程式設計(二)

selenium fb 爬蟲實作

Ming-Hung Wang 王銘宏

tonymhwang@cs.ccu.edu.tw

Department of Computer Science and Information Engineering National Chung Cheng University

Spring Semester, 2022

本章目錄

- 1. 介紹
- 2. 前置作業
- 3. 函式介紹
- 4. facebook 爬蟲
- 5. 總結

介紹

什麼是 selenium

- selenium 可以開啟瀏覽器,並根據使用者編寫的程式碼,模擬對 瀏覽器的操作(輸入、點擊……等)。
- selenium 經常搭配 beautifulsoup 使用,以解析進行操作後的網頁 資料。

前置作業

前置作業

安裝 selenium、beautifulsoup。

pip install selenium pip install bs4

前置作業

前置作業

下載 webdriver。

ChromeDriver

前置作業

前置作業

查看 Chrome 版本,並下載對應版本的 webdriver。

| 關於 Chrome | |
|---|---|
| Google Chrome | |
| ♥ Chrome 目前是最新版本 版本 99.0.4844.84 正式版本) (64 位元) | |
| 前往 Chrome 說明賈茵 | |
| 回報問題 | Ø |

selenium 函式介紹

- driver = webdriver.Chrome(path)
 開啟 webdriver。
- driver.get(url) 操作瀏覽器進入網頁。

selenium 函式介紹

- ele = driver.find_element_by_class_name(value)
 尋找第一個符合 class name = value 的網頁元素並回傳。
- eles = driver.find_elements_by_class_name(value)
 尋找所有符合 class name = value 的網頁元素並回傳。

selenium 尋找網頁元素函式

- ele = driver.find_element_by_id(value)
- ele = driver.find_element_by_name(value)
- ele = driver.find_element_by_xpath(value)

selenium 函式介紹

- ele.send_keys(value) 將 value 填入 ele 中。
- ele.click() 對 ele 進行點擊。

beautifulsoup 函式介紹

 soup = BeautifulSoup(text, parser) 解析網頁內容。

beautifulsoup 函式介紹

- ele = soup.find(type, {key : value})
 回傳第一個符合 key = value 的 type 網頁元素。
 ex. ele = soup.find('div', {'id' : 'myid'})
- ele = soup.find_all(type, {key : value})
 回傳所有符合 key = value 的 type 網頁元素。

開啟 webdriver

首先設定瀏覽器的選項,將放大視窗加入選項,然後開啟 webdriver。

```
options = webdriver.ChromeOptions()
options.add_argument("--start-maximized")

driver = webdriver.Chrome('Crawler/chromedriver.exe', options = options)
```

登入

首先查看登入頁面的 html 碼。



登入

尋找輸入帳號密碼的網頁元素。

```
**In claim** Art and better on Application and the Application and
```

登入

進入 facebook 首頁。

```
driver.get('https://www.facebook.com/')
time.sleep(1)
```

登入

找到輸入帳號密碼的網頁元素並填入帳號密碼。

```
context = driver.find_element_by_id('email')
context.send_keys('peter123@gmail.com')
context = driver.find_element_by_id('pass')
context.send_keys('ptr870229')
```

登入

尋找登入按鈕的網頁元素。

```
* (form class*_0vtf* data-testio="royal login_form" action="/login/privacy_matation_token-ey/daxNiiipucCj(enheS)+b100eidIliput(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt(DipUt
      (input type="hidden" name="Sazoest" value="2940" autocomplete="off")
      (input type="hidden" name="lsd" value="AVgpyE4e148" autocomplete="off")
   ¥ 16512
             (input type="text" class="inputtext _55ri _61yy" name="email" id="email" data-textid="roysl_email" placeholder="椰子製件处於或手類材理" satofocus="1" aria-label="椰子製件处数或手類材理")
          4 felio
        * (div class="_6lus")
           *(div class="_6luv _55rl _1k0t" id="passContainer")
                    (input type-"password" class-"inputtent _Strl _Gluy _Groi" name-"pass" id-"pass" data-testid-"royal_pass" placeholder-"電視" aria-label-"電視")
               * (div class="_91s7 hidden_elen" id="u.0.b.of")_(/div)
               C/4512
          c/divo
      (input type="hidden" autocomplete="off" name="login_source" value="comet_headerless_login")
        (input type="hidden" autocomplete="off" name="next" value)
     " (div class="_6ltg")
      C/dSv>
     odiy class="_Bicz">4/diy> flex
   * odly class=" 6lte">_c/dlv>
```

登入

找到登入按鈕的網頁元素並點擊。

```
commit = driver.find_element_by_name('login')
commit.click()
time.sleep(1)
```

登入

登入後看到以下通知,會阻止我們對瀏覽器進行操作。

| ww | w.facebook.co | om要求下列權同 | 限: |
|----|---------------|----------|------------|
| Ņ | 顯示通知 | | |
| | | 允許 | 幸付鎖 |

