



0 stars

0 forks
































1 watching

1 Branch

0 Tags

Activity

Private repository

 Apple and Apple Init new repository	d0b1fd6 · last year	
 Keywords	Init new repository	last year
 Log	Init new repository	last year
 Scripts	Init new repository	last year
 Temp	Init new repository	last year
 Test_Data	Init new repository	last year
 Tool	Init new repository	last year
 Variables	Init new repository	last year
 bin	Init new repository	last year
 fedora_docker	Init new repository	last year
 API.md	Init new repository	last year
 Dockerfile	Init new repository	last year
 Jenkinsfile	Init new repository	last year
 Keywords.md	Init new repository	last year
 OCAIP_API_RF_Test.robot	Init new repository	last year
 OCAIP_Account_RF_Test.robot	Init new repository	last year
 OCAIP_Admin_RF_Test.robot	Init new repository	last year
 OCAIP_CLI_RF_Test.robot	Init new repository	last year
 OCAIP_RF_Test.robot	Init new repository	last year
 OCAIP_Team_RF_Test.robot	Init new repository	last year
 OCAIP_Topic_RF_Test.robot	Init new repository	last year
 OCAIP_UI_RF_Test.robot	Init new repository	last year
 OCAIP_Upload_RF_Test.robot	Init new repository	last year
 README.md	Init new repository	last year
 Repository.md	Init new repository	last year
 Test_Cases.xlsx	Init new repository	last year
 TestCase.json	Init new repository	last year
 TestCase.xlsx	Init new repository	last year
 clear_container.sh	Init new repository	last year
 json_to_xlsx.py	Init new repository	last year
	Init new repository	last year

python_org_search.py		
requirements.txt	Init new repository	last year
run_test.bat	Init new repository	last year
run_test_docker.sh	Init new repository	last year

README

- 歡迎補上你遇到的各種問題，把步驟說明地更詳細。並保持文件格式一致
- 投影片僅供熟悉使用，Windows 安裝步驟請參考此 README.md。另外，推薦使用 Ubuntu 環境，方便穩定
 - [Robot Framework · 資科中心教育訓練投影片 \(備用連結\)](#)
 - [Using Selenium and Robot Framework · 資科中心教育訓練投影片 \(備用連結\)](#)

For Windows 10

環境建置

1. Install Python 2.7

- 下載 [Python 2.7 版本](#)
- 安裝時點選 "Add python.exe to Path"，或是安裝完成後手動將 C:\Python27 和 C:\Python27\Scripts 手動加進 PATH
 - 控制台->系統安全性->系統。或是在本機點右鍵選取內容
 - 點選進入進階系統設定，點選進入環境變數
 - 在系統變數視窗內，點取 Path，點選進入編輯
 - 點選新增，以新增以下變數：C:\Python27，C:\Python27\Scripts

2. Install Robotframework

- 指令：`pip install robotframework`
- 安裝後確認
 - 指令：`C:\>robot --version`
 - 結果：Robot Framework 3.0 (Python 2.7.12 on win32)

3. Install SeleniumLibrary

- 指令：`pip install robotframework-seleniumlibrary`
- 安裝後確認
 - 指令：`C:\>python -m robot.libdoc SeleniumLibrary version`
 - 結果：3.1.1

4. Install RIDE (python 2 & 3 support)

- RIDE 是 Robot Framework 的 IDE，方便撰寫及執行測試案例
- i. Install wxPython first
 - download from [here](#) (檔案名稱：wxPython2.8-win32-unicode-2.8.12.1-py27)
 - 其安裝位置會是：C:\Python27\Lib\site-packages
- ii. Install robotframework-ride
 - 指令：`pip install robotframework-ride`
 - 安裝後確認

- 指令：C:\Python27\Scripts>ride.py
- 結果：會開啟 IDE
- 下完 ride.py 指令若出現 syntax error 可能是 pypubsub 版本太新，須改安裝 3.3.0 版本，pip install PyPubSub==3.3.0

