

ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

NGÀNH KHOA HỌC MÁY TÍNH

KHOA HỌC WEB

BÁO CÁO BÀI TẬP 01

28/10/2019

Thông tin sinh viên:

Họ tên: Huỳnh Minh Huấn

MSSV: 1612858

Báo cáo chi tiết:

1. Basic Info Scraping

- Em sử dụng thư viện requests-html.
- Gọi lớp HTMLSession, để get url của website https://scrapingclub.com/exercise/detail_basic/.
- Dùng CSS selector 'div.card-body', với tham số first = True-> lấy ra content về card-body đầu tiên.

```
card_body = req.html.find('div.card-body', first = True)
```

- Trong card_body, sử dụng hàm find tương tự ở trên để tìm thông tin về title, description và price theo các thẻ con trong <div.card-body>

```
product_detail['title'] = card_body.find('h3', first = True).text
product_detail['desc'] = card_body.find('p', first = True).text
product_detail['price'] = card_body.find('h4', first = True).text
```

- Rút được thông tin về sản phẩm

```
for detail in product_detail:
    print(detail + ": " + product_detail.get(detail))
```

```
title: Long-sleeved Jersey Top
price: $12.99
desc: CONSCIOUS. Fitted, long-sleeved top in stretch jersey made from organic cotton with a round neckline. 92% cotton, 3% spandex, 3% rayon, 2% polyester.
```

2. Analyze JSON

- Em sử dụng thư viện requests-html, json và re (regex).
- Gọi lớp HTMLSession, để get url của website https://scrapingclub.com/exercise/detail_json/.
- Dùng hàm find để filter content theo script, sau đó xử lý chuỗi để lấy được chuỗi "var obj = { ..."

```
[5]: scripts = req.html.find('script')
```

```
[6]: for script in scripts:
      if 'obj' in script.text:
          objStr = script.text
          break
      objStr
```

```
[6]: 'var obj = { "title": "Short Sweatshirt", "price": "$24.99", "description": "Short sweatshirt with long sleeves and ribbing at neckline, cuffs, and hem. 57% cotton, 43% polyester. Machine wash cold.", "img_path": "/static/img/96230-C" + ".jpg" }; $(function() { $(".card-title").html(obj.title); $(".card-price").html(obj.price); $(".card-description").html(obj.description); $(".card-img-top").attr("src", obj.img_path); });'
```

- Sử dụng regex + json để lấy được nội dung obj

```
obj_text = re.findall(r'var obj = \{(.*)\}', objStr)
text = r'{' + obj_text[0] + '}'
text_edit = text.replace(' ', '')
#text_edit = text_edit.replace('{', '{').replace('}', '}')
#text_edit
```

```
desc_obj = json.loads(text_edit)
for desc in desc_obj:
    print(desc + ": " + desc_obj[desc])
```

```
title: Short Sweatshirt
price: $24.99
description: Short sweatshirt with long sleeves and ribbing at neckline, cuffs, and hem. 57% cotton, 43% polyester. Machine wash cold.
img_path: /static/img/96230-C.jpg
```

3. Recursively Scraping pages

- Em sử dụng thư viện requests-html, time.
- Gọi lớp HTMLSession, để get url của website https://scrapingclub.com/exercise/list_basic/.

- Requests-html hỗ trợ phương thức `.next()` để lấy link kế tiếp phục vụ cho pagination (link về pagination: <https://requests-html.kennethreitz.org/#pagination>).
- Hàm `get_sub_page` để lấy info của các trang con khi get vào các trang đó.

```
def get_sub_page(url):
    session = HTMLSession()
    req = session.get(url)
    card_body = req.html.find('div.card-body', first = True)
    return {'title': card_body.find('h3', first = True).text,
            'price': card_body.find('h4', first = True).text,
            'desc': card_body.find('p', first = True).text}
```

- Với mỗi card tại url chính sẽ có 1 link đến trang con, em gọi hàm trên để lấy thông tin ở trang con đó.

```
<HTML url='https://scrapingclub.com/exercise/list_basic/'>
https://scrapingclub.com/exercise/list_basic_detail/90008-E/
https://scrapingclub.com/exercise/list_basic_detail/96436-A/
https://scrapingclub.com/exercise/list_basic_detail/93926-B/
https://scrapingclub.com/exercise/list_basic_detail/90882-B/
https://scrapingclub.com/exercise/list_basic_detail/93756-C/
https://scrapingclub.com/exercise/list_basic_detail/93926-C/
https://scrapingclub.com/exercise/list_basic_detail/93756-B/
https://scrapingclub.com/exercise/list_basic_detail/93756-D/
https://scrapingclub.com/exercise/list_basic_detail/96643-A/
https://scrapingclub.com/exercise/detail_basic/
<HTML url='https://scrapingclub.com/exercise/list_basic/?page=2'>
https://scrapingclub.com/exercise/list_basic_detail/94766-A/
https://scrapingclub.com/exercise/list_basic_detail/91696-C/
```