登入

在瀏覽器的選項添加以下設定,可以封鎖通知。

```
prefs = {
    'profile.default_content_setting_values' : {
    'notifications': 2
    }
}
options.add_experimental_option('prefs', prefs)
```

爬取粉絲專頁貼文

首先進入粉絲專頁頁面。

由於 facebook 是動態網頁,往下滾動才會顯示更多貼文,所以我們利 用 javescript 指令來滾動瀏覽器。

```
driver.get('https://www.facebook.com/appledaily.tw')
driver.execute_script('window.scrollTo(0, document.body.scrollHeight)')
time.sleep(1)
```

爬取粉絲專頁貼文

找到目前所有貼文。

posts = driver.find_elements_by_class_name('du4w351b k4urcfbm 19j0dhe7 sjgh65i0'.replace(' ', '.'))

爬取粉絲專頁貼文

如果要讀取貼文全部內容的話,我們需要點擊顯示更多。



爬取粉絲專頁貼文

找到顯示更多的按鈕,這裡有多個網頁元素共用同個 class name,所以利用網頁元素的文字來找到我們想要點擊的按鈕。

```
<div class="oajrlxb2 g5ia77u1 qu0x051f esr5mh6w e9989ue4 r7d6kgcz rq0escxv nhd2j8a9
zcic4wl gpro0wi8 oo9gr5id lrazzd5p" role="button" tabindex="0">顯示更多</div> == $0
```

```
for post in posts:

btms = post.find_elements_by_class_name('oajrlxb2 g5ia77u1 qu0x051f esr5mh6w e9989ue4

for btm in btms:

if btm.text == '顯示更多':

btm.click()

break
```

爬取粉絲專頁貼文

click() 會自動將瀏覽器滾動可以點擊的位置。 但是 facebook 上方的區塊會使點擊出錯,如果欲點擊的網頁元素在 瀏覽器當前位置的上方,可能出現無法點擊的情況。



爬取粉絲專頁貼文

可以利用 javescript 指令來滾動瀏覽器到頂部,讓所有需要點擊的網 頁元素都在當前位置的下方。

driver.execute_script('document.documentElement.scrollTop=0')
time.sleep(1)

爬取粉絲專頁貼文

如果想要取得貼文的 ID ,需要從發布時間的連結取得。 但是連結裡面的 href 是空的,需要鼠標移動到該元素上面才會產生 href。

爬取粉絲專頁貼文

首先找到需要連結的元素,然後利用 ActionChains 模擬鼠標移動到該元素上。

link = post.find_element_by_class_name('oajrlxb2 g5ia77u1
ActionChains(driver).move_to_element(link).perform()

爬取粉絲專頁貼文

回去查看網頁 html 碼,發現 href 的欄位出現了貼文連結。

*spec (case*telponet selicité selijées skilón) tidlem kéndőr dilétőy pasétő desertő edesert annol mehenén mátelet negleté jayosátt.

'se mériabetínjel' (case*anjal) dilánd administrational szembe edőbbet érőséger rádsacs nyálásá mátele újását a veget cement syraman közyás jölyjas
szoram filipón liziásá jazlamá produk köznek réferintel/inan/desende, norásoltását, jazjasátalását azadát azadát azadát a közdetőséletényártattal, majonoltón *sindelintelesendőségen előségendőségen előségendősé

爬取粉絲專頁貼文

利用 BeautifulSoup() 來解析網頁資料。 找到目前所有貼文。

```
soup = BeautifulSoup(driver.page_source, 'lxml')

posts = soup.find_all('div', {'class' : 'du4w35lb k4urcfbm 19j0dhe7 sjgh65i0'})
```

爬取粉絲專頁貼文

由於每個貼文不一定都具備內容、按讚人數、留言人數、分享人數, 所以利用 try except 來確保不會有 bug。

```
try:
    message = post.find('div', ('class': 'ecoMobert hvdrvnfc ibqu7lf3 datilw0a'))
    message = re.sub('\n', '', message.text)
    except:
    message = ''

try:
    like_cnt = post.find('span', ('class': 'pcp9lagn'))
    like_cnt = re.findall('[0-9]', like_cnt.text)[0]
    except:
    like_cnt = 0

try:
    comment_cnt = post.find('span', ('class': 'd2edcug0 hpfwergz qv66sadb cletSuccomment_cnt = re.findall('(-*))HEE', comment_cnt.text)[0]
    except:
    comment_cnt = 0

try:
    share_cnt = post.find('span', ('class': 'd2edcug0 hpfwergz qv66sadb cletSuccomment_cnt.text)[0]
    share_cnt = re.findall('(-*))XGE', share_cnt.text)[0]
    except:
    share_cnt = re.findall('(-*))XGE', share_cnt.text)[0]
```

爬取粉絲專頁貼文

成功。

總結

selenium requests 比較

- selenium
- 模擬瀏覽器操作,較為方便理解。
- 會得到所有網頁資料,速度較慢。

- requests
- 送出請求要求資料,參數較多,不易理解。
- 可以只要求目標資料,速度較快。

END