5. Install Python 3

- 本專案執行時需用到 python 3 (make sure install python 3.6 or later)
 - i. 下載 Python 3 版本
 - ii. 安裝時點選 "Add python.exe to Path"，或是安裝完成後手動將 C:\Python37 和 C:\Python37\Scripts 手動加進 PATH
 - 控制台->系統安全性->系統。或是在本機點右鍵選取內容
 - 點選進入進階系統設定，點選進入環境變數
 - 在系統變數視窗內，點取 Path，點選進入編輯
 - 點選新增，以新增以下變數：C:\Python37，C:\Python37\Scripts
 - 請確認 path 中的順序 python 2 必須在 python 3 之前，在 cmd 中執行 python 及 pip 預設必須為 python 2 的環境
 - iii. 將 C:\python37 中的 python.exe 複製並重新命名為 python3.exe，C:\python37\Scripts 中的 pip.exe 複製並重新命名為 pip3.exe，若你的 Python 3 是使用 Anaconda 安裝，你的環境可能沒有 python3、pip3 指令，可將 Anaconda 的 python.exe 及 pip.exe 以上面的步驟修改放在同一資料夾。

6. 下載此 repository 後，請先在 root directory 執行：pip3 install -r requirements.txt

- 此步驟安裝的 packages，供測試案例執行 Scripts 資料夾內的 *.py 使用
- (供參考) Install chrome driver
 - 測試程式會自動進行下列的操作
 - i. 下載符合 Chrome 版本的 [ChromeDriver](#)
 - ii. 把 chromedriver.exe 放進 C:\Python27，或是將 chromedriver.exe 所在位置加進 PATH
- (供參考) 使用 Virtualenv [Robot Framework On Windows 10 and Python 3\(virtualenv\)](#)

跑測試

- 方法一：使用 RIDE IDE
 - i. 指令：C:\Python27\Scripts>ride.py
 - ii. 啟動 IDE 後，開啟此 repository，點選畫面右方「Run」分頁，將 Execution Profile 選取為「pybot」，按 Start 即開始跑測試
- RIDE IDE 簡易說明
 - RIDE IDE 的「Only run tests with these tags」和「Skip tests with these tags」兩個欄位，可使用保留字 AND 與 OR
- 方法二：使用指令
 - 指令：C:\PATH_TO_browser_automation_test>run_test.bat OCAIP_RF_Test.txt
 - 說明：run_test.bat "要測的網址" "要測的 Tag" "Source code"
 - 例：僅執行帶有 ResetPWD Tag 的測試案例，run_test.bat https://ocaip-dev.xaas.tw -i ResetPWD OCAIP_RF_Test.txt

For Ubuntu

- 在 Ubuntu16.04 使用 docker 跑測試

環境建置

README



- 更改 resolv.conf 加入院內 nameserver 140.96.254.101
 - a. sudo vim /etc/resolvconf/resolv.conf.d/head
 - b. sudo resolvconf -u

2. Install docker

- i. [在此下載](#) 或是 `sudo apt-get install docker.io`
- ii. `sudo groupadd docker`
- iii. `sudo gpasswd -a $USER docker`

3. Build docker

- 在 Dockerfile 有變動時才需要重 build
- 中英文環境測試 依據 intl.accept_languages 語言順序
- [Fedora] `docker build -t docker-robot-fedora -f ./fedora_docker/Dockerfile .`
- [Ubuntu] `docker build -t docker-robot .`

跑測試

- 使用指令 `sh run_test_docker.sh -J -m <THREAD_NUMBERS> -g <MEMORY_SIZE> -i <TAG_NAME> -e <TAG_NAME> -f <FILE> -B <BROWSER>`
- 例： `sh run_test_docker.sh -J -m 2 -g 2 -i Login -i UploadAOI -i Upload_Taxi_3 -e UploadAOI_5 -f OCAIP_RF_Test.robot -B firefox` 以 2G RAM 兩個執行緒 Firefox 瀏覽器執行 OCAIP_RF_Test.robot 內有這三個 Tag 名稱的測試案例且排除 UploadAOI_5
- 補充說明：
 - 若輸入 `-J`，則會執行 `create_issue.sh` 自動建立 issue 上傳 log.html
 - 若無輸入 `-i <TAG_NAME>`，則執行全部測試案例
 - 若輸入 `-i <TAG_NAME> -e <TAG_NAME_2>`，則執行測試案例但排除其中測試案例_2
 - 若無輸入 `-m <THREAD_NUMBERS>`，預設為 1，參數位置前後皆可（若有 crash 情況請增加 memory）
 - 若無輸入 `-g <MEMORY_SIZE>`，預設為 1g，參數位置前後皆可
 - 若無輸入 `-B <BROWSER>`，預設為 chrome，參數位置前後皆可（目前有 chrome / firefox）
 - 若無輸入 `-f <FILE>`，預設執行根目錄全部 testcases，參數位置前後皆可

