

# Selenium 網頁自動化程式

## 實作報告

陳英翔

### 動機

在 Appier 沛星互動科技公司擔任工讀生期間，因為本身有在自學程式語言，想說這是一個很好的機會將自己所學的技術與工作內容作結合，藉由程式導入自動化來節省人力以及時間。因此我針對日経テレコン 21 和 OFAC 美國制裁名單這兩個常用的網站，使用名為 Selenium 的套件，運用 Python 語言寫出了自動搜尋化的程式。

### 成果放置處&實測過程

個人 Github : <https://github.com/0307eito/Personal-Portfolio>

### 成果介紹

#### 1.日経テレコン 21(日經新聞資料庫)的自動搜尋化程式

日経テレコン 21 : <http://t21.nikkei.co.jp/g3/CMN0F11.do>

(1)使用的套件

Selenium：針對網頁的自動化

bs4 & request：解析 HTML 網頁原始碼

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.support.select import Select
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.keys import Keys

from bs4 import BeautifulSoup
import requests

import time
import datetime
import os
import shutil
import json
```

## (2) 搜尋關鍵字の設定

分別設定兩組關鍵字。第一組用來篩選資料的類別，而第二組是針對第一組所篩選出來的結果做搜尋。使用二次搜尋是為了避免搜尋結果過多且雜亂。這裡我將第一組關鍵字設定為音樂、專輯、新歌、得獎，而第二組關鍵字設定為我個人喜好的歌手和樂團。

```
C = '音楽 or アルバム or 新曲 or ノミネート'
officialName = ["suchmos", "サカナクション", "北島三郎", "米津玄師", "あいみょん"]
```

## (3) Chrome 驅動程式の設定

執行自動化程式之前，此處要先在 Chrome 驅動程式內設定好列印 PDF 檔的權限。

```
chopt=webdriver.ChromeOptions()
appState = {
    "recentDestinations": [
        {
            "id": "Save as PDF",
            "origin": "local",
            "account":""
        }
    ],
    "selectedDestinationId": "Save as PDF",
    "version": 2
}

prefs = {'printing.print_preview_sticky_settings.appState':
json.dumps(appState)}
chopt.add_experimental_option('prefs', prefs)
chopt.add_argument('--kiosk-printing')

driver = webdriver.Chrome(executable_path='./chromedriver',options=chopt)
driver.get('http://t21.nikkei.co.jp/')
```

## (4) 自動登入化

下半部有使用到 bs4 套件，原因是登入網站時，有時會詢問是否要再次登入而導致後續的程序中斷。我這邊運用 bs4 先找出指定的要素，並且套用至 if 文，設定當遇到此畫面時按下確認鍵，以利完成登入的程序。

```
selector= '#frmLogin2 > div > span.loginText.mt10 > label > input'
element = driver.find_element_by_css_selector(selector)
element.send_keys('帳號') #輸入個人帳號

selector= '#frmLogin2 > div > span:nth-child(2) > label > input'
element= driver.find_element_by_css_selector(selector)
element.send_keys('密碼') #輸入個人密碼

time.sleep(1)

element.send_keys(Keys.ENTER) #按下ENTER鍵

time.sleep(2)
html = driver.page_source
soup = BeautifulSoup(html, "html5lib")
a=soup.find_all('input',attrs={'type':'image'})

try:
    if a[0]['alt']=='再ログイン':
        selector= '#frmLogin1 > div > center > input'
        element= driver.find_element_by_css_selector(selector)
        element.click()
except IndexError:
    print("done")
```

(5) 登入後將網頁導向至搜尋頁面

運用 Selenium 做點擊的動作。首先在網頁中找到想要點擊的部分，再從 HTML 中找出指定的要素，套用至 Selenium 套件做執行。這邊我統一使用 selector 當作要素(也可以是 class, XPath 等)。

```
###按下 [記事検索]
selector= '#nk-mainmenu > div > div:nth-child(1) > div > ul > li.nk-menu-item.nk-menu-contract1.nk-article.haschild > p'
element= driver.find_element_by_css_selector(selector)
element.click()

time.sleep(1)

