

windy.com Webscraper

嘗試下載 windy.com 的氣象預報

- 無法直接解析網頁的 html source code 來獲得資料
 - 例如「港灣環境資訊網」的做法
- 也無法透過 web api 下載資料
 - 沒有公開的 api，尚不清楚私人付費 api 的操作
- 最後只能透過 selenium 來模擬「人操作瀏覽器」
 - 第三方模組 windy-weather-crawler (需要修改才能正常)
 - 參考 windy-weather-crawler 的程式碼，並實現全新的爬蟲
 - 下載並修改 windy.com 的預報資料，最後生成 excel 檔案
 - 直接輸入 gps 座標來準確選定地點
 - 透過 docker 移植程式到群暉主機跑，尚不穩定

Join the official 2020 Python Developers Survey: [Start the survey!](#)

windy-weather-crawler 1.1

```
pip install windy-weather-crawler
```

```
test_windy-weather-crawler.py - D:\Python_projects\
File Edit Format Run Options Window Help
from windy_crawler.run import windy_crawler
windy_crawler()
```

Latest version

Released: Feb 1, 2020

感謝它示範如何抓取 windy.com 的資料

A web crawler that takes input as place gives the weather forecast for 5 days

Navigation

- Project description
- Release history
- Download files

Project links

Project description

Windy Web Crawler

This application is built with scrapy and selenium

It takes places name as input to give weather forecast in commandline for 5 upcoming days

The package can be installed using pip by following command (Python >=3.6 is required for this project)

```
pip3 install windy-weather-crawler
```

Usage :

1. Create a python file with any name
2. write the following code in there
3. Save the file and run it

```
from windy_crawler.run import windy_crawler windy_crawler()
```

Input :

1. You need to enter the firefox binary file path on-request

```
Welcome to Windy Web Crawler Please enter your full path to firefox binary in your system(default: /usr/bin/firefox) : 2. Then type the place
```

```
Now select the place for weather forecast(ex. chennai, hyderabad, delhi etc...) : 3. You will get some suggestions regarding locality in that place
```

使用和操作

windy-weather-crawler 執行過程

C:\Windows\system32\cmd.exe

```
(venv_windy2) D:\Python_projects\windy>python test_windy-weather-crawler.py
Welcome to Windy Web Crawler
Please enter your full path to firefox binary in your system(default: /usr/bin/firefox) :
c:/InstalledApps/Mozilla Firefox/firefox.exe
Path taken: c:/InstalledApps/Mozilla Firefox/firefox.exe
```

1

```
Now select the place for weather forecast(ex. chennai, hyderabad, delhi etc... :
kaohsiung
2020-10-13 10:23:40 [py.warnings] WARNING: D:\venv_windy2\lib\site-packages\scrapy\spidermiddlewares\offsite.py:65: URLW
arning: allowed_domains accepts only domains, not URLs. Ignoring URL entry https://www.windy.com in allowed_domains.
warnings.warn(message, URLWarning)
```

2

```
| Place Name
+-----+
0 | Kaohsiung Sanmin District
1 | Kaohsiung International Airport RCKH
2 | Kaohsiung Lingya District
3 | Kaohsiung
4 | Kaohsiung Sanmin District
5 | Hafen Kaohsiung Qianzhen District
6 | Kaohsiung International Airport Xiaogang District
7 | Kaohsiung Arena Gushan District
8 | Kaohsiung Museum of History Yancheng District
9 | Kaohsiung Museum of Fine Arts Gushan District
Please Select the Index from above table: 3
```

3

```
2020-10-13 10:23:52 [selenium.webdriver.remote.remote_connection] DEBUG: Finished Request
#####Weather Forecast for 5 days#####
=>星期二 13
Time of Day | Temp(C) | Wind Speed(Kt) |
+-----+-----+-----+
2 | 22° | 1 |
5 | 21° | 1 |
8 | 25° | 1 |
11 | 27° | 3 |
14 | 28° | 0 |
17 | 26° | 1 |
20 | 24° | 2 |
23 | 23° | 2 |
=>星期三 14
Time of Day | Temp(C) | Wind Speed(Kt) |
+-----+-----+-----+
2 | 23° | 1 |
5 | 22° | 2 |
```