- Kết quả: một vài kết quả

```
title: Short Dress
price: $24.99
desc: Short dress in woven fabric. Round neckline and opening at back of neck with a button. Yoke at back with c
onced pleats, long sleeves, and narrow cuffs with ties. Side pockets. 100% polyester. Machine wash cold.

title: Patterned Slacks
price: $29.99
desc: Ankle-length slacks in patterned stretch cotton satin. Regular waist with concealed hook-and-eye fastener
and zip fly. Side pockets and tapered legs with slits at hems. 61% cotton, 36% polyester, 3% spandex. Machine wa
sh...
```

4. Mimicking Ajax requests

- Mục tiêu bài này là extract được link `ajax_detail` (<https://scrapingclub.com/exercise/ajaxdetail/>) để truy cập vào get object json
- url gốc là https://scrapingclub.com/exercise/detail_ajax/.
- Em sử dụng thư viện `requests-html`, `re` và `json`.
- Dùng hàm phương thức `find()` của `HTMLSession` để tìm script chứa link cần extract

```
scripts = req.html.find('script')
```

- Sử dụng regex để extract link

```
break
ajax_detail = re.findall(r'url: "(.*?)"', ajax)

new_url = domain + ajax_detail[0]
print(new_url)
req = session.get(new_url)
```

- Get obj json bằng phương thức `.json()`

```
desc = req.json()
print(desc)

{'img_path': '/static/img/96113-C.jpg', 'price': '$19.99', 'description': 'Fitted dress in jersey with long, str
aight sleeves. Unlined. 72% polyester, 23% rayon. 5% spandex. Machine wash...', 'title': 'Jersey Dress'}
```

5. Inspect HTTP request

- Tương tự exercise 04.
- Cần thêm headers (referer và 'x-requested-with') mới có thể get được nội dung link extract từ ajax script https://scrapingclub.com/exercise/ajaxdetail_header/

```
3]: req = session.get(url)
headers = session.headers
headers['referer'] = url
headers['x-requested-with'] = 'XMLHttpRequest'
headers

3]: {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/603.3.8 (KHTML, like Gecko) Version/10.1.2 Safari/603.3.8', 'Accept-Encoding': 'gzip, deflate', 'Accept': '/*/*', 'Connection': 'keep-alive', 'referer': 'https://scrapingclub.com/exercise/detail_header/', 'x-requested-with': 'XMLHttpRequest'}
```

- url gốc https://scrapingclub.com/exercise/detail_header/
- extract link tương tự exercise 04, phương thức .find() filter theo tag script + regex

```
ajax = ''
for script in scripts:
    if script.text.find('ajax') != -1:
        ajax = script.text
        break
ajax_detail = re.findall(r'url: "(.*)"', ajax)

new_url = domain + ajax_detail[0]
print(new_url)
req = session.get(new_url, headers = headers)

https://scrapingclub.com/exercise/ajaxdetail_header/
```

- kết quả:

```
desc = req.json()
print(desc)

{'img_path': '/static/img/00959-A.jpg', 'price': '$24.99', 'description': 'Blouse in airy, crinkled fabric with a printed pattern. Small stand-up collar, concealed buttons at front, and flounces at front. Long sleeves with buttons at cuffs. Rounded hem. 100% polyester. Machine wash cold.', 'title': 'Crinkled Flounced Blouse'}
```

6. Scraping Infinite Scrolling Pages (Ajax)

- Em sử dụng thư viện requests-html và time.
- Tương tự câu 3, việc scrolling (pagination) được requests-html hỗ trợ.
- Hàm get_sub_page để lấy thông tin sản phẩm của card.

```
[3]: def get_sub_page(url):
    session = HTMLSession()
    req = session.get(url)
    card_body = req.html.find('div.card-body', first = True)
    return {'title': card_body.find('h3', first = True).text,
            'price': card_body.find('h4', first = True).text,
            'desc': card_body.find('p', first = True).text}
```

- Filter theo thẻ card bằng phương thức .find để lấy ra các card -> extract link chi tiết.

```
cards_html = req.find('div.card')
for card in cards_html:
    card_link = 'https://scrapingclub.com' + card.find('a')[0].attrs['href']
    print(card_link)
```

- Kết quả

```
[5]: for card in cards:
      for key in card:
          print(key + ": " + card[key])
      print()

title:
price:
desc: Short dress in woven fabric. Round neckline and opening at back of neck with a button. Yoke at back with c
onced pleats, long sleeves, and narrow cuffs with ties. Side pockets. 100% polyester. Machine wash cold.

title:
price:
desc: Ankle-length slacks in patterned stretch cotton satin. Regular waist with concealed hook-and-eye fastener
and zip fly. Side pockets and tapered legs with slits at hems. 61% cotton, 36% polyester, 3% spandex. Machine wa
sh...
```

7. Find gold in cookie

- Có thể sử dụng thư viện requests hay requests-html, thư viện bs4 để tiện lấy content, thư viện re (regex).
- Cần lấy headers và token. Token được extract từ headers (response) 'set-cookie'.

