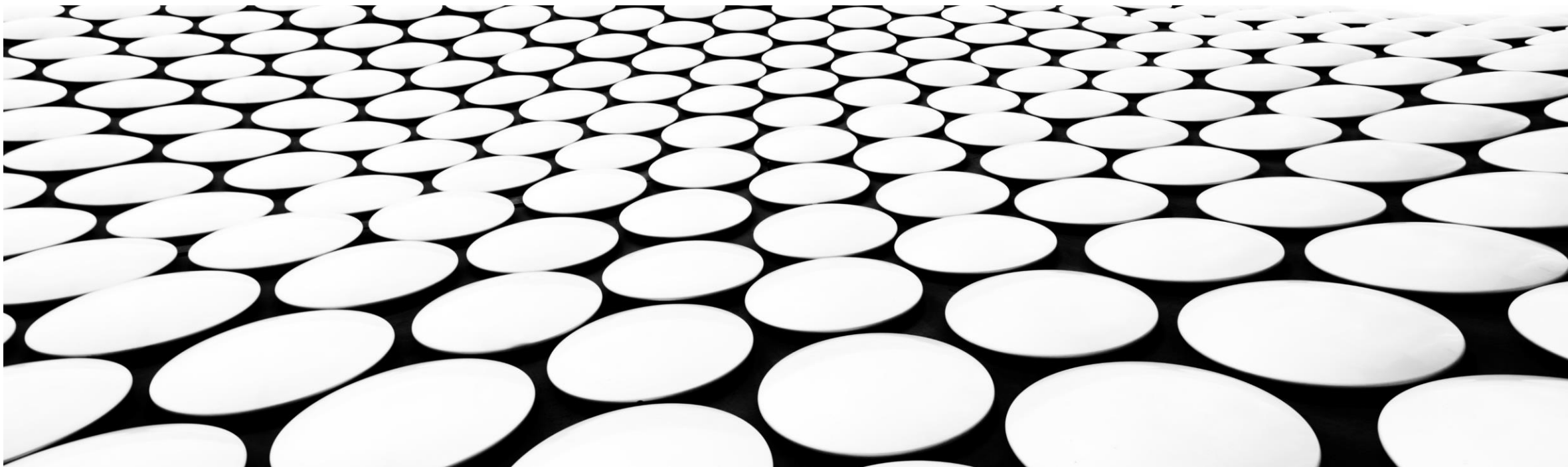

LINE Bot 開發指南

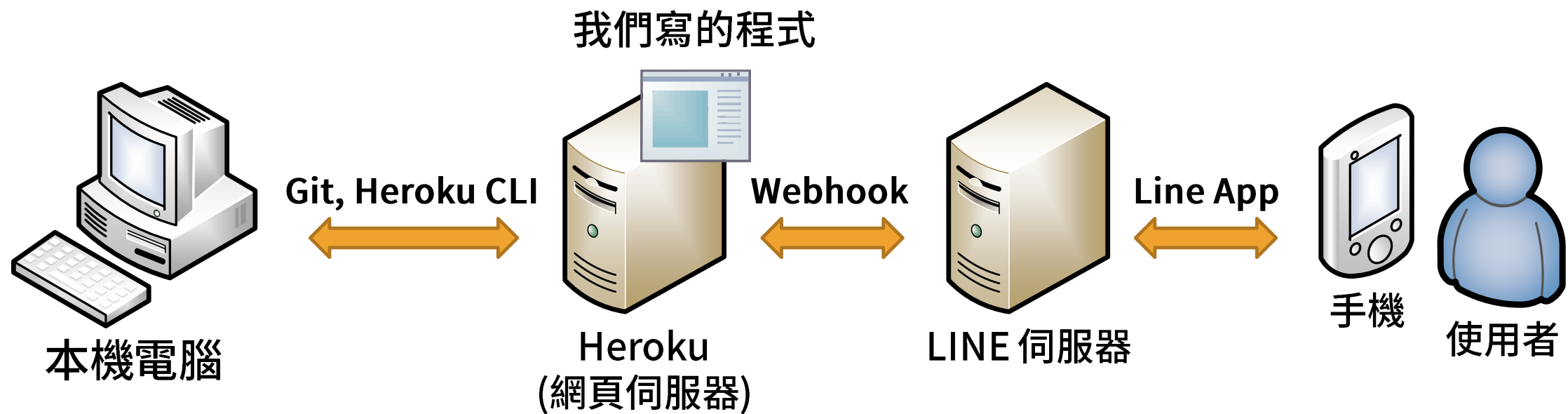
MARs教育團隊 – 孫汶琳 交大電機博士候選人



目錄

- Git 版本管理軟體
- Heroku 網頁伺服器
- LINE 開發者 (LINE Developers)
- 開發實務 – 第一個 LINE Bot 聊天機器人

LINE Bot 運作流程



Git 版本控制軟體安裝與設定

- 到 Git 官方網站下載 git : <https://git-scm.com/downloads>
- 安裝完後打開命令提示字元 (cmd) 輸入: **git --version** 檢查是否安裝成功

```
C:\Users\XiaoLe>git --version  
git version 2.20.1.windows.1
```