4

修改 windy-weather-crawler 後才能動 (windy.com 可能改變了網頁寫法)

forecast_spider.py - D:\venv_windy2\Lib\site-packages\windy_crawler\spiders\forecast_spider.py (3.8.5)

File Edit Format Run Options Window Help

```
46         if airport:
47             max_temp, min_temp = str(els.text).split('\n')
48             temps_list.append(min_temp + '-' + max_temp)
49         else:
50             temps_list.append(els.text)
51     for els in winds:
52         if airport:
53             trash, min_speed, max_speed = str(els.text).split('\n')
54             winds_list.append(min_speed + '-' + max_speed)
55         else:
56             winds_list.append(els.text)
57     for els in times:
58         times_list.append(els.text)
59     print('#####Weather Forecast for 5 days#####')
60     for day in days_list:
61         print('=>' + day)
62         #ind = times_list.index('OAM')
63         #if ind == 0:
64         #    ind = 8
65         ind = 8 # 上面3行拿掉，直接將ind設8
66         current_times, times_list = times_list[:ind], times_list[ind:]
67         current_temps, temps_list = temps_list[:ind], temps_list[ind:]
68         current_winds, winds_list = winds_list[:ind], winds_list[ind:]
69         display_list = []
70         for i in range(0, len(current_times)):
71             display_list.append([current_times[i], current_temps[i], current_winds[i]])
```

OAM 這值在目前的 windy.com 不存在，造成 library 跑不起來 (若不改程式的話)

用 xpath 鎖定資料 (以溫度為例)

| | 星期一 12 | | | | | | | 星期二 13 | | | | | | | | |
|--------|--------|-----|-----|-----|-----|-----|-----|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| 小時 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 |
| 溫度 °C | 23° | 22° | 26° | 28° | 28° | 26° | 24° | 24° | 23° | 22° | 25° | 27° | 27° | 25° | 24° | 24° |
| 降雨量 mm | 0.5 | 1 | | | 0.5 | 0.8 | 0.6 | 0.7 | 0.2 | 0.2 | | | 0.6 | 0.8 | 0.4 | |
| 風速 kt | 1 | 2 | 0 | 2 | 3 | 3 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| 陣風 kt | 4 | 5 | 6 | 10 | 11 | 10 | 8 | 7 | 5 | 5 | 6 | 10 | 11 | 9 | 5 | 4 |
| 風向 | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← |

