**#1 Cài đặt python và vs studio code trên các hệ điều hành khác nhau**

<https://quantrimang.com/cach-cai-dat-python-tren-windows-macos-linux-140625>

<https://tek4.vn/khoa-hoc/lap-trinh-python-can-ban/cai-dat-moi-truong-phat-trien-python-tren-visual-studio-code>

<https://quantrimang.com/cai-dat-visual-studio-code-tren-ubuntu-20-04-182508>

https://visualstudio.microsoft.com/vs/mac/

**#2 Cài đặt và tìm hiểu scrapy**

Scrapy là một thư viện Python được tạo ra để quét và xây dựng các trình thu thập dữ liệu web. Nó nhanh chóng, đơn giản và có thể điều hướng qua nhiều trang web mà không mất nhiều công sức. Scrapy có sẵn thông qua thư viện Pip Installs Python

Cách cài đặt scrapy: mở cmd hoặc terminal nhập lệnh.

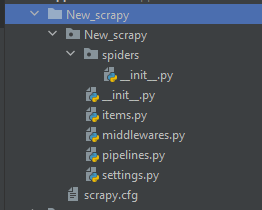
pip install Scrapy

Cách tạo project scrapy:

scrapy startproject <tên project> chạy lệnh sẽ tạo 1 đường dẫn mới với tên theo tên project. Vd: scrapy startproject New\_scrapy



Cấu trúc sau khi tạo project:



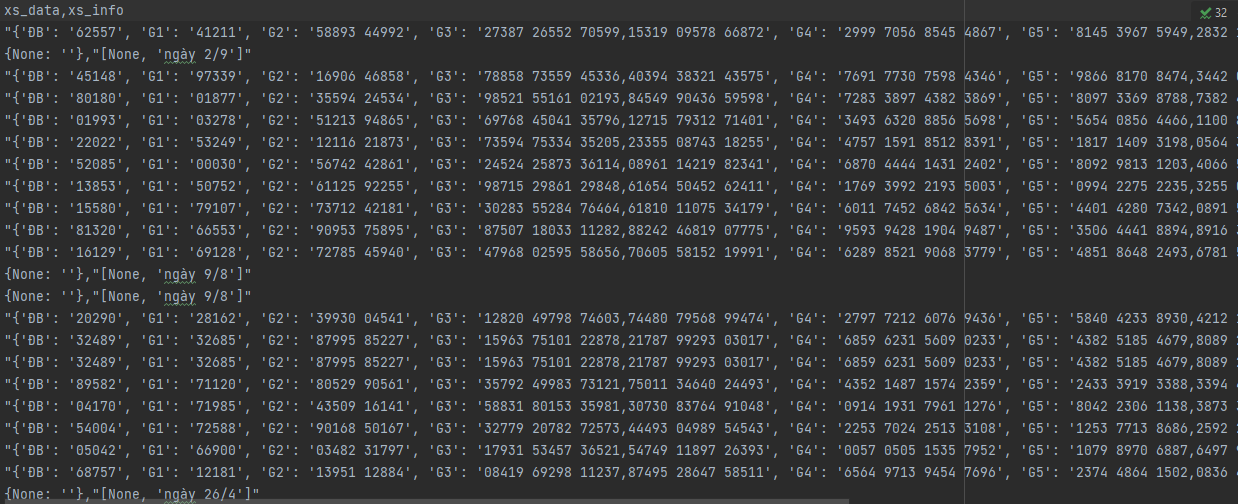
VD: E:\Source\kqxs\kqxs\spiders\kqxs.py

Chuyển con trỏ đến thư mục spider chứa file kqxs.py



Chạy lệnh: scrapy crawl kqxs -o kqxs.csv để chạy scrapy ( kqxs.csv có thể đổi thành .json hay các định dạng khác)

File kqxs.csv kết quả:



<https://docs.scrapy.org/en/latest/intro/install.html>

**#3 Cấu trúc và ý nghĩa các thành phần trong project của scapy**

scraper/

|---scraper # nơi chứa code của dự án

|---\_\_init\_\_.py

|---items.py # cơ cấu cách dữ liệu được xuất ra

|---pipelines.py # cấu hình cách spider làm việc

|---settings.py # cấu hình thêm các phần mở rộng (middlewares) và các thông số cấu hình khác

|---spiders # thư mục chứa các spider

|---\_\_init\_\_.py

|---scrapy.cfg # file cấu hình về deploy và settings của project

**#4 Cài đặt thư viện Beautiful soup:**

BeautifulSoup là một thư viện Python dùng để lấy dữ liệu ra khỏi các file HTML và XML. Nó hoạt động cùng với các parser (trình phân tích cú pháp) cung cấp cho bạn các cách để điều hướng, tìm kiếm và chỉnh sửa trong parse tree (cây phân tích được tạo từ parser). Nhờ các parser này nó đã giúp các lập trình viên tiết kiệm được nhiều giờ làm việc.