測試報告

- 測試自動截圖路徑 尾端 TEARDOWN -> Compatibility Teardown -> Capture Page Screenshot
 - 詳情請見 [BAT-743](#)
- 其他除錯工具如錄影、GIF，請參考 [ScreenCapLibrary](#)

瀏覽器

- Firefox
 - Windows
 - 安裝 [Geckodriver](#)
 - 安裝 [Firefox Developer Edition](#)
 - 路徑位於 C:\Program Files\Firefox Developer Edition zh-TW
 - i18n 英文版 C:\Program Files\Firefox Developer Edition en-US
 - Linux
 - 皆包在 Docker Image 內

排程自動測試

- 腳本位於 `Scripts/create_issue.sh`
 - 須安裝 `jq` 與 `libxml2-utils`
 - [JIRA API](#) 有多種認證方式，目前採用 Basic Authorization
 - 建立 [TOKEN](#) 以 `USER@itri.org.tw:API_TOKEN` 形式並使用 base64 hash 代入
- 在 `/etc/crontab` 或使用 `crontab -e` 建立排程自動執行測試
 - 例如 `0 20 * * 1-5 root cd /PATH/browser_automation_test && bash run_test_docker.sh -J` 每一至五晚上八點執行所有測試案例並上傳結果(-J)

Jenkins 排程測試

- 目前只在分支 `jenkins-pipeline` 更變才會觸發 Jenkins build

- Kubernetes node 分配 4GB ram 故最多跑 **4 parallel jobs**，超過執行中容易中止
- Robot 插件會上傳測試報告在 **舊版 Jenkins** 界面含截圖等
- 遠端手動執行，thread 預設 4 `curl -H 'Authorization: Basic YOUR_JENKINS_TOKEN' -X POST 'https://jenkins.ai.xaas.tw/job/b/job/jenkins-pipeline/buildWithParameters?THREAD=4'`
 - 建立 Jenkins Token User設定 -> API Token
- 因於 Jenkins 跑在 GCP 上，IP 受到 Google 控管預設信箱改為 ProtonMail，詳情見 [BAT-1051](#)
- 其他修改紀錄請見 [BAT-1002](#)

流程檢查表 - 撰寫與修改測試程式

1. 修改前先去[這裡](#)開 ticket 或領 ticket
2. 例行性的小幅修改，可以直接在 master 上；大幅改動或較難解的問題，可開 branch，並在認領的 [ticket](#) 內記錄過程及 references
3. 測試案例完成後，記得更新 `TestCase.json` 檔。並留意 `README.md`、`Repository.md` 是否需要更新
4. 暫時性的修改、進行到一半的修改、後續待加強的修改，可用 `#TODO` 在程式中提示並說明，方便他人接手
5. 若有更動 `.py` 檔或是調整檔案路徑等情形，要確認 Windows 與 Docker 環境都能正確執行

速查

變更測試的網址：localhost、ocaip-dev

- 修改 `Variables/OCAIP_Variables.txt` 內的變數即可，見下表

在哪裡跑測試	<code>\${OCAIP_URL}</code> 變數	<code>\${gTest server argument}</code> 變數
ocaip-dev(預設)	https://ocaip-dev.xaas.tw/	2
localhost	http://localhost:5000 或 https://vagrant.xaas.tw	1

常見錯誤與疑難雜症

1. `StaleElementReferenceException: Message: stale element reference: element is not attached to the page document`



可能原因	解決方法
執行 click、input、Get Text 等相關 element 操作時，因網頁每隔數秒會刷新，selenium 抓到頁面刷新前的元素，導致發生錯誤。 (測試主機滿載時容易發生)	加入 retry 或 sleep 機制來降低錯誤率，但相對應的需要付出一些時間成本 (利用 Wait Until Keyword Succeeds 實作)，舉例 keyword: <code>Check Upload Score Is Correct</code>

2. `UnicodeEncodeError: 'ascii' codec can't encode characters in position xx-xx: ordinal not in range(128)`