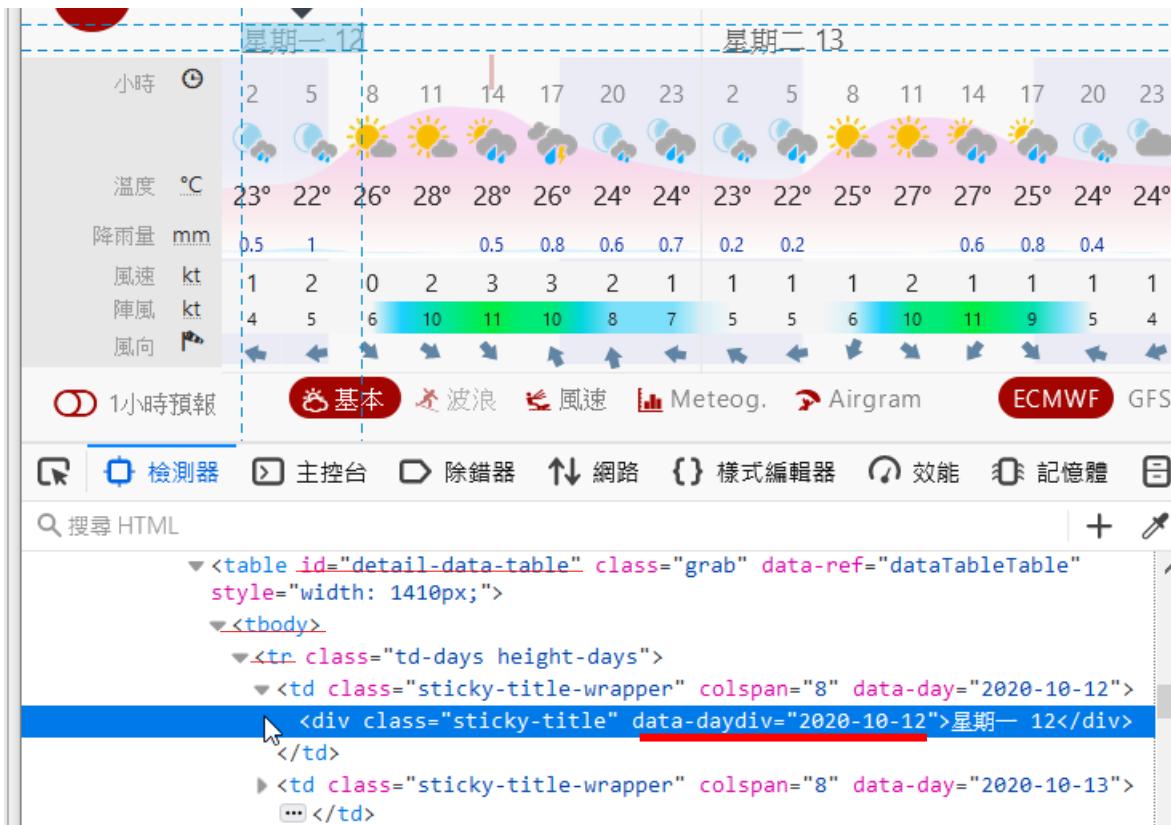
```
<div class="progress-line" style="width: 1410px;">...</div>
<table id="detail-data-table" class="grab" data-ref="dataTableTable" style="width: 1410px;">
  <tbody>
    1 <tr class="td-days height-days">...</tr>
    2 <tr class="td-hour height-hour d-display-table">...</tr>
    3 <tr class="td-icon height-icon d-display-table">...</tr>
    4 <tr class="td-temp height-temp d-display-table">
      <td>23°</td>
      <td>22°</td>
```

瀏覽器的「官方」畫面

瀏覽器的開發者模式下的視窗

temps_xpath='//*[@id="detail-data-table"]/tbody/tr[4]/*'

擷取瀏覽器沒顯示的資料



```
days_xpath='//*[@id="detail-data-table"]/tbody/tr[1]/*'  
days_and_dates = browser.find_elements_by_xpath(xpath=days_xpath)  
...  
for els in days_and_dates:  
    data = els.find_element_by_tag_name('div')  
    print(data.get_attribute('data-daydiv'))
```

```

1 from selenium import webdriver
2 from selenium.webdriver.firefox.firefox_binary import FirefoxBinary
3 from selenium.webdriver.common.keys import Keys
4
5 from selenium.webdriver.firefox.options import Options
6
7 import time
8
9 from json_excel_converter import Converter
10 from json_excel_converter.xlsx import Writer
11 import json
12
13 URL_HOME = "https://www.windy.com"
14
15 #options = Options()
16 #options.add_argument('-headless')
17 binary = FirefoxBinary(r'C:\InstalledApps\Mozilla Firefox\firefox.exe')
18 browser = webdriver.Firefox(firefox_binary=binary)
19 #browser = webdriver.Firefox(firefox_binary=binary, options=options) #headless
20 browser.get(URL_HOME)
21
22 search = browser.find_element_by_xpath(xpath='//*[@id="q"]')
23 search.send_keys('kaohsiung')
24 print('Sending Search Data and retrieving suggestions, Please wait...')
25 time.sleep(2)
26 results = browser.find_elements_by_xpath(xpath='//*[@id="search"]/div[4]/div[1]/*')
27
28 for i, place in enumerate(results):
29     print(i, place.text)
30
31 # select index
32 search.send_keys(Keys.ARROW_DOWN)
33 search.send_keys(Keys.ARROW_DOWN)
34 search.send_keys(Keys.ARROW_DOWN)
35 search.send_keys(Keys.ARROW_DOWN)
36 time.sleep(2)
37 search.send_keys(Keys.ENTER)
38 time.sleep(2)
39
40 days_xpath='//*[@id="detail-data-table"]/tbody/tr[1]/*'
41 hours_xpath='//*[@id="detail-data-table"]/tbody/tr[2]/*'
42 temps_xpath='//*[@id="detail-data-table"]/tbody/tr[4]/*'
43 winds_xpath='//*[@id="detail-data-table"]/tbody/tr[6]/*'
44
45 days_and_dates = browser.find_elements_by_xpath(xpath=days_xpath)
46 hours = browser.find_elements_by_xpath(xpath=hours_xpath)
47 temperatures = browser.find_elements_by_xpath(xpath=temps_xpath)
48 winds = browser.find_elements_by_xpath(xpath=winds_xpath)
49
50 rains_xpath='//*[@id="detail-data-table"]/tbody/tr[5]/*'
51 rains = browser.find_elements_by_xpath(xpath=rains_xpath)