**Cài đặt thư viện:**

pip install beautifulsoup4

**Hướng dẫn sử dụng thư viện beautiful soup:**

<https://www.howkteam.vn/d/thu-vien-beautiful-soup-460>

**Ví dụ đơn giản:**

from bs4 import BeautifulSoup // import thư viện bs

import urllib.request

url = 'https://vnexpress.net' // địa chỉ trang web cần lấy dữ liệu

page = urllib.request.urlopen(url)

soup = BeautifulSoup(page, 'html.parser')

new\_feeds = soup.find(

'section', class\_='featured container clearfix').find\_all('a') // lấy toàn bộ thẻ a trong class

for feed in new\_feeds: // Tạo vòng lặp quét từng thẻ a

title = feed.get('title') // Lấy title trong thẻ a

link = feed.get('href') // Lấy link href trong thẻ a

print('Title: {} - Link: {}'.format(title, link)) // In ra kết quả

**Kết quả khi chạy đoạn code trên:**

Title: Tài xế đầu tiên dừng ôtô quá 5 phút tại trạm BOT bị phạt - Link: https://vnexpress.net/tin-tuc/thoi-su/giao-thong/tai-xe-dau-tien-dung-oto-qua-5-phut-tai-tram-bot-bi-phat-3720924.html

Title: Chi 50 triệu sửa phòng con nhỏ, tôi phải thay tất cả đồ sau 3 năm - Link: https://giadinh.vnexpress.net/tin-tuc/nha-dep/khong-gian-song/chi-50-trieu-sua-phong-con-nho-toi-phai-thay-tat-ca-do-sau-3-nam-3720639.html

Title: 8 điểm phụ nữ hay soi đàn ông nhất - Link: https://giadinh.vnexpress.net/tin-tuc/to-am/8-diem-phu-nu-hay-soi-dan-ong-nhat-3719129.html

Title: Người Việt ngày càng uống nhiều bia - Link: https://kinhdoanh.vnexpress.net/tin-tuc/hang-hoa/nguoi-viet-ngay-cang-uong-nhieu-bia-3721014.html

Title: Người gốc Việt giả danh thanh tra tống tiền hơn chục tiệm nail ở Mỹ - Link: https://vnexpress.net/tin-tuc/the-gioi/nguoi-viet-5-chau/nguoi-goc-viet-gia-danh-thanh-tra-tong-tien-hon-chuc-tiem-nail-o-my-3720994.html

Title: Hiện tượng thác nước hiếm gặp ở ven biển Philippines - Link: https://vnexpress.net/tin-tuc/khoa-hoc/chuyen-la/hien-tuong-thac-nuoc-hiem-gap-o-ven-bien-philippines-3720972.html

Title: Bé trai 5 tuổi ở Trung Quốc trốn khỏi trường mẫu giáo - Link: https://vnexpress.net/tin-tuc/giao-duc/be-trai-5-tuoi-o-trung-quoc-tron-khoi-truong-mau-giao-3721003.html

Title: Trump ân xá thủy thủ bị án tù vì chụp ảnh trong tàu ngầm hạt nhân - Link: https://vnexpress.net/tin-tuc/the-gioi/quan-su/trump-an-xa-thuy-thu-bi-an-tu-vi-chup-anh-trong-tau-ngam-hat-nhan-3720977.html

Title: So sánh CR-V bản E, Santa Fe bản tiêu chuẩn và X-Trail 2.0SL? - Link: https://vnexpress.net/tin-tuc/oto-xe-may/tu-van/so-sanh-cr-v-ban-e-santa-fe-ban-tieu-chuan-va-x-trail-2-0sl-3720560.html

Title: Đạo diễn ‘The Shape of Water’ ly dị vợ sau 32 năm sống chung - Link: https://giaitri.vnexpress.net/tin-tuc/gioi-sao/quoc-te/dao-dien-the-shape-of-water-ly-di-vo-sau-32-nam-song-chung-3721001.html

Title: Góc nhìn - Link: /tin-tuc/goc-nhin/

Title: Đàn ông và nữ quyền - Link: https://vnexpress.net/tin-tuc/goc-nhin/dan-ong-va-nu-quyen-3720424.html

**#5 Cài đặt thư viện selenium (lib hỗ trợ tự động lướt web)**

Selenium là bộ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Nó khá là giống với HP Quick Test Pro (QTP bây giờ là UFT) chỉ khác là Selenium thì tập trung vào việc tự động hoá các ứng dụng dựa trên nền tảng web. Kiểm thử được thực hiện bằng cách sử dụng công cụ Selenium thường được gọi là Kiểm thử Selenium. Selenium không chỉ là 1 công cụ độc lập mà là 1 bộ công cụ của phần mềm, mỗi bộ đều đáp ứng được nhu cầu kiểm thử khác nhau của 1 tổ chức.

**5.1 Cài đặt web driver:**

WebDriver điều khiển trình duyệt một cách nguyên bản, giống như người dùng, hoặc cục bộ hoặc trên một máy từ xa sử dụng máy chủ Selenium.