- 設定 git 使用者名稱和電子信箱
 - 設定名稱: **git --global user.name "名字"**

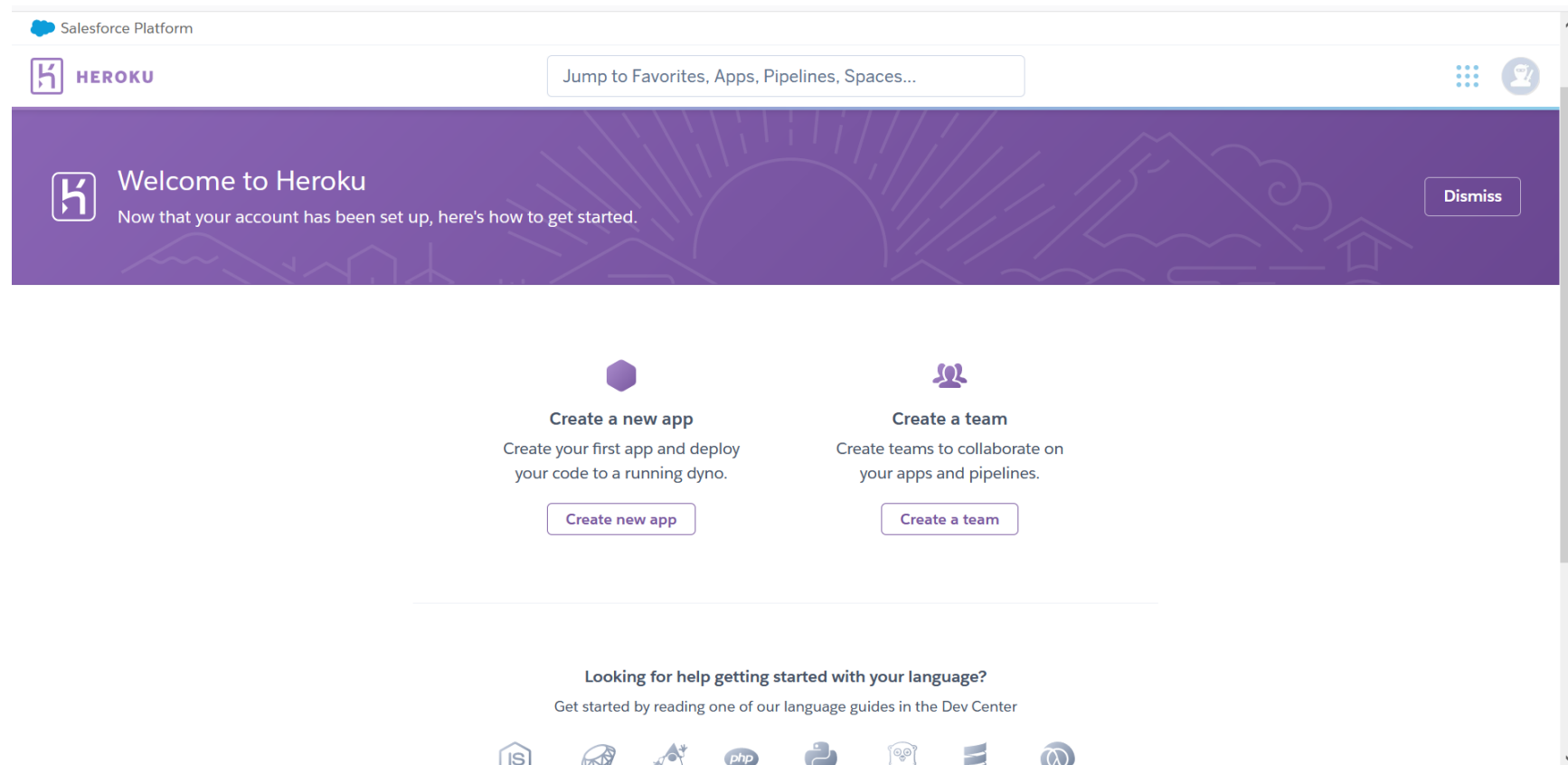
```
C:\Users\XiaoLe>git config --global user.name "Winnie Sun"
```

- 設定電子信箱: **git --global user.email "電子信箱"**

```
C:\Users\XiaoLe>git config --global user.email "wlsun.ee06@nycu.edu.tw"
```

Heroku 帳號申請

- 到 Heroku 官方網站申請帳號: [Link](#)
 - 要確認到信箱驗證和設定密碼成功唷!
- 成功登入後會到主頁面 (dashboard)



Heroku 創建網頁App

- 在主頁面選擇 Create new app
- 輸入 app name (小寫英文字幕開頭)後，選擇 Create
- 創建完後到該app下選擇 Setting，複製自己的 Heroku git URL



1

Create a new app

Create your first app and deploy your code to a running dyno.

Create new app



Create a team

Create teams to collaborate on your apps and pipelines.

Create a team

2

Create New App

App name

python-chatbot-winnie

python-chatbot-winnie is available

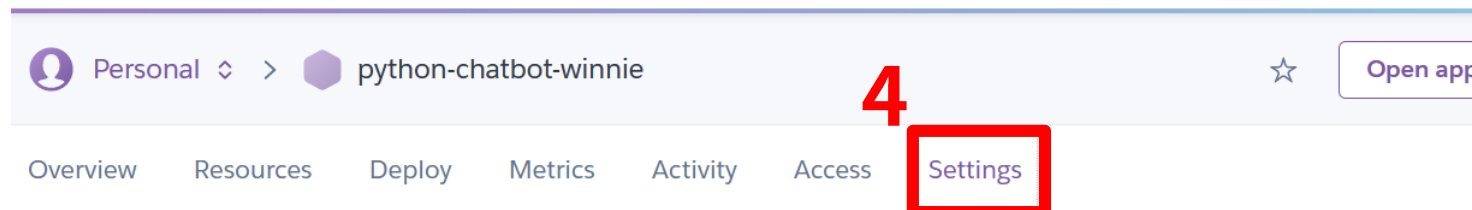
Choose a region

United States

Add to pipeline...