```

參考 windy-weather-crawler
後全新改寫程式

尋找輸入框並填入地點

模擬按下「方向鍵」四次
和最後按下「Enter」

測試改變選單

產生 json

產生 excel

```

52 days_list, hours_list, temps_list, winds_list = [], [], [], []
53 dates_list, rains_list = [], []
54
55
56 for els in days_and_dates:
57     data = els.find_element_by_tag_name('div')
58     print('data:', data.text)
59     days_list.append(data.text)
60     print(data.get_attribute('data-daydiv'))
61     dates_list.append(data.get_attribute('data-daydiv'))
62 for els in temperatures:
63     temps_list.append(els.text)
64 for els in winds:
65     winds_list.append(els.text)
66 for els in hours:
67     hours_list.append(els.text)
68
69 for els in rains:
70     rains_list.append(els.text)
71
72 five_days = []
73 last_hour = 0
74 date_index = 0
75 for i, hour in enumerate(hours_list):
76     item = {}
77
78     # combine 'date' and 'hour' to generate 'time'
79     this_hour = int(hour)
80     if this_hour < last_hour:
81         date_index += 1
82     this_date = dates_list[date_index] + '-{:02d}'.format(this_hour)
83     last_hour = this_hour
84     item['time'] = this_date
85     item['day'] = days_list[date_index]
86
87     item['hour'] = hour
88     item['temp'] = temps_list[i]
89     item['rain'] = rains_list[i]
90     item['wind'] = winds_list[i]
91
92     five_days.append(item)
93
94 #meteog_xpath = '/html/body/div[4]/div[2]/div[6]/div[2]/div[4]' # 語法較死，會掛
95 meteog_xpath='//*[@id="detail-box"]/div[2]/div[4]' # 語法較不特定
96 meteog = browser.find_element_by_xpath(xpath=meteog_xpath)
97 meteog.click()
98
99 with open('test4.json', mode='wt', encoding='utf-8') as f:
100     #f.write(json.dumps(five_days, indent=2)) # generate \uxxxx instead of 中文
101     json.dump(five_days, fp=f, indent=2, ensure_ascii=False)
102
103 filename_excel = 'test4.xlsx'
104 conv = Converter()
105 conv.convert(five_days, Writer(file=filename_excel))
106
107 #browser.close()