可能原因	解決方法
click、input 的操作時，找不到 element 而回報的錯誤	重新瀏覽網頁更新 xpath (詳細實作?)
click、input 的操作時，element 包含中文因此無法解析	使用其他 xpath 或者元素作為操作目標

3. `InvalidElementStateException: Message: invalid element state: Element is not currently interactable and may not be manipulated`



可能原因	解決方法
element 尚未準備好即執行 click、input 等操作	加入 retry 或 sleep 機制來降低錯誤率，但相對應的需要付出一些時間成本 (利用 Wait Until Keyword Succeeds 實作)
element 為 disable 的狀態即執行 click、input 等操作	確認網站行為、架構是否正確，有誤則修改網頁，否則修改測試方法

4. `WebDriverException: Message: unknown error: Element is not clickable at point (xxx, xxx). Other element would receive the click`



可能原因	解決方法
某個 element 在 UI 上擋住實際要點擊的 element	確認網站行為、架構是否正確，有誤則修改網頁，否則修改測試方法
element 並不在 UI 可視範圍內	透過執行 javascript 將 element scroll 到可視範圍內解決: window.scrollTo(x,y)

5. Robot Framework 的 Cache 問題



現象	解決方法
從 .txt 讀取到 Variables 後，我們改變 .txt 中的內容並重新 Import Resource 一次，會無法讀取到更新後的內容	將新的變數放進新的 .txt 檔案中，例：auto_create_topic.py 的作法，會將新建立議題之 topic ID 寫入新的 .txt 檔內。處理紀錄見 BAT-51

6. RIDE 與 Command line (run_test.bat) 測試結果不同



現象	解決方法
在 Command line 測試環境中，Variables 裡的網址變數結尾的斜線在執行時會被去除，造成網址比對判定為錯誤，keyword: Go To Topic	不要將斜線放在 Variables 中

7. RIDE 與 Command line (run_test_docker.sh) 測試結果不同



現象	解決方法
螢幕大小截圖有差異	響應式網頁的變化以實際網頁呈現為準 (RIDE 及 Command line 目前與實際網頁皆有些微差異，可能受到瀏覽器上方的功能列影響)
Windows 與 Linux 字型不同造成排版差異	嘗試在 Linux 環境中加入微軟正黑體、Arial 等用到的字型

測試案例 failed 原因

測試案例 Tag 名稱	注意事項
【事前準備】	
Register 註冊新使用者	執行 Register 系列測試案例前，須確保網站尚無 utest 此帳號。使用 Scripts/change_data.py 執行 python3 change_data.py -s \${gTest server argument} -o DEL_UTEST，可刪除 utest 帳號
ResetPWD 忘記密碼	單獨執行此系列測試案例前，須確保網站已有 utest 此帳號。可執行 Register1_1 測試案例來建立 utest 帳號
Statistics 議題管理	執行 Statistics_1 測試案例，可使 u1 擁有「議題管理人員」角色
【特殊情形】	
20121 議題列表顯示速度	特殊：我們設定 5 秒內要顯示完成，依目前網站速度，很高機率會 failed
Discussion_13 留言兩百則	特殊：我們設定 4 秒內要顯示完成，依目前網站速度，很高機率會 failed
【網站與測試程式問題】	
Grab_UI_Answer	只有在需要更新測試比對圖片時才執行此測試，一般為註解狀態，否則底下的 UI 檢查會失敗 (Statistics_1 可使 u1 擁有「議題管理人員」角色)

將測試案例分類並同時執行，減少執行時間

指令	執行時間
(sudo 可省略)	
sudo sh run_test_docker.sh -i Page -i Login -i Profile -i ResetPWD -i Register	56m
sudo sh run_test_docker.sh -i ChangeDate -i Discussion -i Function_Verification -i Admin	60m
sudo sh run_test_docker.sh -i Statistics -i TimeBar -i RWD	54m
sudo sh run_test_docker.sh -i TeamUp -i Notification	65m