Create app

3



4

App Information

App Name

python-chatbot-winnie

Region United States

Stack heroku-20

Framework No framework detected

Slug size No slug detected

Heroku git URL https://git.heroku.com/python-chatbot-winnie.git

5

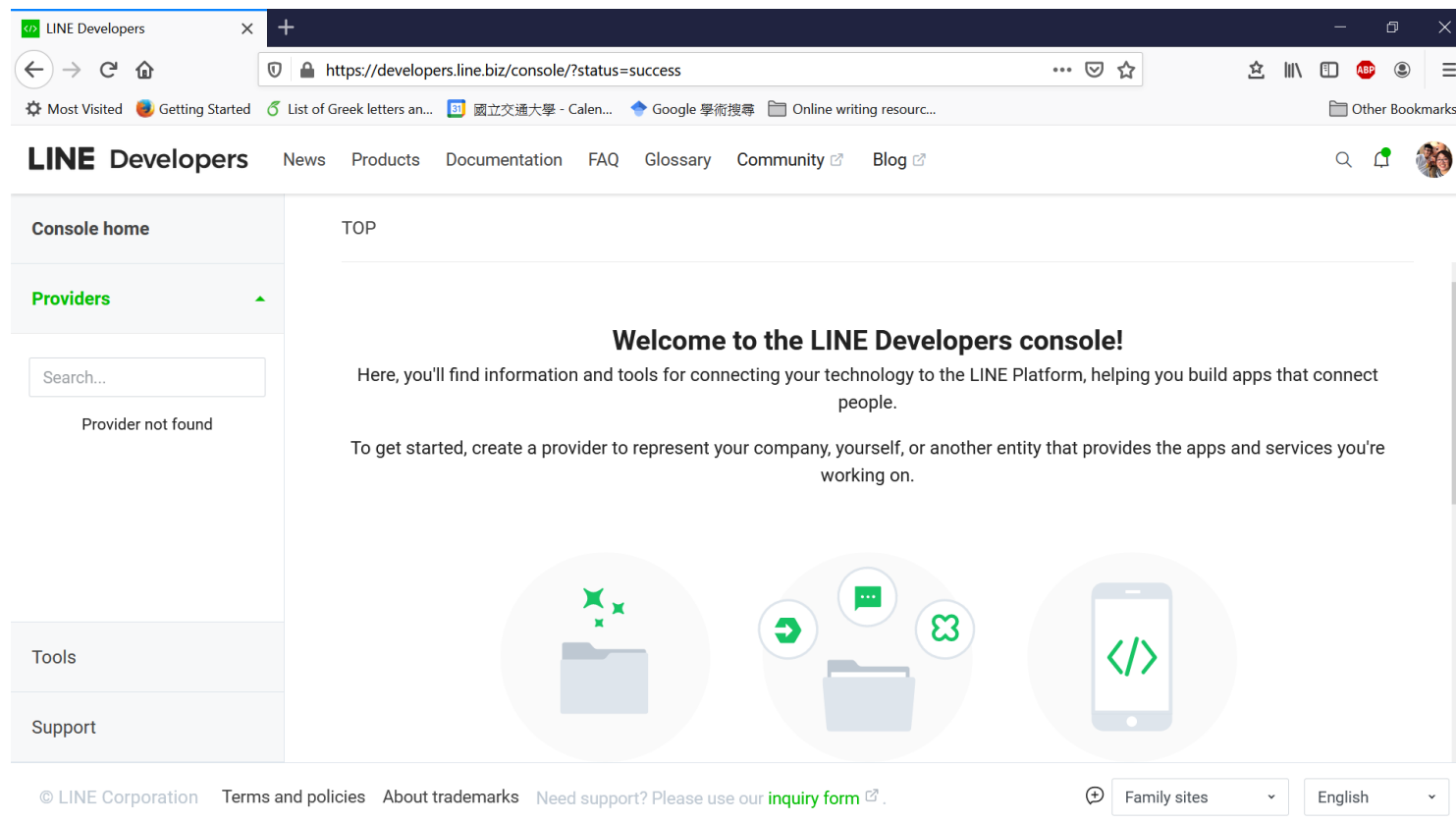
Heroku CLI

- 下載並安裝 Heroku CLI : <https://devcenter.heroku.com/articles/heroku-cli#download-and-install>
- 打開命令提示字元 (cmd)，輸入 **heroku --version**，檢查安裝是否成功

```
C:\Users\XiaoLe>heroku --version  
heroku/7.51.0 win32-x64 node-v12.21.0
```

建立LINE的開發者帳號

- 進入 LINE Developers 開發者頁面: <https://developers.line.biz/zh-hant/>
- 點選 Console，登入自己的 LINE 帳號後，他會要你建立開發者帳號
- 建立成功後會進入到 LINE Developers 管理介面 (LINE Developers console)



設定 LINE 聊天機器人 I

- 在管理介面往下滑，選擇 Create a new provider
- 輸入名稱後按下Create
- 創建完成會在 Providers 的選單看到新的 Admin

Create a new provider

Provider name ² 

孫汶琳

✓ Don't leave this empty

✓ Don't use special characters (4-byte Unicode)

✓ Enter no more than 100 characters

Cancel

³ Create

LINE Developers [News](#) [Products](#) [Documentation](#) [FAQ](#) [Glossary](#) [Community](#) [Blog](#)

Console home

Providers 

Search...

Provider not found

Tools

Support

TOP



Start by creating a provider



Then, create a channel for each app you want to connect to the LINE Platform



Lastly, enter the necessary information to link your apps to your channels, and start building!

¹

Create a new provider

Providers 

Search...