```

因為網頁儲存時間的原始格式
不適合表格的產生，所以新增

整理成表格，方便
程式進一步處理

| | 全... | | A | B | C | D | E | F |
|----|---------------|-----|-----|------|------|------|------|---|
| 1 | tin | | day | hour | temp | rain | wind | |
| 2 | 2010-10-02 | 星期一 | 12 | 2 | 23° | 0.5 | 1 | |
| 3 | 2020-10-12-05 | 星期一 | 12 | 5 | 22° | 1 | 2 | |
| 4 | 2020-10-12-08 | 星期一 | 12 | 8 | 26° | | 0 | |
| 5 | 2020-10-12-11 | 星期一 | 12 | 11 | 28° | | 2 | |
| 6 | 2020-10-12-14 | 星期一 | 12 | 14 | 28° | 0.5 | 3 | |
| 7 | 2020-10-12-17 | 星期一 | 12 | 17 | 26° | 0.8 | 3 | |
| 8 | 2020-10-12-20 | 星期一 | 12 | 20 | 24° | 0.6 | 2 | |
| 9 | 2020-10-12-23 | 星期一 | 12 | 23 | 24° | 0.7 | 1 | |
| 10 | 2020-10-13-02 | 星期二 | 13 | 2 | 23° | 0.2 | 1 | |
| 11 | 2020-10-13-05 | 星期二 | 13 | 5 | 22° | 0.2 | 1 | |
| 12 | 2020-10-13-08 | 星期二 | 13 | 8 | 25° | | 1 | |
| 13 | 2020-10-13-11 | 星期二 | 13 | 11 | 27° | | 2 | |
| 14 | 2020-10-13-14 | 星期二 | 13 | 14 | 27° | 0.6 | 1 | |
| 15 | 2020-10-13-17 | 星期二 | 13 | 17 | 25° | 0.8 | 1 | |
| 16 | 2020-10-13-20 | 星期二 | 13 | 20 | 24° | 0.4 | 1 | |
| 17 | 2020-10-13-23 | 星期二 | 13 | 23 | 24° | | 1 | |
| 18 | 2020-10-14-02 | 星期三 | 14 | 2 | 24° | 0.5 | 1 | |
| 19 | 2020-10-14-05 | 星期三 | 14 | 5 | 23° | 0.4 | 2 | |
| 20 | 2020-10-14-08 | 星期三 | 14 | 8 | 25° | | 0 | |
| 21 | 2020-10-14-11 | 星期三 | 14 | 11 | 27° | 0.9 | 1 | |
| 22 | 2020-10-14-14 | 星期三 | 14 | 14 | 25° | 3.9 | 3 | |
| 23 | 2020-10-14-17 | 星期三 | 14 | 17 | 24° | 1.4 | 1 | |
| 24 | 2020-10-14-20 | 星期三 | 14 | 20 | 23° | 0.2 | 2 | |
| 25 | 2020-10-14-23 | 星期三 | 14 | 23 | 22° | 0.2 | 1 | |
| 26 | 2020-10-15-02 | 星期四 | 15 | 2 | 22° | | 2 | |
| 27 | 2020-10-15-05 | 星期四 | 15 | 5 | 21° | | 2 | |
| 28 | 2020-10-15-08 | 星期四 | 15 | 8 | 24° | | 0 | |
| 29 | 2020-10-15-11 | 星期四 | 15 | 11 | 27° | | 2 | |
| 30 | 2020-10-15-14 | 星期四 | 15 | 14 | 27° | 0.2 | 2 | |
| 31 | 2020-10-15-17 | 星期四 | 15 | 17 | 25° | 0.2 | 1 | |
| 32 | 2020-10-15-20 | 星期四 | 15 | 20 | 23° | 0.2 | 3 | |
| 33 | 2020-10-15-23 | 星期四 | 15 | 23 | 23° | 0.7 | 1 | |
| 34 | 2020-10-16-02 | 星期五 | 16 | 2 | 22° | 0.7 | 2 | |
| 35 | 2020-10-16-05 | 星期五 | 16 | 5 | 21° | | 1 | |
| 36 | 2020-10-16-08 | 星期五 | 16 | 8 | 25° | | 1 | |
| 37 | 2020-10-16-11 | 星期五 | 16 | 11 | 26° | | 1 | |
| 38 | 2020-10-16-14 | 星期五 | 16 | 14 | 26° | 0.6 | 2 | |
| 39 | 2020-10-16-17 | 星期五 | 16 | 17 | 24° | 1.6 | 1 | |
| 40 | 2020-10-16-20 | 星期五 | 16 | 20 | 23° | 2.4 | 2 | |

