



WEATHER REPORT

資管一A 康哲瑋



MOTIVATION

桃園市的降雨天數在全臺灣可以排上前幾名，這也造成在中央大學時常遇到下雨天，所以我想製作一個爬蟲抓取中央氣象局的天氣資料，再利用Line傳送訊息，我就可以知道桃園及實的天氣狀況。



PROBLEM

1. 網頁是動態加載但BeautifulSoup的套件並不支援
2. 網頁回應的資料型態是javascript不是json
3. LineNotify無法互動

SOLUTION - 1

發現網頁資料是動態網頁後，想說可以用Selenium爬蟲，然而在實作的過程中卻爬取不了完整的html，有一半的程式碼顯示不出來，我發現Selenium較適合用來爬取需要使用者登入、滾動卷軸讓網頁載入的資料。

SOLUTION - 2

因為是動態網頁，在開發人員工具頁選取欄位 Network、Fetch/XHR，依序確認左列清單得到的 Response 找到要爬取的資料，但 Response 的資料型態是 javascript 而不是 json，所以我先將字串更改為 json 格式，再轉為 python 的資料型態，便可以用字典找到對應的資料。

SOLUTION - 3

在成功利用爬蟲爬取資料後，想利用Line Notify傳送資料，但在完成後發現每次要取得天氣資料時都必須執行一次程式碼，有點太麻煩了，所以我想說可以結合Line Bot，透過使用者輸入關鍵字的方式取得資料

```
graph TD; PACKAGE[PACKAGE] --- REQUEST[REQUEST]; PACKAGE --- LINEBOT[LINEBOT]; PACKAGE --- BEAUTIFUL_SOUP[BEAUTIFUL SOUP]; PACKAGE --- SELENIUM[SELENIUM];
```