Admin


⁴ 孫汶琳

Tools

設定 LINE 聊天機器人 II

- 點選剛創建好的 Provider
- 選擇 Create a Messaging API channel 創建個人頻道
- 填寫頻道相關資訊後，沒有紅色x就可以按下Create創建了！

Console home

Providers 

Search...

Admin

孫汶琳

Tools

Support

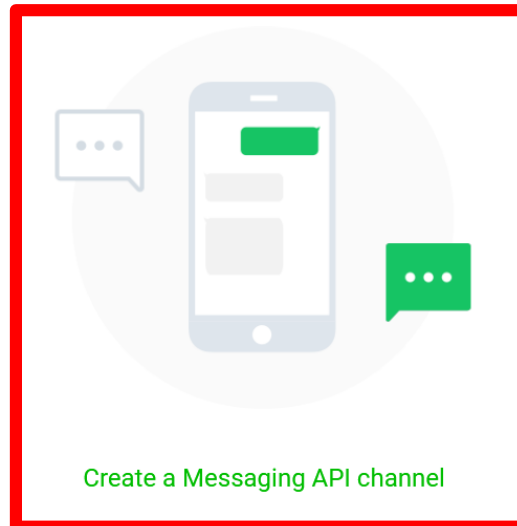
TOP > 孫汶琳

To create one, choose a channel type below



Create a LINE Login channel

2



Create a Messaging API channel

TOP

Channel icon

optional



✓ File type must be one of: PNG, JPG, JPEG, GIF, BMP

✓ File must be no larger than 3 MB

Channel name

Python課程

Note: The channel name can't be changed for seven days.

✓ Don't leave this empty

✓ Don't use special characters (4-byte Unicode)

✓ Enter no more than 20 characters

☒ I have read and agree to the [LINE Official Account Terms of](#)

☒ I have read and agree to the [LINE Official Account API Term](#)

✓ Select the checkbox after reading the related document

4

Create

設定 LINE 聊天機器人 III

- 創建完頻道後，切換到 Messaging API 介面
- 就可以透過 id 或者 QR code 用LINE加好友啦!

TOP > 孫汶琳 > Python課程 > Messaging API



Python課程

Admin

... Messaging API

Basic settings

Messaging API

LIFF

Security

Statistics

Roles

Messaging API settings

Bot information

Bot basic ID

@331ohvfd 

QR code



設定 LINE 聊天機器人 IV

- 同樣是在 Messaging API 介面往下滑找到 Webhook settings
- 按下 Edit 編輯 Webhook URL，根據在Heroku上所創建的 app 名稱輸入