`r = s.get(url)` : get url để nhận được response -> extract token

`headers['user-agent'] = s.headers['user-agent']`
`headers['x-requested-with'] = 'XMLHttpRequest'` : cần headers với 2 nội dung này 'user-agent' và 'x-requested-with'.

```
soup = BeautifulSoup(r.content, 'html5lib')
token_cookies = re.findall(r'token=(.*?);', r.headers['set-cookie'])
new_url = domain + '/exercise/ajaxdetail_cookie/?token=' + token_cookies[0] : extract
```

token -> tạo link url để get

```
r = s.get(new_url, headers = headers) #, headers = r.headers
print(r)
```

- : get url để nhận response -> chứa content thông tin hàng.

```
KeyValue({'User-Agent': 'python-requests/2.22.0', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*', 'Connecti
on': 'keep-alive'})
https://scrapingclub.com/exercise/ajaxdetail_cookie/?token=17L09YYKCG
<Response [200]>
```

- Kết quả:

```
[5]: desc = r.json()
desc

[5]: {'img_path': '/static/img/94323-B.jpg',
      'price': '$29.99',
      'description': 'Short bib overall dress in twill. Shoulder straps tied together with metal eyelets at top. Ches
t pocket, front pockets, and back pockets. Unlined. 58% cotton, 42% lyocell. Machine wash cold.',
      'title': 'Bib Overall Dress'}
```

8. Login form

- Em sử dụng thư viện requests và bs4
- Để điền form đăng nhập, ta cần headers của requests và data.

```
headers = {
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0
    'referer': 'https://scrapingclub.com/exercise/basic_login/'
}
data = {
    'csrfmiddlewaretoken': '',
    'name': 'scrapingclub',
    'password': 'scrapingclub'
}
```

- Trong data thì 'csrfmiddlewaretoken' sẽ unique với mỗi session

```
<form method="post" novalidate> == $0
<input type="hidden" name="csrfmiddlewaretoken" value=
"Q9tfinSRxgOo0WGFvTqyHhf3VKDWACdfvzzeJ7ku7w6qYXjjL83NNLTH9IbPmma">
▶<div id="div_id_name" class="form-group">...</div>
▶<div id="div_id_password" class="form-group">...</div>
<button type="submit" class="btn btn-primary">Log in</button>
```

- Gọi phương thức get của session trong requests để lấy giá trị 'csrfmiddlewaretoken' trong session đó

```
soup = BeautifulSoup(r.content, 'html5lib')
csrf_token = soup.find('input', attrs = {'name': 'csrfmiddlewaretoken'})['value']
data['csrfmiddlewaretoken'] = csrf_token
```

- Gọi phương thức post của session để send data form thực hiện log in

```
r = s.post(url, data = data, headers = headers)
print(r)
```

<Response [200]>

- Kết quả:

```
[4]: soup = BeautifulSoup(r.content, 'html5lib')
      soup.find('p').text
```

```
[4]: 'You have successfully login in, Congratulations'
```

9. Solve Captcha

- Tương tự exercise 08, cần thêm 2 nội dung truyền vào data form để send post lên là captcha_0 và captcha_1
- Cách để lấy captcha_0: sử dụng session.get để lấy nội dung html. Dùng BeautifulSoup để convert response.content. find theo 'input' với attrs = {'name': 'captcha_0'} và get value.

```
captcha_0 = soup.find('input', attrs = {'name': 'captcha_0'})['value']
data['captcha_0'] = captcha_0
```

- Cách lấy captcha_1:
 - o Lấy link image captcha:

```
img_url = "https://scrapingclub.com" + soup.find('img', attrs = {'alt': 'captcha'})['src']
print(img_url)
```

- o Sử dụng API py9kw để nhận được mã captcha

```
print(service.getcredits())
service.uploadcaptcha(imagedata=image_data.content, maxtimeout=60, prio=8)
service.sleep()
service.getresult()
print(service.rslt)
result = service.rslt
```

- Với captcha_1 và captcha_0 có được, thêm vào data['captcha_0'] và data['captcha_1'], gửi POST lên tương tự như exercise 08.