Link download: <https://www.selenium.dev/documentation/webdriver/getting_started/install_drivers/>

Kéo file .exe đến folder chứa code để dễ sử dụng;



Import web driver:

from selenium import webdriver

Khai báo driver trong py:

# 1. Khai bao bien browser  
options = webdriver.ChromeOptions()  
prefs = {"profile.default\_content\_setting\_values.notifications" : 2}  
options.add\_experimental\_option("prefs",prefs)  
options.add\_argument("start-maximized")  
browser = webdriver.Chrome('chromedriver97.exe', options=options)

Cài đặt thư viện selenium: pip install selenium

<https://www.selenium.dev/documentation/>

sử dụng selenium kết hợp với beautifulsoup để crawl content trên website

Ví dụ : sử dụng selenium và webdriver để tự động đăng nhập facebook.

Import thư viện:

from selenium import webdriver  
from selenium.webdriver.common.keys import Keys

Khai báo driver:

options = webdriver.ChromeOptions()  
prefs = {"profile.default\_content\_setting\_values.notifications" : 2}  
options.add\_experimental\_option("prefs",prefs)  
options.add\_argument("start-maximized")  
browser = webdriver.Chrome('E:\Source\chromedriver97.exe', options=options)

Get link web facebook:

browser.get("https://www.facebook.com")  
wait()

Login:

id = browser.find\_element\_by\_xpath('//\*[@id="email"]')  
id.send\_keys("zetabase3i@gmail.com")  
pwd = browser.find\_element\_by\_xpath('//\*[@id="pass"]')  
pwd.send\_keys("Langnghiem@79")  
pwd.send\_keys(Keys.ENTER)

**5.2 Sử dụng selenium, webdriver và beautiful soup để download tài liệu và cách giải captcha:**

Dịch vụ giải captcha:

[**https://github.com/AZcaptcha/azcaptchaapi**](https://github.com/AZcaptcha/azcaptchaapi)

**cài đặt Azcaptcha vào py:**

pip install azcaptchaapi

Vị trí file code: E:\Source\Webscarapy\Webscarapy\spiders\tailieu\_download modified.py

import random  
import pandas as pd  
import urllib.request  
from selenium import webdriver  
from time import sleep  
from bs4 import BeautifulSoup  
from selenium.webdriver.common.by import By  
from selenium.webdriver.common.keys import Keys  
from selenium.webdriver.common.action\_chains import ActionChains  
from selenium.common.exceptions import NoSuchElementException, TimeoutException, UnexpectedAlertPresentException  
from selenium.webdriver.support.wait import WebDriverWait  
from selenium.webdriver.support import expected\_conditions as EC  
from azcaptchaapi import AZCaptchaApi  
from tqdm import tqdm  
  
# spider\_name  
name = 'test'  
some\_attribute = "Yes|No"  
# Thông tin đăng nhập  
Username = 'Langnghiem79'  
Password = 'Langnghiem79'  
  
# Đường dẫn đến driver của webdriver  
driver\_path = r'E:\Source\chromedriver97.exe'  
  
# Đường dẫn đến list sản phẩm cần download, dạng file csv  
list\_products\_path = r'E:\Source\Webscarapy\Webscarapy\products-test.csv'  
  
# Vị trí download, đường dẫn phải tồn tại thì mới có thể tải về được  
download\_dir = r'F:\Download'  
  
# API key của bên giải quyết captcha  
API\_KEY = 'lzqixvjq2byp9zuxahp5dkwfnf1jsy8v'  
  
api = AZCaptchaApi(API\_KEY)

Khai báo browser và địa chỉ trang web;

def delay():  
 sleep(random.randint(3,5))  
# 1. Khai bao bien browser  
options = webdriver.ChromeOptions()  
prefs = {"profile.default\_content\_setting\_values.notifications" : 2, "download.default\_directory" : download\_dir}  
options.add\_experimental\_option("prefs",prefs)  
options.add\_argument("--start-maximized")  
options.add\_argument("log-level=3")  
# options.add\_argument("--headless")  
browser = webdriver.Chrome(driver\_path, options=options)  
  
  
# 2. Mở web  
browser.get("https://tailieu.vn/")

Viết hàm login:

def login():  
 # 2a. Điền thông tin vào ô user và pass  
 #Click vào button đăng nhập  
 WebDriverWait(browser, 60).until(EC.element\_to\_be\_clickable((By.XPATH,'/html/body/div[4]/div/div[5]/span/span'))).click()  
 #Click vào đăng nhập bằng tài khoản Tailieu.vn  
 WebDriverWait(browser, 20).until(EC.element\_to\_be\_clickable((By.XPATH,'/html/body/div[18]/div[1]/div[1]/div[1]/div[1]/div[1]/div[1]/div[2]/p[2]'))).click()  
 sleep(1)  
 #Nhập tài khoản, mật khẩu  
 txtLoginUsername = WebDriverWait(browser, 20).until(EC.element\_to\_be\_clickable((By.ID,"txtLoginUsername")))  
 txtLoginUsername.click()  
 txtLoginUsername.send\_keys(Username)  
  
 txtLoginPassword = browser.find\_element\_by\_id("txtLoginPassword")  
 txtLoginPassword.click()  
 txtLoginPassword.send\_keys(Password)  
  
 # 2b. Submit form  
  
 txtLoginPassword.send\_keys(Keys.ENTER)  
  
 sleep(5)

Viết hàm logout:

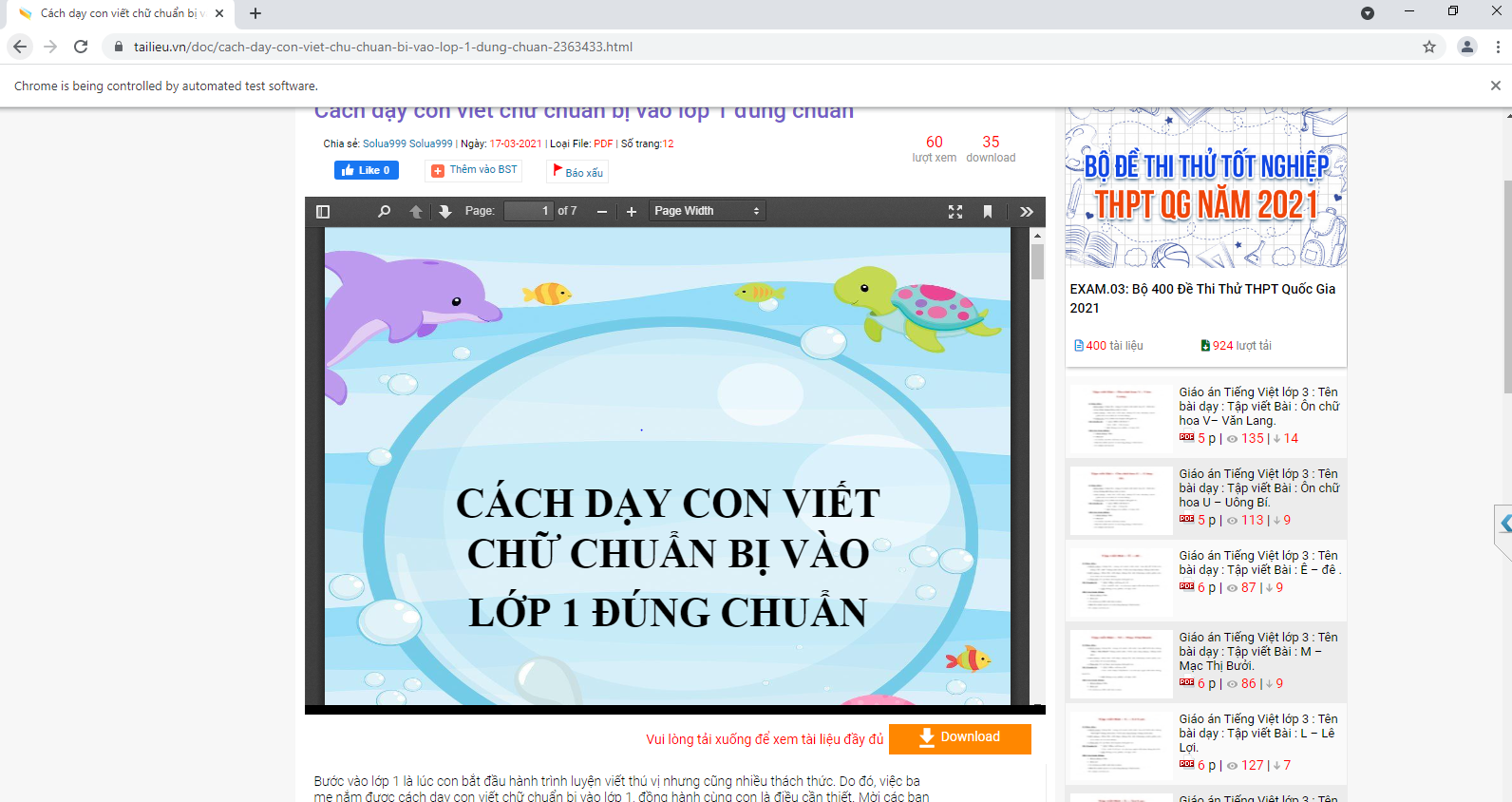
def logout():  
 # 5. Logout tài khoản sau khi hoàn thành  
 action = ActionChains(browser)  
 account\_info = browser.find\_element\_by\_css\_selector('#header > div.wrapheader > div.login > ul > li')  
 action.move\_to\_element(account\_info)  
 action.perform()  
 log\_out\_btn = browser.find\_element\_by\_css\_selector('#header > div.wrapheader > div.login > ul > li > ul > li:nth-child(7) > a')  
 log\_out\_btn.click()

Login và load các đường link product từ file csv:

# 3. Chạy login vào link tailieu.vn  
login()  
  
# 4. loop qua các đường link sản phẩm  
products\_data = pd.read\_csv(list\_products\_path)  
  
# Note: trong file phải có sẵn cột download status để khi dùng chỉ lấy những link chưa được download  
# nếu file dùng lần đầu thì uncomment dòng dưới để tự động add cột dowload status vào file và nhớ comment lại cho những lần dùng sau  
# nếu không comment với file đã sử dụng, value mới sẽ bị đè vào và những file đã được download rồi sẽ bị download lại lần nữa  
#products\_data['download'] = np.nan  
  
