

Lab 4 Report

Name: 許哲維

Student ID: 111598066

Date: 2023.05.24

1 Test Plan

1.1 Test requirements

在 Lab 4 中，要測試的系統是 KeystoneJS，這是個開源的框架，用於在 Node.js 中開發 Database 導向的網站、應用程式與 API。KeystoneJS 使創建複雜的網站和應用程式變得容易，並帶有一個漂亮的自動生成的管理 UI。

本次實驗只需要測試要求之功能，從中設計至少 8 個測試案例並為每個特性設計一個或多個測試案例。

1.2 Test Strategy

- Post Features
 - Create a post on the Admin UI page
 - Edit a post on the Admin UI page
 - Delete a post on the Admin UI page
 - Search posts by keyword on the Admin UI page
- Comment Features
 - Create a comment on Admin UI page
 - Edit a comment on Admin UI page
 - Delete a comment on Admin UI page
- Category Features
 - Create a category on Admin UI page
 - Show posts of the specific category by pressing category name on the "Blog" page
- Enquirie Feature
 - Create an enquiry on the "Contact" page
 - Delete an enquiry on Admin UI page
- User Feature
 - Create a new user on Admin UI page (Name, Email, Phone, and Password must be set when creating the new user)

1.3 Test activities

To implement the proposed strategy, the following activities are planned to perform.

No.	Activity Name	Plan hours	Schedule Date
1	Study KeystoneJS and Docker	3	2023.05.18
2	Learn Selenium IDE and Selenium-Python	6	2023.05.19
3	Design test cases for the selected methods	4	2023.05.20
4	Implement test cases	5	2023.05.23
5	Complete Lab4 report	3	2023.05.24

1.4 Design Approach

使用 ISP 為每個待測試功能設計案例，並基於 Basis Path Coverage。測試工具將使用 Selenium IDE、Selenium-Python 與 Pytest 來達成測試需求。

1.5 Success criteria

所有所選功能之設計案例必須全部通過。

2 Test Design

To fulfill the test requirements listed in section 1.1, the following methods are selected and corresponding test cases are designed.

Test Case ID	Test Case Name	Pre-Test Setup	Test Steps	Test Data	Coverage Criteria	Expected Results	Post Tear-Down	Comments
TC-001	test_Create Post	1. KeystoneJS 正在運行 2. 有 chromedriver.exe 3. 有 Selenium-Python 4. 有 Pytest	1. 開啟 Chromedriver、2. 用 URL = \$url 跳轉到登錄畫面、3. 用 \$username 與 \$password 以管理員身分登錄、4. 選擇【Posts】 5. 建立一則新貼文、6. 用 \$title 輸入新貼文標題、7. 建立貼文、8. 透過 \$title 驗證是否成功創建新貼文、9. 關閉	\$url = "http://127.0.0.1:3000/keystone/signin" \$username = "demo@keystonejs.com" \$password = "demo" \$title = {Null, Blank Space, String(一般長度、最大長度、混和字符), Digit, Symbol}	1. \$title 使用 Each Choice criterion 2. 建立貼文之相關路徑使用 Basis Path Coverage	建立一則具有正確標題的貼文。 所有不正確的輸入都會被 Catch 並顯示錯誤訊息。	None	None

