都會阿嬤
city Grandma

PYTHON

Python 爬蟲教學：爬蟲進化 – 偽裝篇 fake_useragent 介紹

前言 – 阿嬤碎碎念

在寫爬蟲程式的時候，遇到最困擾的事情就是有些網站會阻擋爬蟲，畢竟爬蟲程式會消耗對方伺服器的資源，因此對方有可能會把你的 IP 封鎖、把你的 Python爬蟲程式阻擋下來。

今天將介紹一個 Python 套件 `fake_useragent`，他可以讓我們將程式加上一個 `User-Agent`，假裝是一個瀏覽器在瀏覽該網站。

User-Agent

如果你按 `F12` [開發者工具] → Network → 查看某一個資源的 `Request Headers`
就可以看到一個 `User-Agent` 的欄位，那個就是你的瀏覽器會發出去的一串文字，告訴對方你的瀏覽器是什麼、作業系統是什麼。

一般瀏覽器常見的 User-Agent 的格式是

```
User-Agent: Mozilla/5.0 (<system-information>) <platform> (<platform-details>) <extensions>
```

fake-useragent 套件

fake-useragent 套件可以幫助你隨機產生 User-Agent 字串，比起在程式裡寫死的一串文字，fake-useragent 有兩大優點特色：

- grabs up to date useragent from useragentstring.com
- randomize with real world statistic via w3schools.com

白話文翻譯翻譯：

- 自動從 useragentstring.com 抓最新的 user-agent 字串，瀏覽器會更新，User-agent字串當然也需要更新！

- 根據 w3schools.com 統計的瀏覽器使用頻率來產生 user-agent 字串，要偽裝就偽裝到底，瀏覽器出現的頻率也可以考慮進去！

用 pip 安裝

```
$ pip install fake_useragent
```

import 套件並產生一個 UserAgent

```
from fake_useragent import UserAgent

ua = UserAgent()
```

試試看產生不同瀏覽器的 User-Agent 字串

```
ua.ie
```

'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; chromeframe/12.0.742.112)'

```
ue.google
```

'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1664.3 Safari/537.36'

```
ua.firefox
```

'Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:23.0) Gecko/20131011 Firefox/23.0'

```
ua.safari
```

'Mozilla/5.0 (Windows; U; Windows NT 6.0; de-DE) AppleWebKit/533.20.25 (KHTML, like Gecko) Version/5.0.3 Safari/533.19.4'

> *ua.random*

最厲害、最實用、實做中最常用到的是 ua.random，根據真實世界的統計隨機產生一個 User-Agent 字串。

```
ua.random
```

用 requests 實戰股票爬蟲

聲明：本網站上的爬蟲教學為純粹技術分享，請不要進行大量、高頻的爬蟲做出不正當的行為，造成他人的困擾及損害他人的權利！

在 Python 股票分析教學：爬取台積電 (2330) 歷史股價中我們教大家如何從臺灣證券交易所 爬取台積電的股價及其他日成交資訊：

```
import requests
import pandas as pd

dates = [20200201, 20200101, 20191201]
stockNo = 2330
url_template = "https://www.twse.com.tw/exchangeReport/STOCK_DAY?response=html&date={}&stockNo={}"

for date in dates :
    url = url_template.format(date, stockNo)
    file_name = "{}_{}.csv".format(stockNo, date)

    data = pd.read_html(requests.get(url).text)[0]
    data.columns = data.columns.droplevel(0)
    data.to_csv(file_name, index=False)
```

現在我們可以加進 fake_useragent 強化我們的爬蟲：

```
user_agent = ua.random
headers = {'user-agent': user_agent}
```

並且在使用 requests 時把我們創造的 header 加進去

```
requests.get(url, headers=headers)
```

requests.get(url, headers=headers)另外我們也可以在每爬完一個檔案後，讓爬蟲休息一下，因為如果短時間內大量爬取檔案，有很大的機率會被該網站擋下來、鎖 IP。

```
time.sleep(5)
```

完整程式碼

聲明：本網站上的爬蟲教學為純粹技術分享，請不要進行大量、高頻的爬蟲做出不正當的行為，造成他人的困擾及損害他人的權利！

```
import requests
import pandas as pd
import time
from fake_useragent import UserAgent


dates = [20200201, 20200101, 20191201]
stockNo = 2330
url_template = "https://www.twse.com.tw/exchangeReport/STOCK_DAY?response=html&date={}&stockNo={}"
ua = UserAgent()
user_agent = ua.random

for date in dates :
    # 產生 headers
    headers = {'user-agent': user_agent}

    # url
    url = url_template.format(date, stockNo)

    # output file name
```

```
file_name = "{}_{}.csv".format(stockNo, date)

# 開始爬取檔案
data = pd.read_html(requests.get(url, headers=headers).text)[0]
data.columns = data.columns.droplevel(0)
data.to_csv(file_name, index=False)
time.sleep(5)
```

Python, 爬蟲