`https://在Heroku的app名稱.herokuapp.com/callback`

- 下方的 Use webhook 也要記得打開

Webhook settings

Webhook URL ⓘ

`https://python-chatbot-winnie.herokuapp.com/callback`

Verify

Edit

Use webhook ⓘ

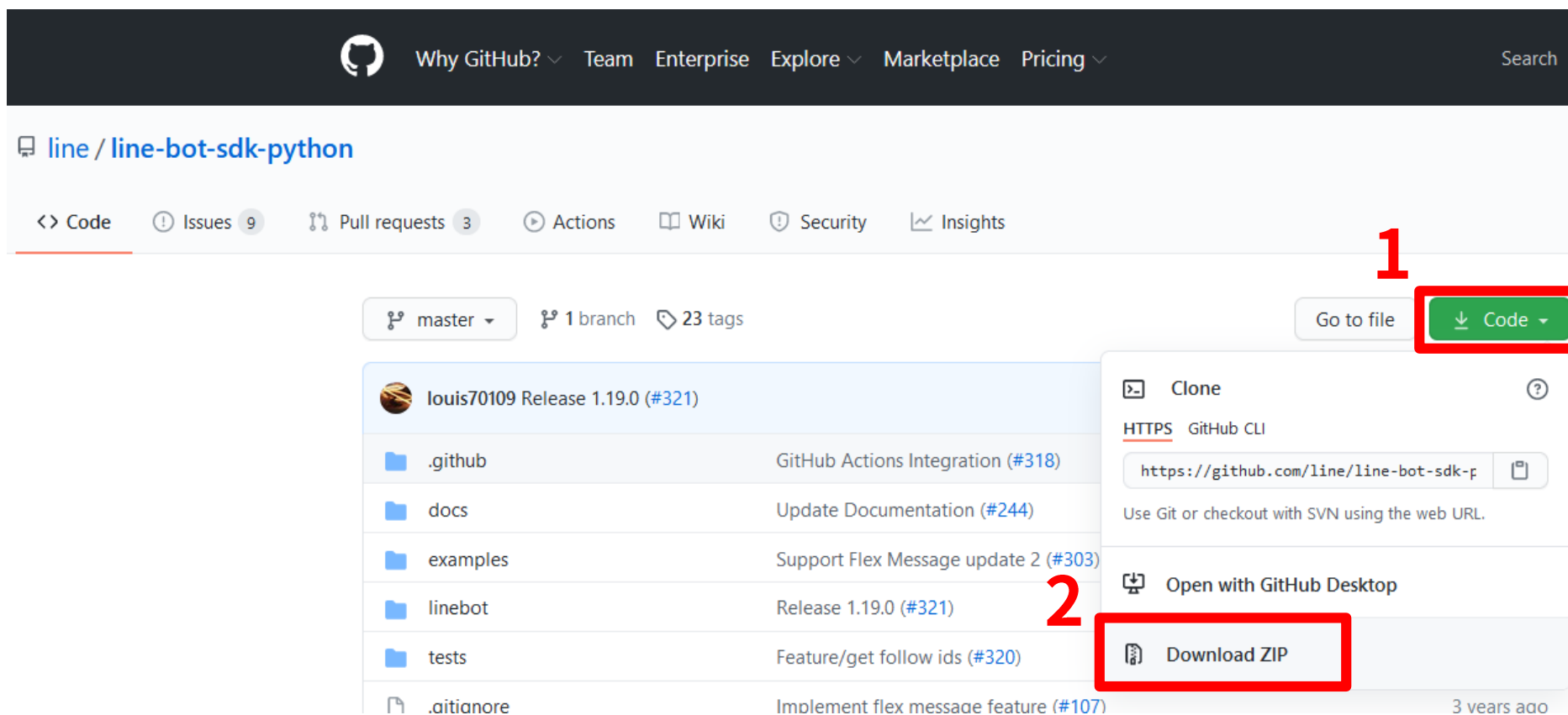


獲取Channel Access Token 和 Channel Secret

- 取得 Channel Access Token
 - Messaging API 頁面滑到最下方，找到 Channel Access Token
 - 按下 issue，會出現一長串亂碼，那就是你的 Channel Access Token 了!
- 取得 Channel Secret
 - Basic Settings 頁面滑到最下方，找到 Channel Secret
 - 會有一小串亂碼，那就是你的 Channel Secret 了!
- 先記得有這兩樣東西，之後編寫 LINE Bot 應用程式的時候會用到!

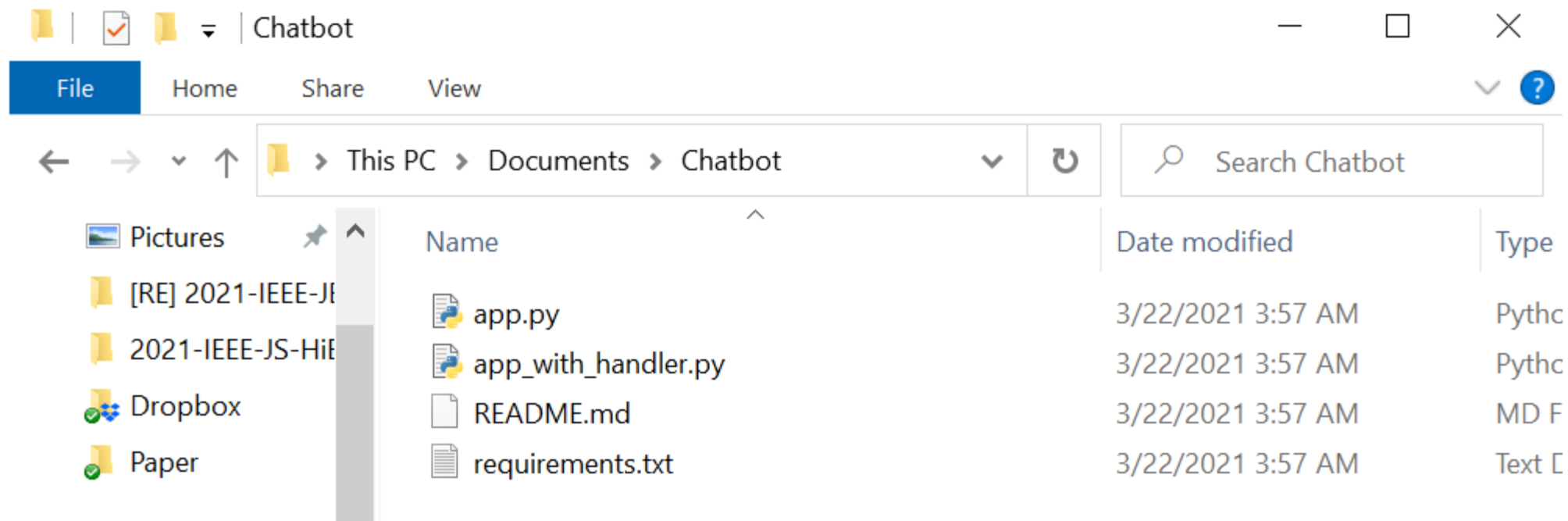
下載官方範例程式

- 到 LINE 官方 Github 帳號的 line-bot-sdk-python 頁面：<https://github.com/line/line-bot-sdk-python>
- 打包下載壓縮檔