			Chromedriver					
TC-002	test_SearchPostTest		<p>1. 開啟Chromedriver、2. 用URL = \$url跳轉到登錄畫面、3. 用\$username與\$password以管理員身分登錄、4. 選擇【Posts】、5. 用\$title輸入貼文標題、6. 搜尋貼文、7. 透過\$title驗證是否成功搜尋貼文、8. 關閉Chromedriver</p>	<p>\$url = "http://127.0.0.1:3001/keystone/signin" \$username = "demo@keystonejs.com" \$password = "demo" \$title = {Null, Blank Space, String(一般長度、最大長度、混和字 符), Digit, Symbol}</p>	<p>1. \$title 使用 Each Choice criterion 2. 搜尋貼文之相關路徑使用 Basis Path Coverage</p>	<p>搜尋具有正確標題的貼文。所有不正確的輸入都會被 Catch 並顯示錯誤訊息。</p>		
TC-003	test_EditPostTest		<p>1. 開啟Chromedriver、2. 用URL = \$url跳轉到登錄畫面、3. 用\$username與\$password以管理員身分登錄、4. 選擇【Posts】、5. 編輯貼文、6. 用\$state選擇發佈狀態、7. 用\$author選擇貼文作者、8. 用\$contentbrief編輯貼文大</p>	<p>\$url = "http://127.0.0.1:3000/" \$username = "demo@keystonejs.com" \$password = "demo" \$state = {Draft, Published, Archived} \$author = {所有 User} \$contentbrief = {Null, Blank Space, String(一般長度、</p>	<p>1. \$state、\$author、\$contentbrief與\$contentextended使用 Each Choice criterion 2. 編輯貼文之相關路徑使用 Basis Path Coverage</p>	<p>藉由正確的輸入編輯貼文。所有不正確的輸入都會被 Catch 並顯示錯誤訊息。</p>		

			<p>綱、9. 用 \$contentextended 編輯貼文內容、10. 儲存貼文、11. 透過 \$state 、 \$author 、 \$contentbrief 與 \$contentextended 驗證是否編輯成功、12. 關閉 Chromedriver</p>	<p>最大長度、混和字 符), Digit, Symbol} \$contentextended = {Null, Blank Space, String(一般長度、最大長度、混和字 符), Digit, Symbol}</p>			
TC-004	test_CreateCommentTest		<p>1. 開啟 Chromedriver、2. 用 URL = \$url 跳轉到登錄畫面、3. 用 \$username 與 \$password 以管理員身分登錄、4. 選擇【Posts】、5. 選擇【Comments】、6. 建立新評論、7. 用 \$author 選擇作者、8. 用 \$post 選擇貼文、9. 建立評論、10. 透過 \$author 與 \$post 驗證確認是否成功建立評論、11. 關閉 Chromedriver</p>	<p>\$url = "http://127.0.0.1:3000/keystone/signin" \$username = "demo@keystonejs.com" \$password = "demo" \$author={所有使用者} \$post={所有貼文}</p>	<p>1. \$author 與 \$post 使用 Each Choice criterion 2. 新增評論之相關路徑使用 Basis Path Coverage</p>	<p>藉由合法的使用者與貼文建立評論。所有不正確的輸入都會被 Catch 並顯示錯誤訊息。</p>	

TC-005	test_EditCommentTest	<p>1. 開啟Chromedriver、2. 用URL = \$url跳轉到登錄畫面、3. 用\$username與\$password以管理員身分登錄、4. 選擇【Posts】、5. 選擇【Comments】、6. 編輯評論、7. 用\$content編輯評論、8. 儲存評論、9. 透過\$content驗證確認是否成功編輯評論、10. 關閉Chromedriver</p>	<p>\$url = "http://127.0.0.1:3000/keystone/signin" \$username = "demo@keystonejs.com" \$password = "demo" \$content = {Null, Blank Space, String(一般長度、最大長度、混和字 符), Digit, Symbol}</p>	<p>1. \$content 使用 Each Choice criterion 2. 編輯評論之相關路徑使用 Basis Path Coverage</p>	<p>編輯一則正確的評論。所有不正確的輸入都會被 Catch 並顯示錯誤訊息。</p>	
TC-006	test_CreateGalleryTest	<p>1. 開啟Chromedriver、2. 用URL = \$url跳轉到登錄畫面、3. 用\$username與\$password以管理員身分登錄、4. 選擇【Galleries】、5. 建立一個新相簿、6. 用\$key輸入新相簿名稱、7. 建立相簿、8. 透過\$key</p>	<p>\$url = "http://127.0.0.1:3000/keystone/signin" \$username = "demo@keystonejs.com" \$password = "demo" \$key = {Null, Blank Space, String(一般長度、最大長度、混和字 符), Digit,</p>	<p>1. \$key 使用 Each Choice criterion 3. 建立相簿之相關路徑使用 Basis Path Coverage</p>	<p>建立一個新相簿。所有不正確的輸入都會被 Catch 並顯示錯誤訊息。</p>	