PACKAGE

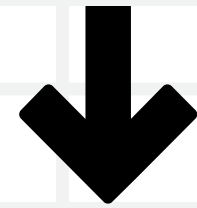
REQUEST

LINEBOT

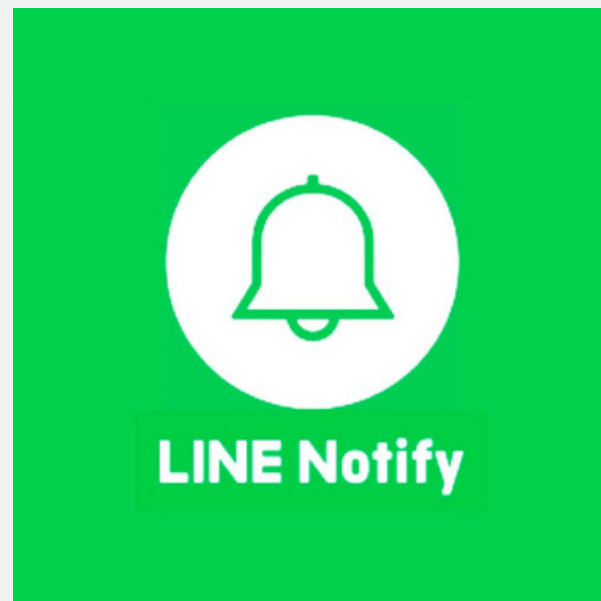
**BEAUTIFUL
SOUP**

SELENIUM

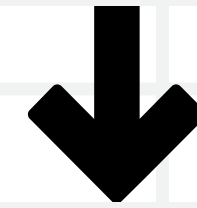
DIFFERENT METHOD



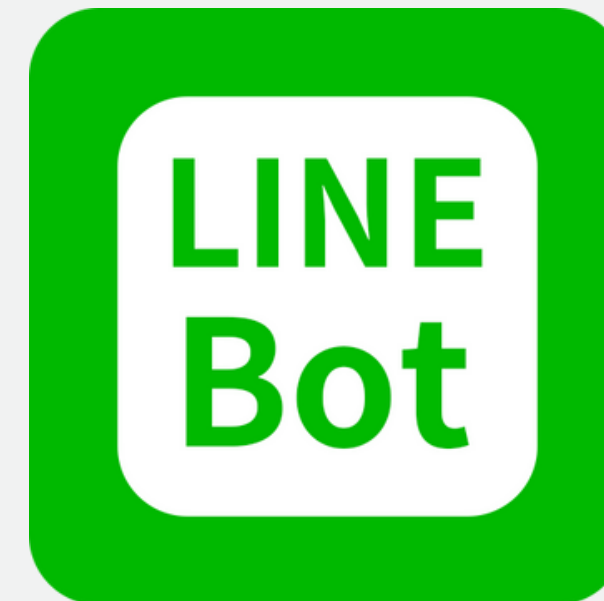
LINE NOTIFY



每次發送通知都需要執行程式碼

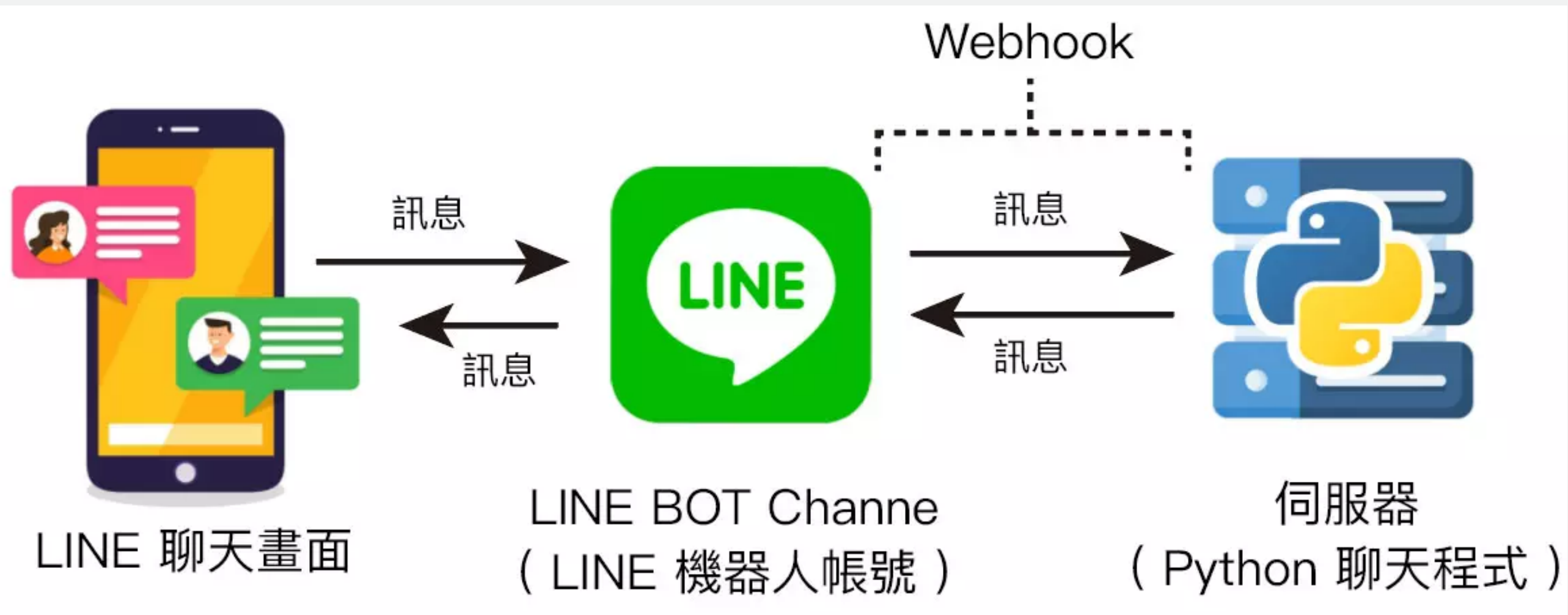


LINE BOT



使用者可以透過訊息取得天氣狀況

HOW LINEBOT WORK



CODE

WeatherData.py

```
import requests
import json

def getTimeData():
    r = requests.get('https://www.cwb.gov.tw/Data/js/GT/TableData_GT_T_68.js?T=202308010000')

    # time data
    time_start = r.text.find('C')+4
    time_end = r.text.find(',')-1
    time_data = "<" + r.text[time_start:time_end].replace("<br>", " ") + ">"
    return time_data

def getWeatherData():
    r = requests.get('https://www.cwb.gov.tw/Data/js/GT/TableData_GT_T_68.js?T=202308010000')

    # weather data
    weather_start = r.text.find('6800200')+9
    weather_end = r.text.find('}', weather_start, 350)+1

    weather_data = r.text[weather_start:weather_end].replace("'", '"').replace("-", " ")
    weather_dic = json.loads(weather_data)

    weather_data = "          溫度 : " + weather_dic['C_T'] + "°C\n體感溫度 : " + weather_dic['C_T'] + "°C"
    return weather_data

def mergeData():
    mergeData = "\n" + getTimeData() + "\n===== \n" + getWeatherData()
    return mergeData
```