指令	執行時間
sudo sh run_test_docker.sh -i UploadAOI -i Upload_Marathon -i Except	84m

測試案例完整清單

測試案例 Tag 名稱	總執行時間	測試案例數量	說明	檔案位置
API	1m 14s	12	確認API功能正確	OCAIP_API_RF_Test.robot
Page	10m 14s	32	確認未登入的訪客看到的內容(包含英文版本)。以及 Admin 登入看到的內容	OCAIP_RF_Test.robot
Login	8m 31s	19	使用者登入	OCAIP_Account_RF_Test.robot
Profile	5m 12s	14	個人檔案頁面測試	OCAIP_Account_RF_Test.robot
ResetPWD	6m 26s	9	使用者忘記密碼	OCAIP_Account_RF_Test.robot
Register	26m 38s	42	使用者註冊。Register1業界，Register2學界，Register~註冊共用	OCAIP_Account_RF_Test.robot
Timeout	0m 29s	8	特定動作完成的執行速度	OCAIP_RF_Test.robot
Admin	12m 22s	12	確認平台管理頁面是否顯示正確	OCAIP_Admin_RF_Test.robot
ChangeDate	6m 33s	5	確認識題在各個階段可以看到與看不到的內容	OCAIP_Topic_RF_Test.robot
Discussion	14m 27s	12	使用者在議題中「討論」頁的操作	OCAIP_Topic_RF_Test.robot
Except	4m 2s	4	異常操作行為	OCAIP_RF_Test.robot
Function_Verification	24m 59s	18	確認各種議題在各階段（報名中、進行中、已結束）其頁面顯示是否正確，並檢查未登入、未報名、已報名的各項功能	OCAIP_Topic_RF_Test.robot
Notification	10m 33s	6	確認 Notification 正確顯示	OCAIP_Team_RF_Test.robot
RWD	6m 12s	11	確認 Navbar 的功能及確認各種議題在各階段其頁面(寬度 <768)顯示是否正確	OCAIP_UI_RF_Test.robot
Statistics	37m 58s	30	確認「管理」頁面的統計數據是否正確顯示	OCAIP_Admin_RF_Test.robot
TeamUp	55m 6s	21	組隊	OCAIP_Team_RF_Test.robot
TimeBar	11m 31s	11	確認識題時間軸是否顯示正確	OCAIP_Topic_RF_Test.robot
Upload	19m 27s	12	Upload暫時註解，待新議題替換	OCAIP_Upload_RF_Test.robot
UploadAOI	29m 32s	16	上傳，並確認分數、排名是否顯示正確	OCAIP_Upload_RF_Test.robot
Upload_Hydraulic	20m 27s	12	上傳，並確認分數、排名是否顯示正確	OCAIP_Upload_RF_Test.robot
Upload_SinglePublic	20m 28s	12	SinglePublic暫時註解，待新議題替換	OCAIP_Upload_RF_Test.robot
Upload_News_Honor	24m 48s	14	上傳，並確認分數、排名是否顯示正確	OCAIP_Upload_RF_Test.robot

測試案例 Tag 名稱	總執行時間	測試案例數量	說明	檔案位置
Upload_Taxi	20m 15s	12	上傳，並確認分數、排名是否顯示正確	OCAIP_Upload_RF_Test.robot
Upload_Marathon	73m 21s	7	安全議題相關測試	OCAIP_Upload_RF_Test.robot
Upload_Braints	8m 2s	5	單檔多解答議題相關測試	OCAIP_Upload_RF_Test.robot
Upload_Team	15m 4s	6	上傳，並確認分數、排名是否顯示正確	OCAIP_Upload_RF_Test.robot
UI	30m 1s	16	確認各頁面於三種螢幕大小顯示是否正確，並比對與測試圖片的相似度	OCAIP_UI_RF_Test.robot
Grab_UI_Answer	19m 51s	3	用於更新測試圖片，一般測試時為註解狀態，更新時依序交替執行 UI_13-Grab_UI_Answer_13，產生 Test_Data/Grab_UI_Answer 資料夾	OCAIP_UI_RF_Test.robot
Stress	37m 45s	2	各類型壓力測試	OCAIP_RF_Test.robot

Releases

No releases published
[Create a new release](#)

Packages


No packages published
[Publish your first package](#)

Languages

● RobotFramework 77.0% ● Python 16.0% ● HTML 5.1% ● Dockerfile 1.1% ● Other 0.8%


Suggested workflows

Based on your tech stack

 Django


Build and Test a Django Project

Configure

 Python application

Create and test a Python application.

Configure

 Python Package using Anaconda

Create and test a Python package on multiple Python versions using Anaconda for package management.

Configure

[More workflows](#)

[Dismiss suggestions](#)