###將預設的搜尋關鍵字刪除
selector= '#contentsPanel > div > form > div.nk-list-search-header > div.nk-list-search.nk-art-list-search > div.nk-art-search-con > div.nk-art-key.fixer > input'
element= driver.find_element_by_css_selector(selector)
element.clear()

time.sleep(1)

###輸入上述的第一組關鍵字
selector= '#contentsPanel > div > form > div.nk-list-search-header > div.nk-list-search.nk-art-list-search > div.nk-art-search-con > div.nk-art-key.fixer > input'
element= driver.find_element_by_css_selector(selector)
element.send_keys(C)

time.sleep(1)

###執行搜尋
selector= '#contentsPanel > div > form > div.nk-list-search-header > button'
element= driver.find_element_by_css_selector(selector)
element.click()

time.sleep(3)

###按下[絞り込み]
selector= '#contentsPanel > div > form > div.nk-list-search-header > div.nk-list-search.nk-art-list-search > div.nk-selectbox-padding-head.nk-art-mode > select > option:nth-child(2)'
element= driver.find_element_by_css_selector(selector)
element.click()
```

#搜尋介面如下圖

4635239件です 新しい順 に 50 件ずつ 見出しを表示 印刷

新規 キーワードを入力してください 再検索 履歴

絞り込みキーワード候補 (記事の分類・主題語で絞り込み検索します)

テーマ	業界	会社・団体・人物	一般用語
政策・制度 906511	公的機関・大学 726898	政府 97279	新型 629114
行政 887662	建設 114433	安倍晋三 67910	ウイルス 565713
事件・裁判 305341	新聞・放送・出版 87510	米国政府 57430	コロナウイルス 557562
政治運営 157229	加工食品 74503	厚生労働省 45754	感染 301604
消費トレンド 150522	銀行・信用金庫 71306	菅義偉 41057	新型コロナ 209123
経済・財政 147464	自動車・二輪車 68511	ドナルド・トラ... 40937	コロナ 122975
エンターテイン... 131773	鉄道・バス・タ... 61953	自民党 36730	県内 102576
社会問題 124727	不動産・住宅 58281	内閣府 35613	工事 62040

分類から選ぶ 企業活動 政治 経済 技術 社会 業界 地域 記事種別

検索条件 詳細 HC事例C 現在の検索条件を保存 特定の記事を除く クリッピング登録

期間 1カ月 3カ月 6カ月 1年 全期間 20200514 ~ 最新 閉

## (6)寫出自動搜尋化程式

上述的步驟是為了建構出一個可以執行迴圈的環境，而現在的課題是要製作出一個執行迴圈的自動化程式。在這裡會遇到非常多的分支點，必須全部納入考量，盡可能地排出所有可能導致迴圈中斷的要因。

```
def nikkei(List):
    for i in range(len(List)):
        ###輸入上述的第二組關鍵字
        selector= '#contentsPanel > div:nth-child(2) > form > div.nk-list-search-header > div.nk-list-search.nk-art-list-search > div.nk-art-search-con > div.nk-art-key.fixer > input'
        element= driver.find_element_by_css_selector(selector)
        element.send_keys(List[i])
        ###執行搜尋
        selector= '#contentsPanel > div:nth-child(2) > form > div.nk-list-search-header > button'
        element= driver.find_element_by_css_selector(selector)
        element.click()
        time.sleep(3)
        ###展開搜尋結果
        html = driver.page_source
        soup = BeautifulSoup(html, "html5lib")
        a=soup.find_all('button',class_='nk-pn-info-listup nk-pn-info-listup-active',attrs={'name':'listUp'})
        ###有搜尋結果的情況下所執行的存檔程序
        try:
            if a[0].text=='':
                selector= '#contentsPanel > div:nth-child(2) > form > div.nk-pn-info-search-result.nk-np-info-search-result-up > button'
                element= driver.find_element_by_css_selector(selector)
                element.click()
                time.sleep(1)
                selector= 'body > div.nk-popup > div.nk-popup-content > div > div.nk-popup-btn > button.nk-popup-ok'
                element= driver.find_element_by_css_selector(selector)
                element.click()
                time.sleep(1)
                selector= '#contentsPanel > div:nth-child(2) > form > div.nk-list-search-header > div.nk-list-search-buttons > button'
                element= driver.find_element_by_css_selector(selector)
                element.click()
                time.sleep(3)
                aa='C:\\Users\\user\\Downloads\\下載.pdf'
                bb='C:\\Users\\user\\Downloads\\'+List[i]+' _日経'+'.pdf'
                os.rename(aa,bb)
                handle_array = driver.window_handles
                driver.switch_to.window(handle_array[0])