# lọc ra những link cần download  
to\_dowload = products\_data.loc[products\_data['download'] != 'Yes']  
to\_dowload = to\_dowload.loc[to\_dowload['download'] != 'Die']['Link'].head(10) #số lượng file cần download  
  
count\_time = 0

Truy cập các link từ to\_download và kiểm tra link còn tồn tại không, ghi vào csv nếu file chết:

for product in tqdm(to\_dowload):  
 browser.get(product)  
   
 # đề phòng trường hợp đến lúc lên download mà link đó đã chết rồi, nên kiểm tra xem link sản phẩm còn tồn tại không  
 try:  
 WebDriverWait(browser, 5).until(EC.element\_to\_be\_clickable((By.CSS\_SELECTOR, '#mainContent > div.con660.mh250 > div.margin405 > div.pnfright > p')))  
 products\_data.loc[products\_data['Link'] == product, 'download'] = 'Die'  
 products\_data.to\_csv("E:\Source\learn\_scrapy\products-test.csv")



Click vào button download và hiện ô giải captcha:

except:  
 # print("Page available")  
 #click vào download:  
 WebDriverWait(browser, 20).until(EC.element\_to\_be\_clickable((By.XPATH,'/html/body/div[6]/div/div[3]/div[3]/div[1]/div/a'))).click()  
   
 WebDriverWait(browser, 20).until(EC.element\_to\_be\_clickable((By.XPATH,'//\*[@id="lnkSubmitDL"]')))



Xử lý captcha:

captcha = True  
while captcha:  
 # print("get page source")  
 page\_source = browser.page\_source  
  
 # print("find captcha image")  
 soup = BeautifulSoup(page\_source, 'lxml')  
  
 links\_captcha\_img = soup.find\_all("span", {"id": "spm\_DocImgCaptchaDL"})[0].img['src']  
 # có trường hợp bắt không được link hình ảnh, nên sẽ sử dụng hình ảnh cũ để cố tình nhập sai cho website nhả link hình captcha mới  
 try:  
 img = urllib.request.urlretrieve(links\_captcha\_img, '../../../captcha.png')[0]  
 except ValueError:  
 img = 'captcha.png'  
 # print('captcha img downloaded')  
 #Gửi hình ảnh captcha cho AZcaptcha  
 captcha = api.solve(img)  
   
 # print('\nTry to get captcha answer')

# Nhận kết quả và ghi vào text box  
 result = captcha.await\_result()  
 #print('\nGot answer: ' + result)  
 result\_captcha\_box = WebDriverWait(browser, 20).until(EC.element\_to\_be\_clickable((By.XPATH,'/html/body/div[5]/div/div[1]/div[1]/div/div/span[1]/form/div/ul/li[1]/input')))  
 result\_captcha\_box.clear()  
 result\_captcha\_box.send\_keys(result)  
 result\_captcha\_box.send\_keys(Keys.ENTER)

Chạy lại vòng lặp while nếu captcha nhập sai:

try:  
 WebDriverWait(browser, 3).until(EC.alert\_is\_present())  
 alert = browser.switch\_to.alert  
 alert.accept()  
 #print("\nwrong captcha")  
 # áp dụng delay tránh trường hợp code chạy nhanh hơn quá trình reload hình ảnh captcha  
 delay()  
  
except TimeoutException:  
 captcha = False

Download tài liệu nếu captcha nhập đúng:

# 4a. click vào ô download của sản phẩm  
try:  
 WebDriverWait(browser, 20).until(EC.element\_to\_be\_clickable((By.XPATH,'/html/body/div[5]/div/div[1]/div[1]/div/div/span[1]/div[2]/div/div/div/ul/li/p/span/span[2]/span/a'))).click()  
 products\_data.loc[products\_data['Link'] == product, 'download'] = 'Yes'  
 products\_data.to\_csv("E:\Source\learn\_scrapy\products-test.csv")  
except:  
 logout()  
 browser.quit()  
 browser = webdriver.Chrome(driver\_path, options=options)  
 browser.get("https://tailieu.vn/")  
 login()

Logout rồi login lại khi count time = 10:

count\_time += 1  
if count\_time == 10:  
 sleep(2)  
 logout()  
 browser.quit()  
 browser = webdriver.Chrome(driver\_path, options=options)  
 browser.get("https://tailieu.vn/")  
 login()  
  
sleep(3)

Đóng browser khi download hoàn tất:

# 6. Đóng trình duyệt  
sleep(2)  
browser.quit()

**#6-1 Scrapyrt:**

Máy chủ HTTP cung cấp API để lập lịch trình thu thập dữ liệu Scrapy và đưa ra yêu cầu với trình thu thập dữ liệu.

Cài đặt scrapyrt:

<https://scrapyrt.readthedocs.io/en/stable/api.html>

pip install scrapyrt

Vị trí thư mục sau khi cài đặt:

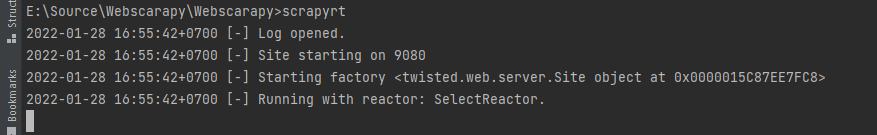
c:\users\admin\appdata\roaming\python\python37\site-packages

cách gọi scrapyrt:

chuyển đường dẫn đến thư mục chứa file scrapy.cfg và chạy lệnh scrapyrt:



Nếu không xuất hiện lỗi thì sẽ mở cổng 9080;



Sau đó mở chrome và truy cập đường dẫn:

<http://localhost:9080/crawl.json?spider_name=Tailieu&url=https://tailieu.vn/>

với spider \_name là tên của spider cần chạy.