Kết quả:

```
[py9kw] Captcha solved! String: 'rwgh'
('rwgh', True)
('rwgh', True)
RWGH
<Response [200]>
```

```
: success_msg = r.html.find('p', first = True).text
print(success_msg)
```

You have successfully login in, Congratulations

10. Decode minified javascript

a) Cách 1: Sử dụng Selenium để load website

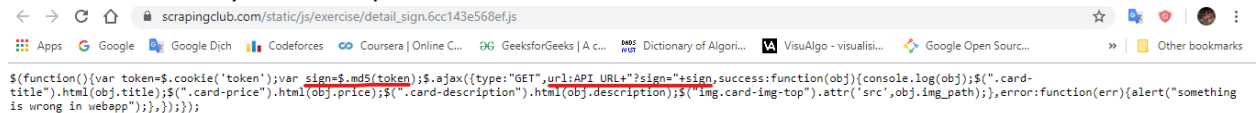
- Cài đặt selenium theo link <https://selenium-python.readthedocs.io/installation.html>.
- Mở driver chrome bằng webdriver selenium
- Get link https://scrapingclub.com/exercise/detail_sign/ để browser load website.
- Sử dụng `find_element_by_css_selector` để lấy thông tin.

```
browser.get('https://scrapingclub.com/exercise/detail_sign/')
|
title = browser.find_element_by_css_selector('h4.card-title').text
price = browser.find_element_by_css_selector('h4.card-price').text
description = browser.find_element_by_css_selector('p.card-description').text
```

- Kết quả:
Loose-knit Sweater
\$17.99
Soft, loose-knit sweater with a V-neck, long raglan sleeves, and roll edges at cuffs and hem. Longer at back. 100% acrylic. Machine wash warm.

b) Cách 2:

- Sử dụng thư viện requests-html + inspect để lấy thông tin API_URL trong file js.
- Url và token cần lấy sau khi inspect để xem



```
$(function(){var token=$.cookie('token');var sign=$.md5(token);$.ajax({type:"GET",url:API_URL+"?sign="+sign,success:function(obj){console.log(obj);$(".card-title").html(obj.title);$(".card-price").html(obj.price);$(".card-description").html(obj.description);$(".img.card-img-top").attr('src',obj.img_path);},error:function(err){alert("something is wrong in webapp");}});});
```

- Cần hash md5 token -> sign. Sau đó new link sẽ có dạng:
https://scrapingclub.com/exercise/ajaxdetail_sign/?sign={sign} -> với sign là kết quả hash bằng md5.
- Thực hiện phương thức get với https://scrapingclub.com/exercise/detail_sign/ để lấy session request headers. (Thêm vào headers 'x-requested-with': 'XMLHttpRequest').
- Token được lấy bằng cách sử dụng regex đối với response headers, sau đó sign sẽ lưu giá trị hash md5 của token.

```
token = re.findall(r'token=(.??);',r.headers['Set-Cookie'])[0]
sign = md5(token.encode()).hexdigest()
```

- Lấy API_URL bằng cách sử dụng regex đối với tag script chứa 'API_URL'

```

API_URL = ''
script_tags = r.html.find('script')
for script_tag in script_tags:
    tmp = re.findall(r'API_URL = "(.*?)"', script_tag.text)
    if (len(tmp) > 0):
        API_URL = tmp[0]
        break

```

- Get new_url với headers ở trên

```

new_url = domain + API_URL + '?sign=' + sign
print(new_url, end = '\n\n')
r = session.get(new_url, headers = headers)

```

- Kết quả

```

information = r.json()
for detail in information:
    print(information[detail])

```

https://scrapingclub.com//exercise/ajaxdetail_sign/?sign=c0a14b797e0a8b46859ce2c1ec1adb7

/static/img/71342-J.jpg

\$17.99

Soft, loose-knit sweater with a V-neck, long raglan sleeves, and roll edges at cuffs and hem. Longer at back. 100% acrylic. Machine wash warm.

Loose-knit Sweater

Tham Khảo:

Documentation:

- Requests: <https://requests.kennethreitz.org/en/master/user/quickstart/#custom-headers>.
- Requests-html: <https://requests-html.kennethreitz.org/>.

Books:

- [https://marcell.memoryoftheworld.org/Richard%20Lawson/Web%20Scraping%20With%20Python%20\(2685\)/Web%20Scraping%20With%20Python%20-%20Richard%20Lawson.pdf](https://marcell.memoryoftheworld.org/Richard%20Lawson/Web%20Scraping%20With%20Python%20(2685)/Web%20Scraping%20With%20Python%20-%20Richard%20Lawson.pdf).

API:

- <https://github.com/JanHelbling/py9kw>.