			驗證是否成功創建新相簿、9. 關閉 Chromedriver	Symbol}				
TC-007	test_DeleteCommentTest		1. 開啟 Chromedriver、2. 用 URL = \$url 跳轉到登錄畫面、3. 用 \$username 與 \$password 以管理員身分登錄、4. 選擇【Posts】、5. 選擇【Comments】、6. 刪除評論、7. 驗證確認是否成功刪除評論、8. 關閉 Chromedriver	\$url = "http://127.0.0.1:3000/keystone/signin" \$username = "demo@keystonejs.com" \$password = "demo"	1. 刪除評論之相關路徑使用 Basis Path Coverage	刪除評論。所有不正確的結果都會被 Catch 並顯示錯誤訊息。		
TC-008	test_DeletePostTest		1. 開啟 Chromedriver、2. 用 URL = \$url 跳轉到登錄畫面、3. 用 \$username 與 \$password 以管理員身分登錄、4. 選擇【Posts】、5. 刪除貼文、6. 驗	\$url = "http://127.0.0.1:3000/keystone/signin" \$username = "demo@keystonejs.com" \$password = "demo"	1. 刪除貼文之相關路徑使用 Basis Path Coverage	刪除貼文。所有不正確的結果都會被 Catch 並顯示錯誤訊息。		

			證確認是否成功刪除貼文、7. 關閉 Chromedriver				
TC-009	test_Delete GalleryTest		1. 開啟 Chromedriver、2. 用 URL = \$url 跳轉到登錄畫面、3. 用 \$username 與 \$password 以管理員身分登錄、4. 選擇【Galleries】、5. 刪除相簿、6. 驗證確認是否成功刪除相簿、7. 關閉 Chromedriver	\$url = "http://127.0.0.1:3000/keystone/signin" \$username = "demo@keystonejs.com" \$password = "demo"	1. 刪除相簿之相關路徑使用 Basis Path Coverage	刪除相簿。所有不正確的結果都會被 Catch 並顯示錯誤訊息。	
TC-010	test_Create User		1. 開啟 Chromedriver、2. 用 URL = \$url 跳轉到登錄畫面、3. 用 \$username 與 \$password 以管理員身分登錄、4. 選擇【Users】、5. 建立新的 User、6. 用 \$firstname 輸入名字、7. 用 \$lastname 輸入姓氏、8. 用	\$url = "http://127.0.0.1:3000/keystone/signin" \$username = "demo@keystonejs.com" \$password = "demo" \$firstname = {Null, Blank Space, String(一般長度、最大長度、混和字符), Digit,	1. \$firstname、\$lastname 與 \$email 使用 Each Choice criterion 2. 建立 User 之相關路徑使用 Basis Path Coverage	使用姓、名、E-mail、密碼、確認密碼建立新 User。所有不正確的輸入都會被 Catch 並顯示錯誤訊息。	

			<p>\$email 輸入 E-mail、</p> <p>9. 用 \$newpassword 輸入密碼、</p> <p>10. 用 \$confirmnewpassword 輸入密碼確認、</p> <p>11. 建立 User、12. 透過 \$firstname、\$lastname 與 \$email 驗證是否成功建立新 User、</p> <p>13. 關閉 Chromedriver</p>	<p>Symbol}</p> <p>\$lastname = {Null, Blank Space, String(一般長度、最大長度、混和字 符), Digit, Symbol}</p> <p>\$email = {Null, Blank Space, E-mail}</p> <p>\$newpassword = {Null, Blank Space, String(一般長度、最大長度、混和字 符), Digit, Symbol}</p> <p>\$confirmnewpassword = {Null, Blank Space, String(一般長度、最大長度、混和字 符), Digit, Symbol}</p>			
--	--	--	---	--	--	--	--

The details of the design are given below:

Lab4_TestCase.xlsx

3 Test Implementation

Section 2 中指定的測試用例設計是使用 Selenium-Python 實現的。下面給出了 3 個選定測試案例的測試腳本。其餘測試腳本實現可以在[連結](#)中找到。

Test Case ID.	Test method	Source test code
TC-001	test_CreatePostTest	<pre> def test_CreatePostTest(): # 開啟Chromedriver _service = Service(r"./chromedriver.exe") chromedriver = webdriver.Chrome(service = _service) # 跳轉到登錄畫面 chromedriver.get("http://127.0.0.1:3000/keystone/signin") # 以管理員身分登錄 chromedriver.find_element(by = By.NAME, value = "email").send_keys('demo@keystonejs.com') chromedriver.find_element(by = By.NAME, value = "password").send_keys('demo') chromedriver.find_element(by = By.CLASS_NAME, value = "css-2960tt").click() time.sleep(1.2) # 跳轉到Posts頁面 chromedriver.find_element(by = By.LINK_TEXT, value = "Posts").click() time.sleep(1.2) # 建立一則新貼文 chromedriver.find_element(by = By.CLASS_NAME, value = "css-h629qq").click() chromedriver.find_element(by = By.CLASS_NAME, value = "css-f0h633").send_keys('Test') chromedriver.find_element(by = By.CLASS_NAME, value = "css-3dhv4s").find_element(by = By.CLASS_NAME, value = "css-h629qq").click() # 驗證是否成功創建新貼文 chromedriver.get("http://127.0.0.1:3000/keystone/posts") result_table = chromedriver.find_element(by = By.CLASS_NAME, value = "Table").text assert 'Test' in result_table # 關閉Chromedriver chromedriver.close() </pre>
TC-003	test_EditPostTest	<pre> def test_EditPostTest(): # 開啟Chromedriver _service = Service(r"./chromedriver.exe") chromedriver = webdriver.Chrome(service = _service) # 跳轉到登錄畫面 chromedriver.get("http://127.0.0.1:3000/keystone/signin") # 以管理員身分登錄 chromedriver.find_element(by = By.NAME, value = "email").send_keys('demo@keystonejs.com') chromedriver.find_element(by = By.NAME, value = "password").send_keys('demo') chromedriver.find_element(by = By.CLASS_NAME, value = "css-3dhv4s").click() time.sleep(1.2) # 跳轉到Posts頁面 chromedriver.get("http://127.0.0.1:3000/keystone/posts") # 編輯貼文 list = chromedriver.find_elements(by = By.CLASS_NAME, value = "ItemListItem")[0].find_element(by = By.CLASS_NAME, value = "ItemListItem").get_attribute("href") chromedriver.get(list) time.sleep(1.2) chromedriver.find_element(by = By.ID, value = "react-select-3--value").click() time.sleep(1.2) chromedriver.find_element(by = By.CLASS_NAME, value = "select-menu-control").find_element(by = By.ID, value = "react-select-3--option-1").click() chromedriver.find_element(by = By.ID, value = "react-select-3--value").click() chromedriver.find_element(by = By.CLASS_NAME, value = "select-menu-control").find_element(by = By.ID, value = "react-select-3--option-0").click() chromedriver.find_element(by = By.ID, value = "react-menu-item-0-1-if").send_keys('Test123') chromedriver.find_element(by = By.ID, value = "react-menu-item-1-if").send_keys('Test123') chromedriver.find_element(by = By.CLASS_NAME, value = "css-2960tt").click() time.sleep(1.2) # 驗證是否編輯成功 result_info = chromedriver.find_element(by = By.CLASS_NAME, value = "css-0rps").text assert 'Your changes have been saved successfully' == result_info # 關閉Chromedriver chromedriver.close() </pre>
TC-009	test_DeleteGalleryTest	<pre> def test_DeleteGalleryTest(): # 開啟Chromedriver _service = Service(r"./chromedriver.exe") chromedriver = webdriver.Chrome(service = _service) # 跳轉到登錄畫面 chromedriver.get("http://127.0.0.1:3000/keystone/signin") # 以管理員身分登錄 chromedriver.find_element(by = By.NAME, value = "email").send_keys('demo@keystonejs.com') chromedriver.find_element(by = By.NAME, value = "password").send_keys('demo') chromedriver.find_element(by = By.CLASS_NAME, value = "css-2960tt").click() time.sleep(1.2) # 跳轉到Galleries頁面 chromedriver.find_element(by = By.LINK_TEXT, value = "Galleries").click() time.sleep(1.2) # 刪除相簿 chromedriver.find_element(by = By.CLASS_NAME, value = "ItemListItem-control--delete").click() chromedriver.find_element(by = By.CLASS_NAME, value = "css-t4884").click() time.sleep(1.2) # 驗證是否成功刪除相簿 result_info = chromedriver.find_element(by = By.CLASS_NAME, value = "css-pbvij").text assert 'No albums found...' == result_info # 關閉Chromedriver chromedriver.close() </pre>