使用官方範例程式 I

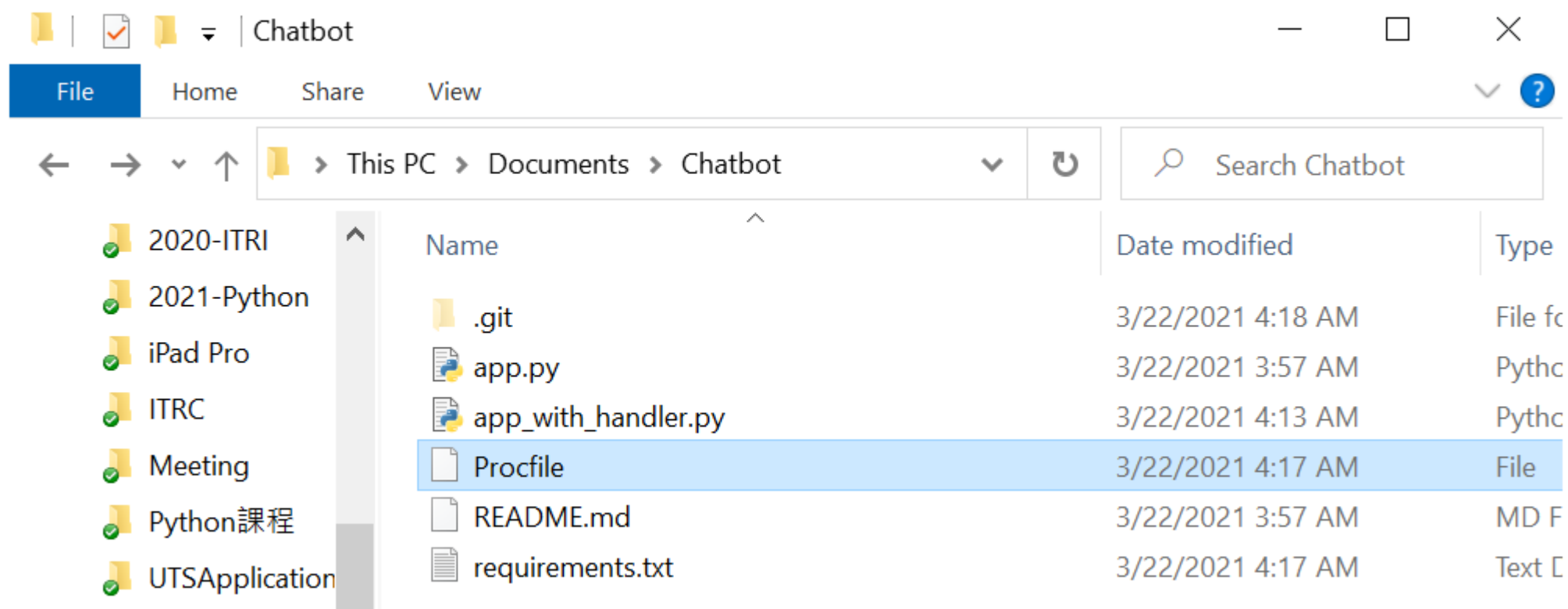
- 在自己電腦上開一個跟Heroku同步用的資料夾，這裡以 Chatbot 為例
- 將壓縮檔裡面 examples 目錄下的 flask-echo 目錄中的所有檔案解壓縮到 Chatbot 資料夾



使用官方範例程式 II

- 在 Chatbot 資料夾中新增一個 Heroku 設定文件，名稱: **Procfile**，不要有副檔名!
- 內容為:

```
web: gunicorn app_with_handler:app
```




修改官方範例程式

- 打開 requirements.txt ，在最下面新增一行，存檔

gunicorn

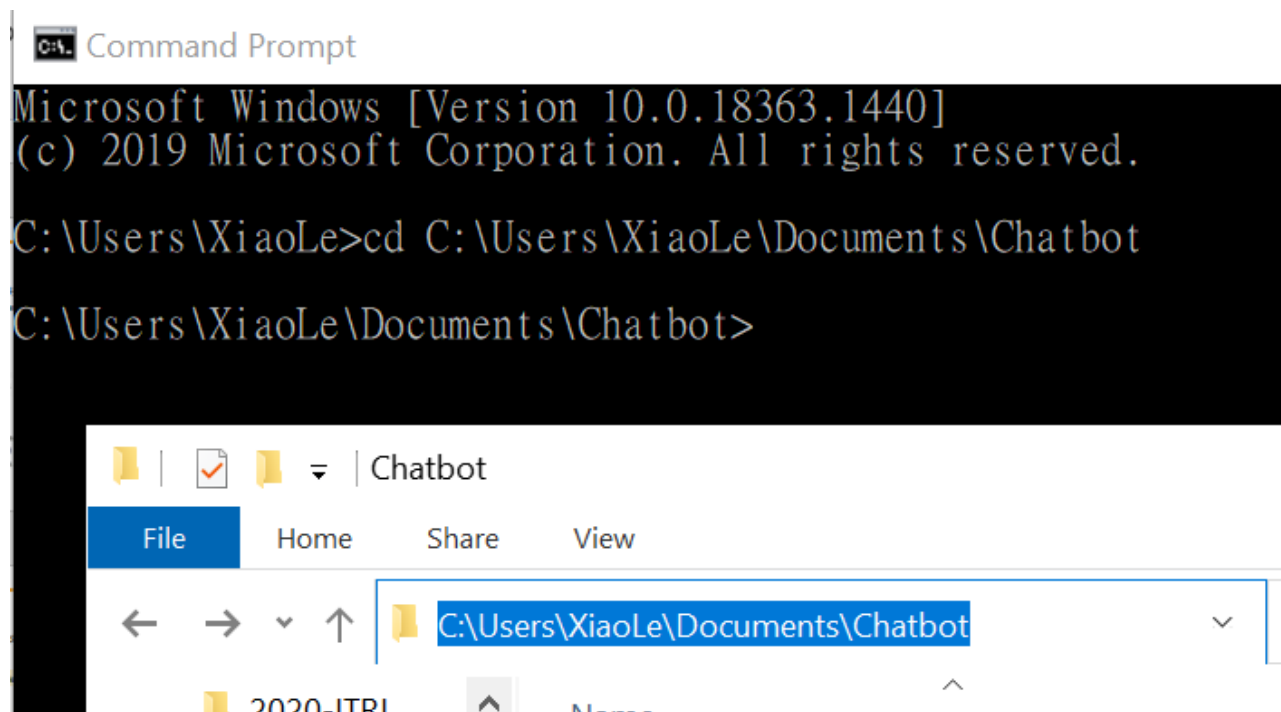
- 打開 app_with_handler.py ，在第32行附近可以看到設定 channel_secret 和 channel_access_token 兩個變數的程式碼，把原本的註解掉後，使用自己的替代 (用String型態)



```
32 # get channel_secret and char
33 #channel_secret = os.getenv('
34 channel_secret = '0ba021d1e71
35 #channel_access_token = os.ge
36 channel_access_token =
    'KsdhW+fZxBILCq2zLqNnSumgeu1v
```