            except IndexError:
                print(List[i]+' : '+'検索結果なし')

            html = driver.page_source
            soup = BeautifulSoup(html, "html5lib")
            a=soup.find_all('div',class_='nk-pn-info-total')
            ###無搜尋結果的情況下所執行的存檔程序
            try:
                if a[0].text=='0件です':
                    selector= '#contentsPanel > div:nth-child(2) > form > div.nk-pn-info-search-result.nk-np-info-search-result-up > div.nk-list-search-buttons > button'
                    element= driver.find_element_by_css_selector(selector)
                    element.click()
                    time.sleep(3)
                    aa='C:\\Users\\user\\Downloads\\下載.pdf'
                    bb='C:\\Users\\user\\Downloads\\'+List[i]+' _日経'+'.pdf'
                    os.rename(aa,bb)

                    if a[0].text!='0件です':
                        print(List[i]+' : '+'-----検索結果あり-----')
            except IndexError:
                print('error')
            ###轉換至原本的視窗
            handle_array = driver.window_handles
            driver.switch_to.window(handle_array[0])
            ###按下[新規]
            selector= '#contentsPanel > div:nth-child(2) > form > div.nk-list-search-header > div.nk-list-search.nk-art-list-search > div.nk-selectbox-padding-head.nk-art-mode > select > option:nth-child(1)'
            element= driver.find_element_by_css_selector(selector)
            element.click()
            ###輸入上述的第二組關鍵字
            selector= '#contentsPanel > div:nth-child(2) > form > div.nk-list-search-header > div.nk-list-search.nk-art-list-search > div.nk-art-search-con > div.nk-art-key.fixer > input'
            element= driver.find_element_by_css_selector(selector)
            element.send_keys(C)
            ###執行搜尋
            selector= '#contentsPanel > div:nth-child(2) > form > div.nk-list-search-header > button'
            element= driver.find_element_by_css_selector(selector)
            element.click()
            time.sleep(3)
            ###按下[絞り込み]
            selector= '#contentsPanel > div:nth-child(2) > form > div.nk-list-search-header > div.nk-list-search.nk-art-list-search > div.nk-selectbox-padding-head.nk-art-mode > select > option:nth-child(2)'
            element= driver.find_element_by_css_selector(selector)
            element.click()
```

## (7)執行自動搜尋化程式

將搜尋關鍵字帶入函數去執行，等待程式結束，完成自動搜尋的程序。



```
nikkei(officialName)
```

#函數裡我設定回報每一個關鍵字的搜尋狀況

```
suchmos : ----- 検索結果あり -----  
サカナクション : ----- 検索結果あり -----  
北島三郎 : ----- 検索結果あり -----  
米津玄師 : ----- 検索結果あり -----  
あいみょん : ----- 検索結果あり -----
```

#PDF 檔案有被順利的建構在指定的下載目錄當中

	suchmos_日経	Microsoft Edge PDF...	123 KB
	サカナクション_日経	Microsoft Edge PDF...	192 KB
	北島三郎_日経	Microsoft Edge PDF...	198 KB
	米津玄師_日経	Microsoft Edge PDF...	179 KB
	あいみょん_日経	Microsoft Edge PDF...	199 KB

#PDF 檔內容如下圖

キーワード : 音楽 or アルバム or 新曲 or ノミネート <絞り込み> → suchmos

### 音楽ライブそろり再開 厳しい業界基準、収益に不安

2020/07/22 03:00 日本経済新聞電子版 絵写表有 1621 文字 画像有

### 音楽ライブ、そろり再開——演出・収益、道のり険しく(タ刊文化)

...ライブハウス「ラ・ママ」(同・渋谷)はSuchmos、あいみょんといった人気者...

2020/07/20 日本経済新聞 夕刊 10ページ 絵写表有 1647 文字

### とんがりエンタ=伊豆スタジオは「第2の実家」 カネコアヤノがライブ生配信-新型コロナ

2020/07/16 静岡新聞 夕刊 4ページ 絵写表有 574 文字 PDF有

### ◎【文化 ふくい】いま表現者は 1年延期をチャンスに ワンパークフェス音楽顧問・「社長」さん 生配信続け熱気保つ

2020/06/03 福井新聞 14ページ 1423 文字 PDF有

### ◎ワンパークフェス延期 福井 コロナ考慮、来年へ 出演予定アーティストら トーク、ライブ SNS配信 来月から 機運盛り上げ

2020/05/19 福井新聞 3ページ 1176 文字 PDF有

### ◎サチモスも参戦決定 福井「ワンパークフェス」 追加6組発表

2020/03/27 福井新聞 3ページ 742 文字 PDF有

### 嵐、新型コロナ感染拡大で北京公演中止...櫻井が『NEWS ZERO』で心境吐露「断腸の思い」

2020/02/18 サンケイスポーツ 1215 文字

## 2.美國 OFAC 制裁名單網站的自動搜尋化程式

OFAC 美國制裁名單：<https://sanctionssearch.ofac.treas.gov/>

(1)使用的套件一樣

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.support.select import Select
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.keys import Keys

from bs4 import BeautifulSoup
import requests

import time
import datetime
import os
import shutil
import json
```