星期一

瀏覽器上看到的排列



直接輸入 GPS 座標

```
windy_firefox_selenium_InputGpsLocation.py - D:\Python_projects\windy\wi...
File Edit Format Run Options Window Help
browser = webdriver.Firefox(firefox_binary=binary)
#browser = webdriver.Firefox(firefox_binary=binary, options=options) #h
browser.get(URL_HOME)

gps_pos = '22.6874, 120.3074' # 22.6874° N, 120.3074° E 左營高鐵
search = browser.find_element_by_xpath(xpath='//*[@id="q"]')
search.send_keys(gps_pos)
print('Sending Search Data and retrieving suggestions, Please wait...')
time.sleep(2)

search.send_keys(Keys.ARROW_DOWN)
time.sleep(2)
search.send_keys(Keys.ENTER)
time.sleep(2)
```

windy.com 除了可以輸入地名查詢外，也可以直接輸入 GPS 座標，不過不是所有格式都支援

輸入帳號和密碼以獲得每小時資料

1. login

```
browser.get(URL_HOME)
time.sleep(2)
LOGIN_REGISTER = 'div[id="login"]'
USERNAME_ENTRY = 'input[id="username"]'
USERNAME = 'xx'
PASSWORD_ENTRY = 'input[id="password"]'
PASSWORD = 'XXX'
LOGIN_BUTTON = 'button[id="login"]'
register_login = browser.find_element_by_css_selector(LOGIN_REGISTER)
if register_login:
    register_login.click()
    time.sleep(2)
    username = browser.find_element_by_css_selector(USERNAME_ENTRY)
    time.sleep(1)
    username.send_keys(USERNAME)
    username = browser.find_element_by_css_selector(PASSWORD_ENTRY)
    time.sleep(1)
    username.send_keys(PASSWORD)
    login = browser.find_element_by_css_selector(LOGIN_BUTTON)
    login.click()
    time.sleep(2)
```

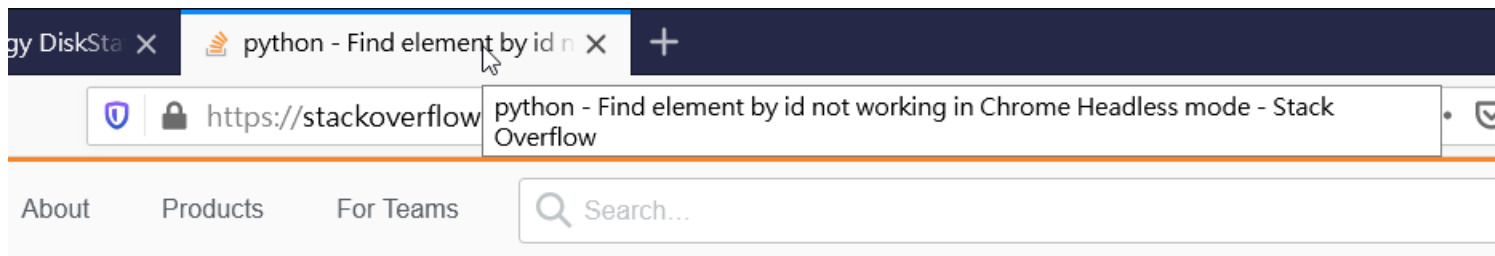
2. switch mode

```
HOUR_SWITCH = 'div[data-t="D 1H FORECAST"]'
hour_switch = browser.find_element_by_css_selector(HOUR_SWITCH)
hour_switch.click()
time.sleep(3) # for browser to load data
```

3. logout

```
IMG_ICON = 'img[class="avatar"]'
img_icon = browser.find_element_by_css_selector(IMG_ICON)
img_icon.click()
time.sleep(1)
LOGOUT = 'a[href="#logout"]'
logout = browser.find_element_by_css_selector(LOGOUT)
logout.click()
time.sleep(1)
```

Chrome 下的 headless Mode



1 Answer

Active

Oldest

Votes

for headless browser you have to set the window size to fire on event. Because headless browser can't recognise where to click without window size.