4 Test Results

Environment & Infrastructure
windows 10
chromedriver
Selenium IDE
Selenium-Python、Pytest

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19045.2965]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
D:\STV\docker_test\selenium>pytest TestKeystoneJS.py
===== test session starts =====
platform win32 -- Python 3.10.7, pytest-7.3.1, pluggy-1.0.0
rootdir: D:\STV\docker_test\selenium\
collected 10 items

TestKeystoneJS.py
DevTools listening on ws://127.0.0.1:62398/devtools/browser/9747966a-0908-496d-81b9-642e047b18f0
DevTools listening on ws://127.0.0.1:62419/devtools/browser/5c6c261-d55f-4b51-a518-5575bef737ce
DevTools listening on ws://127.0.0.1:62442/devtools/browser/9f7ff08a-7700-4939-92d9-a798ab169c4f
DevTools listening on ws://127.0.0.1:62463/devtools/browser/69bc6419-792d-4b2a-9fed-7f68fab157c2
[31264:21296:0523/204412.556:ERROR:page_load_metrics_update_dispatcher.cc(178)] Invalid first_paint 0.452 s for first_image_paint 0.443 s
[31264:21296:0523/204412.618:ERROR:page_load_metrics_update_dispatcher.cc(178)] Invalid first_paint 0.452 s for first_image_paint 0.443 s
[31264:21296:0523/204412.666:ERROR:page_load_metrics_update_dispatcher.cc(178)] Invalid first_paint 0.452 s for first_image_paint 0.443 s
[31264:21296:0523/204412.887:ERROR:page_load_metrics_update_dispatcher.cc(178)] Invalid first_paint 0.452 s for first_image_paint 0.443 s
[31264:21296:0523/204413.517:ERROR:page_load_metrics_update_dispatcher.cc(178)] Invalid first_paint 0.452 s for first_image_paint 0.443 s
DevTools listening on ws://127.0.0.1:62483/devtools/browser/6d664b83-df14-46f8-bdbd-73e485dddc9
DevTools listening on ws://127.0.0.1:62503/devtools/browser/73083660-a6ab-4a44-9a69-a6beed9b3f8e
DevTools listening on ws://127.0.0.1:62524/devtools/browser/c25bc2f8-ade2-454d-bb83-ed10841e038c
DevTools listening on ws://127.0.0.1:62544/devtools/browser/1fea5e82-011c-4b79-blee-aa9d05e37b15
DevTools listening on ws://127.0.0.1:62564/devtools/browser/2eff2162-0c39-4435-8463-a739bbb33303
DevTools listening on ws://127.0.0.1:62601/devtools/browser/4dcad0df-cf1e-4d11-a774-9bf2bf61fe7
[100%]
===== 10 passed in 77.62s (0:01:17) =====
```

5 Summary

在 Lab 4 中，使用 Selenium-Python 與 ISP 設計並實現了 10 個測試案例，且執行結果皆全部通過，因此滿足 Section 1 中的要求。

而在 Lab 4 的練習中，先透過 Selenium IDE 的操作、輸出成 Pytest 的方式來學習該如何撰寫 Selenium-Python 之測試腳本，藉由這次 Lab 令我學到許多。