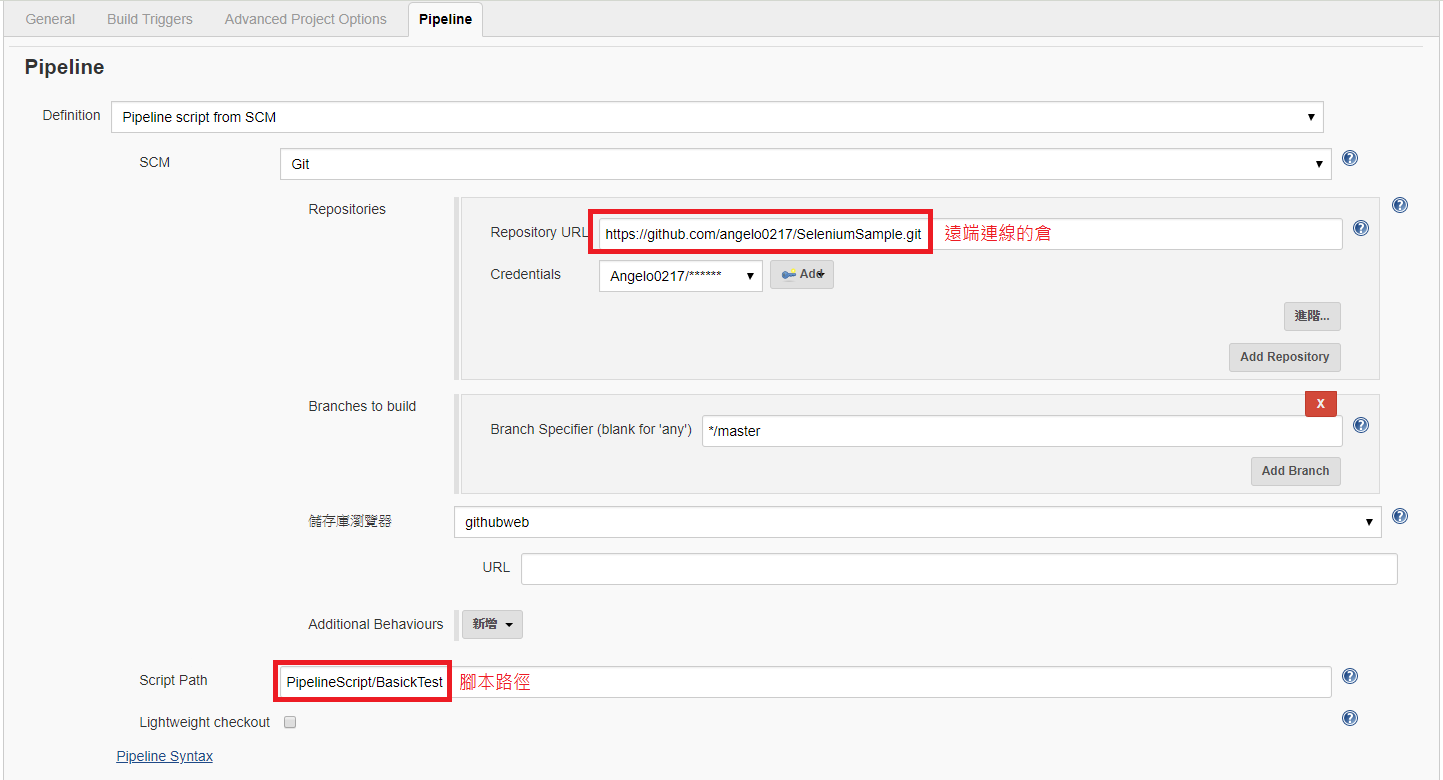
## Jenkins新增腳本





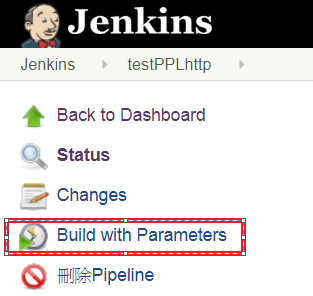
### 搭建啟動時可手動傳入參數

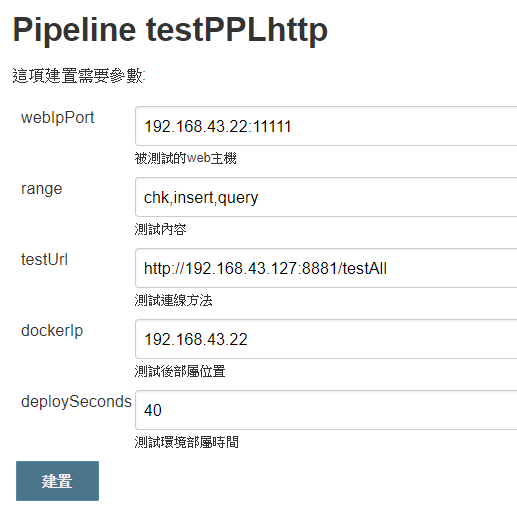




儲存後完成

### 啟動腳本





會開始跑腳本，看到的結果如下