1

```
options= Options()
options.add_argument('--headless')
options.add_argument('window-size=1920x1080');
```

在 headless 模式下得指定 window-size



share improve this answer follow

edited Jul 18 '19 at 8:46

answered Jul 17 '19 at 16:34



Kunduk

22.7k ● 2 ● 9 ● 30

讓程式在群暉主機上執行成為可能



joyzoursky/python-chromedriver ☆

By joyzoursky · Updated 3 months ago

Python with Chromedriver, for running automated tests

Container

感謝作者

Overview

Tags

Dockerfile

Builds

docker-python-chromedriver

Python with Chromedriver, for running automated tests

Quick Try

```
$ git clone https://github.com/joyzoursky/docker-python-chromedriver.git
$ cd docker-python-chromedriver
$ docker run -it -w /usr/workspace -v $(pwd):/usr/workspace joyzoursky/python-chromedriver:3.7 bash
/usr/wrokspace# pip install selenium
/usr/wrokspace# python test_script.py
```

Image includes

需要的工具都包裝在一起了

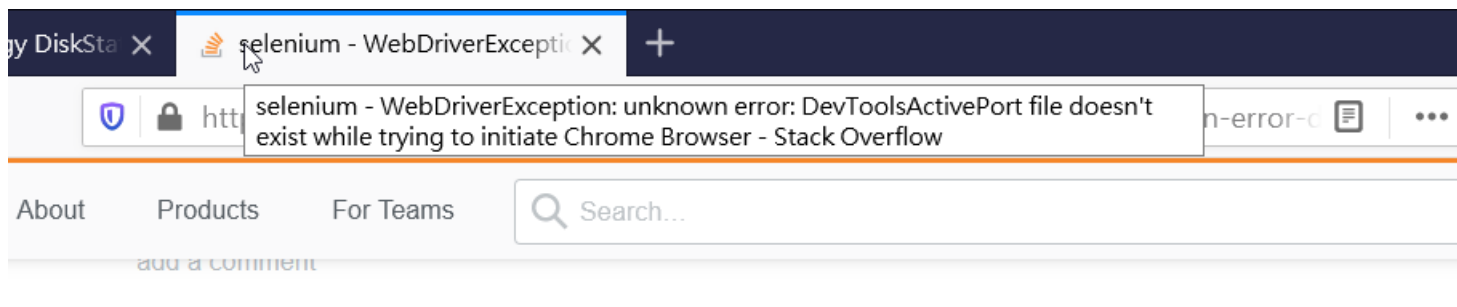
- Python (Debian or Alpine based)
- Google Chrome
- Chromedriver
- Selenium (in some versions)
- Xvfb (in some versions)

Versions

Python 3 (Debian-based)

- `3.8`, `latest` (based on python:3.8, Dockerfile)
- `3.8-selenium` (based on python:3.8 with selenium installed, Dockerfile)

在群暉的 docker 環境下的限制



▲ I had the same problem in python. The above helped. Here is what I used in python -

24



```
chrome_options = Options()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome('/path/to/your_chrome_driver_dir/chromedriver',chrome_options=chr
```

這三個選項似乎缺一不可

share improve this answer follow

answered Jun 13 '18 at 1:15



Vinodh Periyasamy

241 ● 1 ● 2

將 Windy 爬蟲納入任務排程表

新增 編輯 執行 動作 儲存 設定

| 已啟動 | 擁有者 | 應用程式 | 任務名稱 | 動作 | 下次執行時間 |
|-------------------------------------|---------------|----------|------------------|----------------------------|------------------|
| <input checked="" type="checkbox"/> | root | 使用者定義指令碼 | windy_webscraper | 執行: docker container st... | 2020-10-16 19:00 |
| <input checked="" type="checkbox"/> | root | | | container st... | 2020-10-17 00:00 |
| <input checked="" type="checkbox"/> | root | | | 和 | 2020-10-17 00:20 |
| <input checked="" type="checkbox"/> | Administrator | | | HONIOE... | 2020-10-17 10:30 |
| <input checked="" type="checkbox"/> | root | | | 則功能的... | 2020-10-18 00:00 |