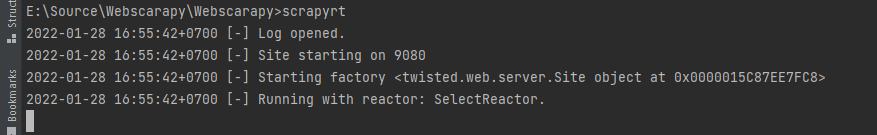
7-1: Upload file:

Sử dụng scapyrt để upload file, text1.docx là file cần upload

#Địa chỉ upload

url\_upload = "https://dieuhanh.vatco.vn/MobileLogin/InsertFile"  
response\_upload = requests.post(url\_upload, data={"CateRepoSettingId": 2247, "CreatedBy": "phancuoc\_cntt\_3i"}, files={  
 "fileUpload": (  
 'text1.docx', open('text1.docx', 'rb'), 'application/vnd.openxmlformats-officedocument.wordprocessingml.document')})  
if response\_upload.ok:  
 print("Upload completed successfully!")  
 print(response\_upload.text)  
else:  
 print("Something went wrong!")

Chạy scapyrt:

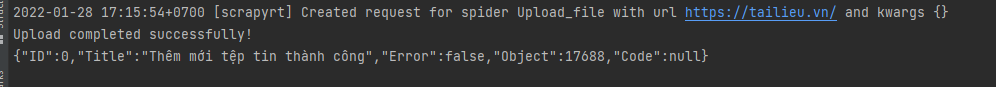


Truy cập url khi đặt tên spider:

name = 'Upload\_file'  
some\_attribute = "Yes|No"

<http://localhost:9080/crawl.json?spider_name=Upload_file&url=https://tailieu.vn/>

Thông báo nếu upload thành công:



**#8- 1 số lib hỗ trợ crawl dữ liệu qua mạng xã hội**

-Facebook: <https://pypi.org/project/facebook-scraper/>

-Instagram: <https://github.com/aahouzi/Instagram-Scraper-2021>

-Twitter: <https://github.com/twintproject/twint>

-Youtube: https://pypi.org/project/youtube-comment-scraper-python/

**#8 Cấu trúc hiện tại của project**

Trong project hiện tại có 2 project nhỏ:

- Crawl Envato lấy link sản phẩm và sử dụng selenium để có thể download sản phẩm về máy

- Thực hiện giả lập một số thao tác trên FB (sẽ được giải thích kỹ hơn bên dưới)

Envato project:

Trước tiên cần phải tạo project scrapy để có thể tạo spider để crawl dữ liệu

scrapy startproject <tên project> chạy lệnh sẽ tạo 1 đường dẫn mới với tên theo tên project. Cấu trúc project thì như bên trên đã giải thích.

Crawl Envato bằng scrapy để lấy link product: “learn\_scrapy\spiders\envato.py” file chứa spider thực

hiện tác vụ (copy con spider này về project mới tạo) và chạy spider bằng lệnh

scrapy crawl <tên spider> (1) --o output <tên file> (2)

1: tên spider đặt như nào thì dung đúng tên đó, mỗi 1 spider nên có 1 tên riêng

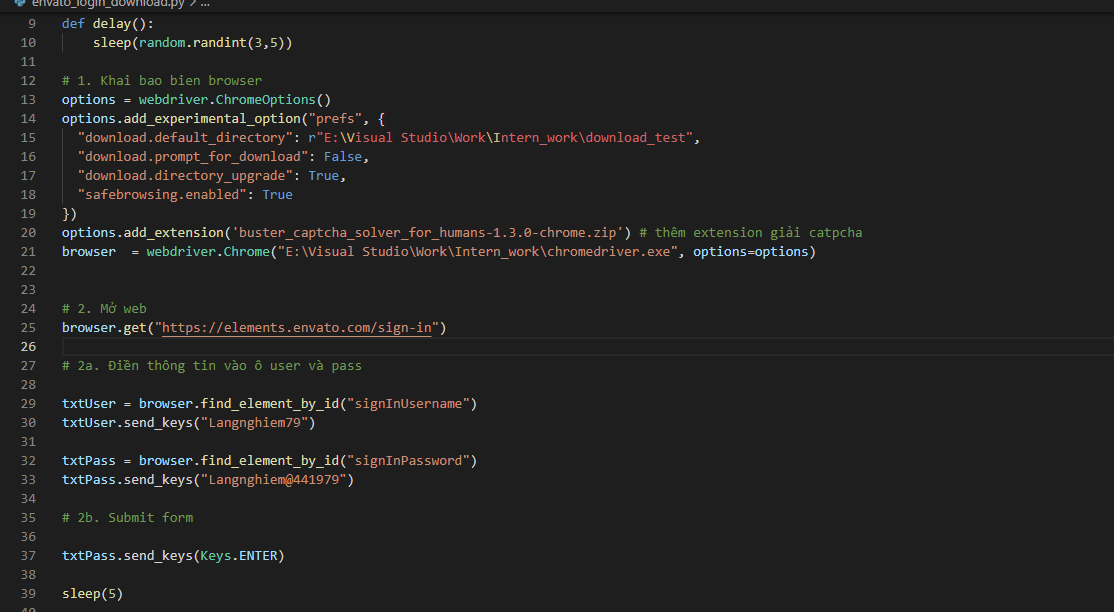
2: ouput dữ liệu ra 1 file ngoài để sau đó sử dụng với selenium