SeleniumTest.py

```
import requests
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from PIL import Image
import time

url = 'https://www.cwb.gov.tw/V8/C/'
chrome_options = Options()
chrome_options.add_experimental_option("detach", True)
# chrome_options.add_argument("--headless")
chrome_options.add_argument("--start-maximized")
driver = webdriver.Chrome(options = chrome_options)
driver.get(url)

#select district
select_element = driver.find_element(By.XPATH, '//*[@id="home-select-town"]')
selectDistrict = Select(select_element)
selectDistrict.select_by_visible_name('中壢區')

#click button
button = driver.find_element(By.CLASS_NAME, 'btn.btn-default.btn-block.btn-sm')
button.click()

r = requests.get(driver.current_url)
result = r.text
print(result)
# move = driver.find_element(By.TAG_NAME, 'body')
# move.send_keys(Keys.PAGE_DOWN)
# time.sleep(1)

driver.quit()
```

CODE

LineNotify.py

```
import requests
import WeatherData

def sendLineNotify(msg):
    url = 'https://notify-api.line.me/api/notify'
    token = 'bDdLChdJvNjdM2ju5pSRDgVyFTEK7XH5vPZurhUOnlm'
    # token = 'Iw00zNSJ1wlvCd207Ntx1tE0JUyggzYHlPHGPpqYSM9'
    headers = {
        'Authorization': 'Bearer ' + token,
        "Content-Type": "application/x-www-form-urlencoded"
    }
    data = {
        'message': msg
    }
    r = requests.post(url, headers = headers, data = data)
    print(r.status_code)

if __name__ == '__main__':
    message = WeatherData.mergeData()
    sendLineNotify(message)
```

LineBot.py

```
from flask import Flask, request,
from linebot import LineBotApi, WebhookHandler
from linebot.exceptions import InvalidSignatureError
from linebot.models import MessageEvent, TextMessage, TextSendMessage
import configparser
import WeatherData

LineBot = Flask(__name__)

# LineBot info
config = configparser.ConfigParser()
config.read('config.ini')

line_bot_api = LineBotApi(config.get('line-bot', 'channel_access_token'))
handler = WebhookHandler(config.get('line-bot', 'channel_secret'))

# callback
@LineBot.route("/callback", methods = ['POST'])
def callback():
    signature = request.headers['X-Line-Signature']

    body = request.get_data(as_text = True)
    LineBot.logger.info("Request body: " + body)

    try:
        print(body, signature)
        handler.handle(body, signature)

    except InvalidSignatureError:
        abort(400)


    return 'OK'

# send weather data
@handler.add(MessageEvent, message = TextMessage)
def sendWeather(event):
    if(event.message.text == "天氣"):
        line_bot_api.reply_message(
            event.reply_token,
            TextSendMessage(text = "_____及時天氣_____" + WeatherData.mergeData())
        )

if __name__ == "__main__":
    LineBot.run()
```



NGROK

- LineBot只能呼叫 public 網址，此外Webhook URL 必須有SSL，也就是https
 - 而ngrok 是一個 reverse proxy 的工具，透過將 local 的 port 對應到一個 public 的地址，讓外部可以透過該public地址連到 local 端，繼而取得或傳送資料。
- 

NGROK

C:\Users\USER\Desktop\pythc X + v

ngrok

Send your ngrok traffic logs to Datadog: <https://ngrok.com>

Session Status

online

Session Expires

57 minutes

Terms of Service

<https://ngrok.com/tos>

Version

3.3.1

Region

Japan (jp)

Latency

36ms

Web Interface

<http://127.0.0.1:4040>

Forwarding

<https://1c86-140-115-217-12.jp.ngrok.io> -> <http://localhost:5000>

Connections

ttl	opn	rt1	rt5	p50	p90
28	0	0.02	0.01	0.60	0.69

HTTP Requests

POST /callback

200 OK

POST /callback

200 OK

POST /callback

200 OK

Webhook settings

Webhook URL ?

<https://24fe-140-115-217-12.jp.ngrok.io/callback>

Verify

Edit

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\USER\Desktop\python\WebCrawler> python LineBot.py

* Serving Flask app 'LineBot'

* Debug mode: off

WARNING: This is a development server. Do not use it in a prod

* Running on <http://127.0.0.1:5000>

Press CTRL+C to quit

RESULT

LINE NOTIFY



LINE BOT





REFERENCES



🔍 REFERENCES 1

<https://www.learncodewithmike.com/2020/06/python-line-bot.html>

🔍 REFERENCES 2

<https://ithelp.ithome.com.tw/articles/10234576>

🔍 REFERENCES 3

<https://ithelp.ithome.com.tw/articles/10301898>

🔍 REFERENCES 4

老師上課簡報