編輯任務

一般 排程 任務設定

日期

在以下時間執行

每天

在以下日期執行

2020/10/16

不重複

時間

開始執行時間: 01 : 00

執行頻率: 每 6 小時

最後執行時間: 19:00

執行命令

使用者定義指令碼

```
docker container start sel_chrome
docker container exec sel_chrome sh -c "cd /usr/workspace ; python3
windy_chrome_selenium_RegisteredUser_docker.py ; exit"
docker container stop sel_chrome
```

- (1) 每天測試下載 4 次分別 1,7,13 和 19 時
- (2) 每次執行時，CPU 利用率會從不到 10% 上升至 6??% 左右，RAM 會多用 4??MB

When and how often are the weather data updated?

公開的每 3 小時間隔資料似乎是一天兩次更新，付費的預計 2020 底每天 4 次更新

TomSlavkovsky 2016年12月2日 下午11:29

The table below describes when and how often are the weather models updated. All times are in UTC.

| Model / Runtime | Update interval | 0:00 | 6:00 | 12:00 | 18:00 |
|-----------------|-----------------|-------|-------|-------|-------|
| ECMWF 0.1° | 12h | 07:15 | - | 19:15 | - |
| GFS | 6h | 06:15 | 12:15 | 18:15 | 00:15 |
| NESDIS | 5 days | - | - | - | - |
| WaveWatch3 | 6h | 06:00 | 12:00 | 18:00 | 00:00 |
| NEMS Europe | 12h | 08:30 | - | 20:30 | - |
| NAM CONUS | 6h | 04:15 | 10:15 | 16:15 | 22:15 |
| NAM Alaska | 6h | 03:15 | 09:15 | 15:15 | 21:15 |
| NAM Hawaii | 6h | 04:15 | 10:15 | 16:15 | 22:15 |
| ICON-EU | 6h | 04:30 | 10:30 | 16:30 | 22:30 |

無法完全掌控 loading time

```
63 #gps_pos = '22.6874, 120.3074' # 22.6874° N, 120.3074° E 左營高鐵
64 search = browser.find_element_by_xpath(xpath='//*[@id="q"]')
65 search.send_keys(gps_pos)
66 print('Sending Search Data and retrieving suggestions, Please wait...')
67 time.sleep(2)
68
69 search.send_keys(Keys.ARROW_DOWN)
70 time.sleep(2)
71 search.send_keys(Keys.ENTER)
72 time.sleep(10) # 2 may fail, 5 seems ok
73
74 HOUR_SWITCH = 'div[data-t="D_1H_FORECAST"]'
75 hour_switch = browser.find_element_by_css_selector(HOUR_SWITCH)
76 hour_switch.click()
77 time.sleep(5) # for browser to load data, 3 seems ok
```

1. 輸入 GPS 座標後，第一次載入資料

2. 從「3小時預報」切換成1小時

```
113 # switch to waves mode and extract forecast records
114 waves_select_xpath='//*[@id="detail-box"]/div[2]/div[2]'
115 waves_select = browser.find_element_by_xpath(xpath=waves_select_xpath)
116 waves_select.click()
117 time.sleep(5) # 2 secs failed
118 waves_xpath='//*[@id="detail-data-table"]/tbody/tr[4]/*' # same as temps
119 waves = browser.find_elements_by_xpath(xpath=waves_xpath)
120 waves_list = []
121 for els in waves:
122     waves_list.append(els.text)
```

3. 從「基本」模式切到「波浪」模式