Sau khi đã có dữ liệu lấy được từ spider tiếp theo, chạy file “envato\_login\_download.py”.

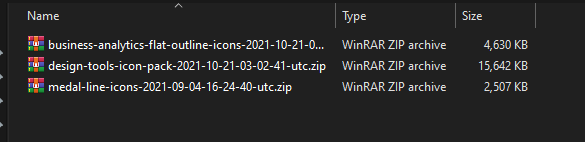
File này sẽ sử dụng selenium để giả lập hành vi người dung, login vào trang web và thông qua những đường link được lấy về từ scrapy, selenium sẽ đến từng trang trong đó và download sản phẩm về.

Vấn đề cần được giải quyết với công đoạn trên là làm thế nào để có thể giải được captcha, cho đến hiện tại vẫn chưa có cách giải nào thành công cả, do đó script trên chỉ chạy tốt nhất với trường hợp captcha không xuất hiện trong quá trình download sản phẩm

File script



Thành quả sau khi chạy file script với



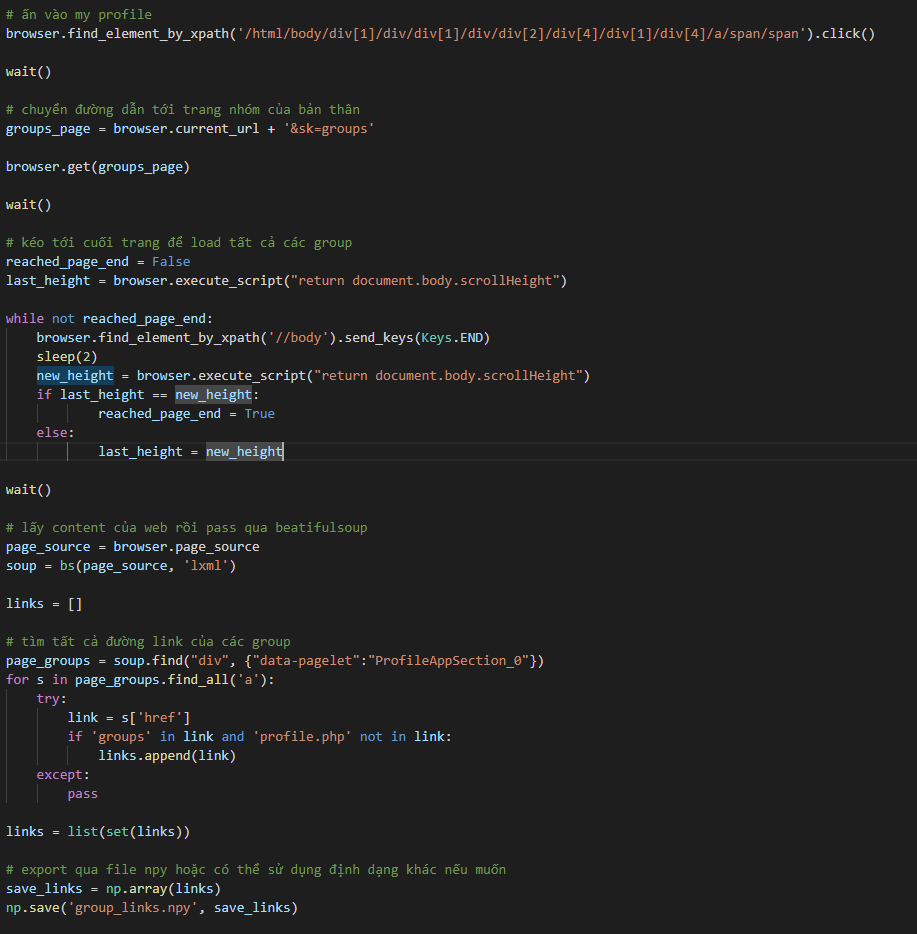
Facebook project:

Project Facebook hiện sử dụng selenium để giả lập thao tác kết hợp cùng với Beautifulsoup để có thể tìm được những nút ấn mong muốn và giả lập thao tác người dùng. Facebook có chính sách chặn bot rất mạnh nên việc sử dụng selenium tuy chậm nhưng sẽ được đảm bảo hơn nếu biết cài đặt các quãng thời gian nghỉ một cách hợp lý để tránh bị nhận dạng là bot.