(2)搜尋關鍵字的設定

將想查詢的公司名稱或人物姓名，以 list 做儲存。

```
other = ["Apple Inc.", "Google LLC", "KOREA COMPUTER CENTER"]
```

(3)Chrome 驅動程式的設定

與上的步驟一樣，設定好 Chrome 驅動程式內的列印 PDF 檔權限。

```
# アドレスの設定
url='https://sanctionssearch.ofac.treas.gov/'
# Google Chrome Driverの設定
chopt=webdriver.ChromeOptions()
appState = {
    "recentDestinations": [
        {
            "id": "Save as PDF",
            "origin": "local",
            "account":""
        }
    ],
    "selectedDestinationId": "Save as PDF",
    "version": 2
}
prefs = {'printing.print_preview_sticky_settings.appState':
json.dumps(appState)}
chopt.add_experimental_option('prefs', prefs)
chopt.add_argument('--kiosk-printing')
driver =
webdriver.Chrome(executable_path='./chromedriver',options=chopt
)

driver.get(url)
```



(4)寫出自動搜尋化的程式

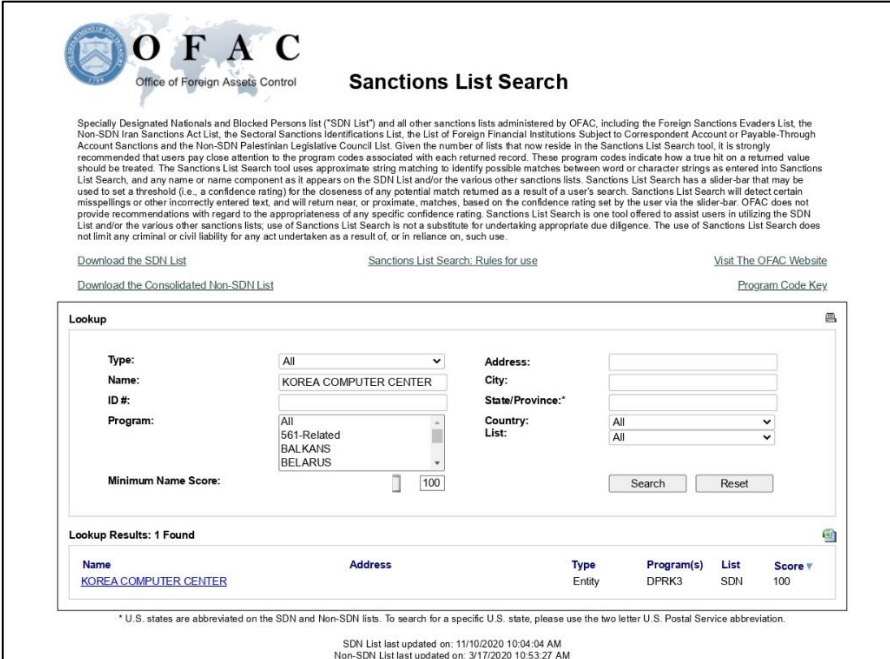
和日経テレコン 21 的網頁相比，OFAC 的網頁結構相對簡單許多，而且程序單調，因此只需要用迴圈去一個個執行搜尋，不需考慮額外的突發狀況。

```
for i in range(len(other)):  
  
    time.sleep(3)  
    # 輸入關鍵字  
    selector= '#ctl00_MainContent_txtLastName'  
    element = driver.find_element_by_css_selector(selector)  
    element.send_keys(other[i])  
    # 按下搜尋鍵  
    selector= '#ctl00_MainContent_btnSearch'  
    element = driver.find_element_by_css_selector(selector)  
    element.click()  
    time.sleep(2)  
    # 將搜尋結果存為PDF檔  
    driver.execute_script('return window.print();')  
    # 設定檔名以及下載路徑  
    a='C:\\Users\\user\\Downloads\\Sanctions List Search.pdf'  
    b='C:\\Users\\user\\Downloads\\'+other[i]+' .pdf'  
    os.rename(a,b)  
    # 為了重新搜尋下一個，將搜尋完的關鍵字清除  
    selector= '#ctl00_MainContent_txtLastName'  
    element = driver.find_element_by_css_selector(selector)  
    element.clear()  
  
print('done')
```