同步到 Heroku I

- 複製 Chatbot 的路徑，打開命令提示字元後，輸入: **cd 複製的路徑**，用來切換到該目錄



同步到 Heroku II

- (只有第一次使用要做!) 輸入 `git init`，創建 git 目錄 (repository)
- 設定 Heroku 的遠端 git 伺服器網址，輸入

```
heroku git:remote -a 自己的Heroku app名稱
```

- 登入 Heroku，輸入

```
heroku login -i
```

- 輸入完帳號、密碼後，確認成功登入!

```
C:\Users\XiaoLe\Documents\Chatbot>heroku git:remote -a python-chatbot-winnie
set git remote heroku to https://git.heroku.com/python-chatbot-winnie.git

C:\Users\XiaoLe\Documents\Chatbot>heroku login -i
heroku: Enter your login credentials
Email [wlsun.ee06@nycu.edu.tw]:
Password: *****
Logged in as wlsun.ee06@nycu.edu.tw
```

同步到 Heroku III

- 記錄所有修改過的檔案，輸入: **(不要忘記最後面的點!)**

```
git add .
```

- 對於本次的修改紀錄給予說明，輸入: **(雙引號內的說明可自由編寫)**

```
git commit -m "一些關於本次修改的說明"
```

- 將修改紀錄同步到 Heroku

```
git push heroku master
```

- 等待同步完成就可以測試啦!

檢查 Heroku 和 LINE Webhook 是否連通

- 在設定 Webhook URL 的頁面，點選 Verify
- 成功會出現 Success 訊息

Webhook settings

Webhook URL ⓘ <https://python-chatbot-winnie.herokuapp.com/callback>

Verify

Edit

Use webhook ⓘ



Success

OK

Verify

Edit

實際測試 LINE 聊天機器人

- 範例程式是一個會回復使用者所輸入的訊息之機器人
- LINE 官方有預設會一些自動回覆的罐頭訊息
- 關掉自動回覆：
 - 到 Messaging API 設定頁面
 - 往下拉會看到 LINE Official Account features
 - 將其中的 Auto-reply messages 關掉 (Disabled) 即可

TOP > 孫汶琳 > Python課程 > Messaging API

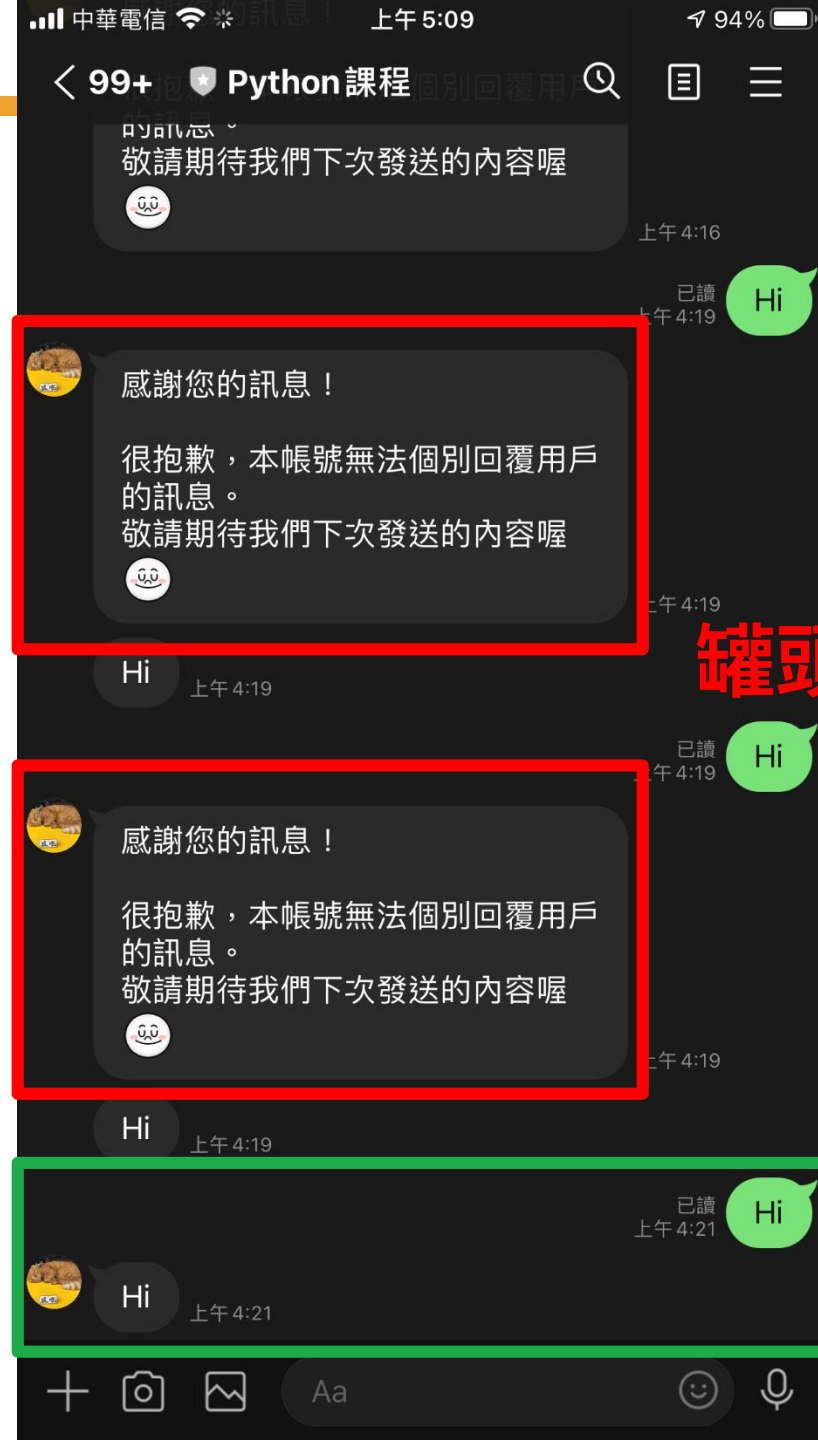
LINE Official Account features

Edit the message text and other settings for these features in the LINE Official Account Manager

Allow bot to join group chats ③ Disabled

Auto-reply messages ③ Disabled

成功!