Tự động lấy link tất cả group mà account đã gia nhập:

Chạy function get\_all\_self\_groups trong file facebook.py

File script:

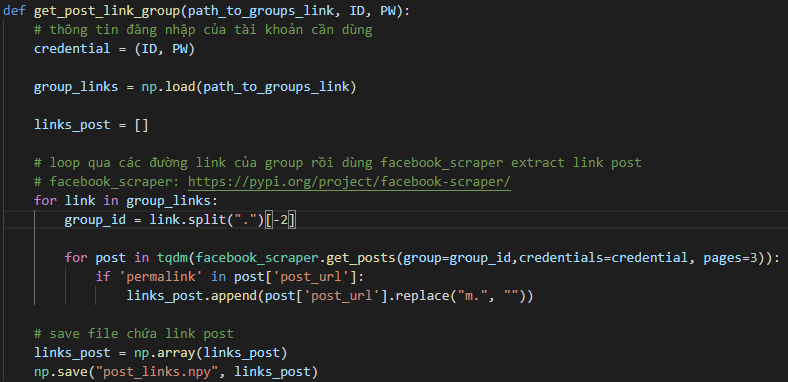


Sau khi chạy script sẽ có file output chứa các đường link dẫn tới group:



Sau khi đã có các link của group, tiếp tục kết hợp với [facebook-scraper](https://pypi.org/project/facebook-scraper/) để lấy được link những bài post trong nhóm:

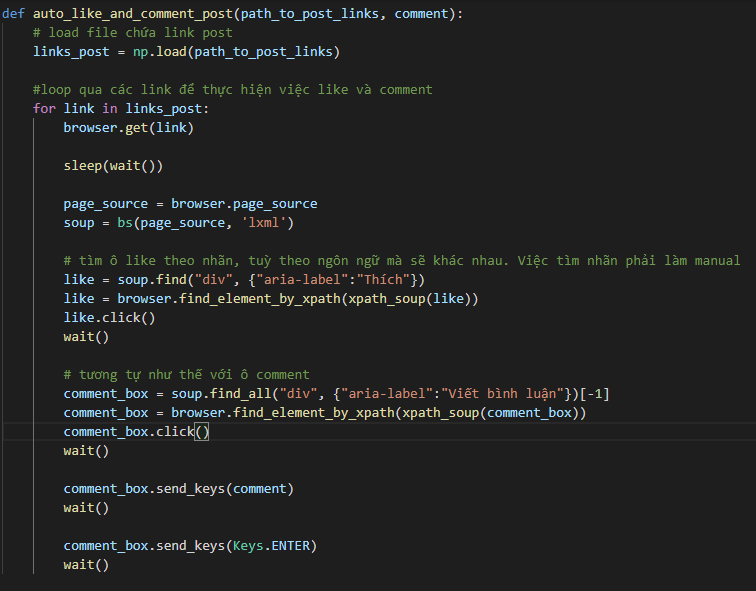
Script để lấy link các bài post trong các group:



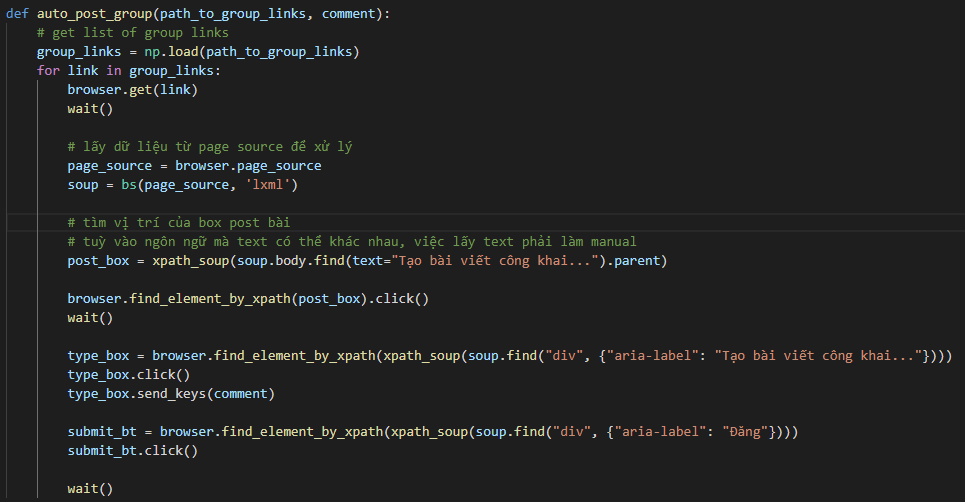
Sau khi chạy xong sẽ có file chứa các đường link của post đó:



Tù việc có những đường link các bài post, với selenium có thể giả lập hành vi như là like với comment trong post đó. Từ đó, có script như sau:



Cũng tương tự như trên, với việc có link của các group thì có thể dùng selenium để có thể giả lập được hành vi đăng bài trong nhóm. Do đó có fucntion:



Với function này, đưa vào file chứa các link của group. Selenium sẽ đi tới từng group và post bài.

Trên là những project đã được làm. Cách làm như trên cũng có thể áp dụng được với những trang mạng xã hội hoặc các trang web khác nhau.