#PDF 檔案有被順利的建構在指定的下載目錄當中

	Apple Inc.	Microsoft Edge PDF...	93 KB
	Google LLC	Microsoft Edge PDF...	93 KB
	KOREA COMPUTER CENTER	Microsoft Edge PDF...	93 KB

# PDF 檔內容如下圖，可以看出 KOREA COMPUTER CENTER 這間公司有被納入制裁名單



**OFAC**  
Office of Foreign Assets Control

### Sanctions List Search

Specially Designated Nationals and Blocked Persons list ("SDN List") and all other sanctions lists administered by OFAC, including the Foreign Sanctions Evaders List, the Non-SDN Iran Sanctions Act List, the Sectoral Sanctions Identifications List, the List of Foreign Financial Institutions Subject to Correspondent Account or Payable-Through Account Sanctions and the Non-SDN Palestinian Legislative Council List. Given the number of lists that now reside in the Sanctions List Search tool, it is strongly recommended that users pay close attention to the program codes associated with each returned record. These program codes indicate how a true hit on a returned value should be treated. The Sanctions List Search tool uses approximate string matching to identify possible matches between word or character strings as entered into Sanctions List Search, and any name or name component as it appears on the SDN List and/or the various other sanctions lists. Sanctions List Search has a slider-bar that may be used to set a threshold (i.e., a confidence rating) for the closeness of any potential match returned as a result of a user's search. Sanctions List Search will detect certain misspellings or other incorrectly entered text, and will return near, or proximate, matches, based on the confidence rating set by the user via the slider-bar. OFAC does not provide recommendations with regard to the appropriateness of any specific confidence rating. Sanctions List Search is one tool offered to assist users in utilizing the SDN List and/or the various other sanctions lists; use of Sanctions List Search is not a substitute for undertaking appropriate due diligence. The use of Sanctions List Search does not limit any criminal or civil liability for any act undertaken as a result of, or in reliance on, such use.

[Download the SDN List](#)      [Sanctions List Search: Rules for use](#)      [Visit The OFAC Website](#)  
[Download the Consolidated Non-SDN List](#)      [Program Code Key](#)

**Lookup**

Type: All  
Name: KOREA COMPUTER CENTER  
ID #:   
Program: All  
561-Related  
BAL KANS  
BELARUS  
Minimum Name Score: 100

Address:   
City:   
State/Province:   
Country: All  
List: All

Search Reset

**Lookup Results: 1 Found**

Name	Address	Type	Program(s)	List	Score
KOREA COMPUTER CENTER		Entity	DFPRK3	SDN	100

\* U.S. states are abbreviated on the SDN and Non-SDN lists. To search for a specific U.S. state, please use the two letter U.S. Postal Service abbreviation.

SDN List last updated on: 11/10/2020 10:04:04 AM  
Non-SDN List last updated on: 3/17/2020 10:53:27 AM