罐頭訊息

LINE Bot 機器人 – 分析程式

- 主要修改的部分在 handler 的部分
- 範例程式的 handler 是針對收到 文字訊息 時的處理

```
66  @handler.add(MessageEvent, message=TextMessage)
67  def message_text(event):
68      line_bot_api.reply_message(
69          event.reply_token,
70          TextSendMessage(text=event.message.text)
71      )
```

設定handler所負責的訊息種類

回覆訊息的函式

使用者所輸入的訊息

LINE Bot 機器人 – 修改程式

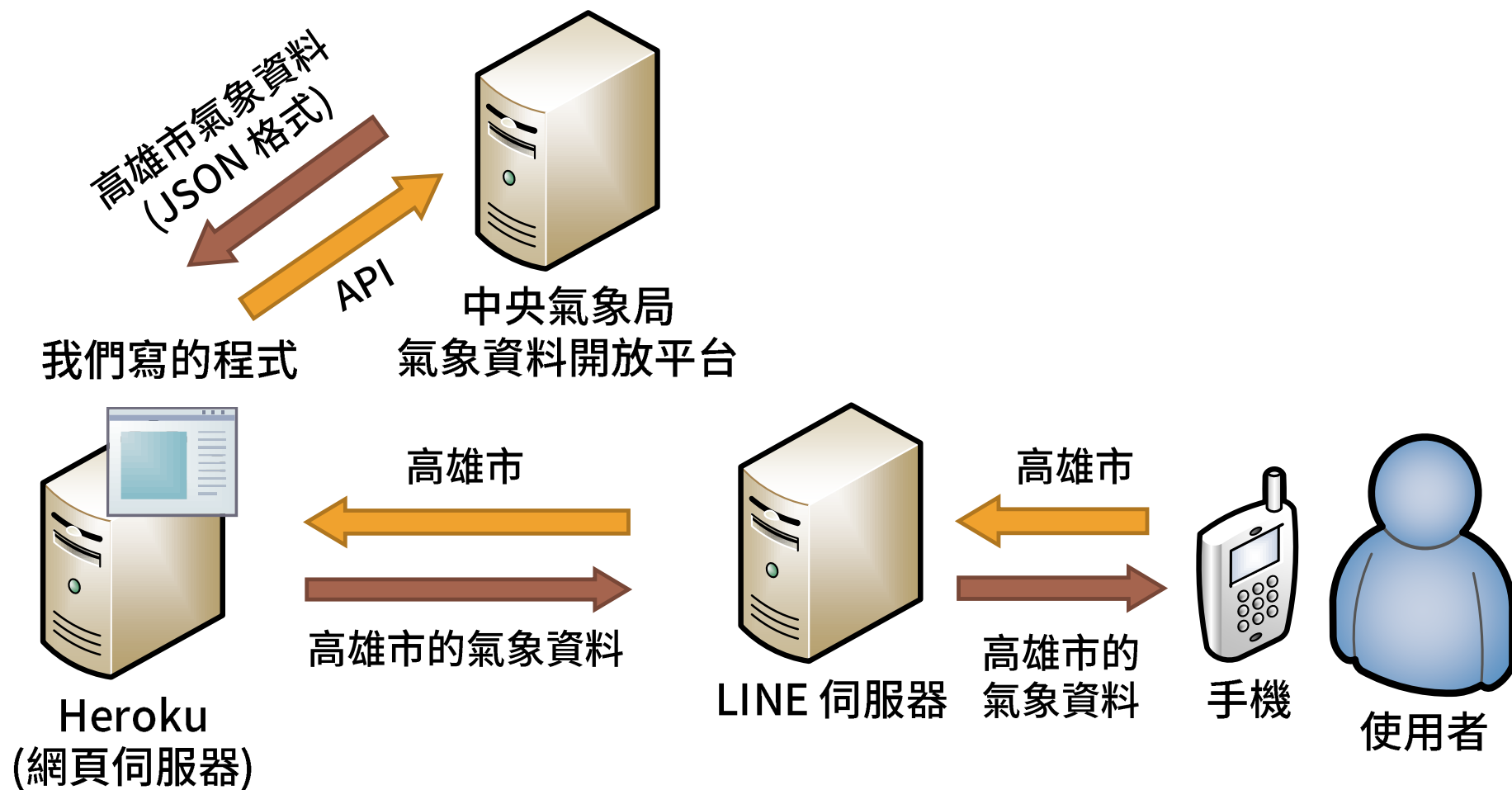
- 簡單修改回覆的訊息，確認分析沒有錯
- 更新到 Heroku 上測試
- 若有錯誤可以使用 Heroku logs 去查閱錯誤訊息

`heroku logs -a` 自己的Heroku app 名稱



```
@handler.add(MessageEvent, message=TextMessage)
def message_text(event):
    custom_reply = "你是說 " + event.message.text + " 嗎?"
    line_bot_api.reply_message(
        event.reply_token,
        TextSendMessage(text=custom_reply)
    )
```


[練習] 天氣報你知 Line Weather

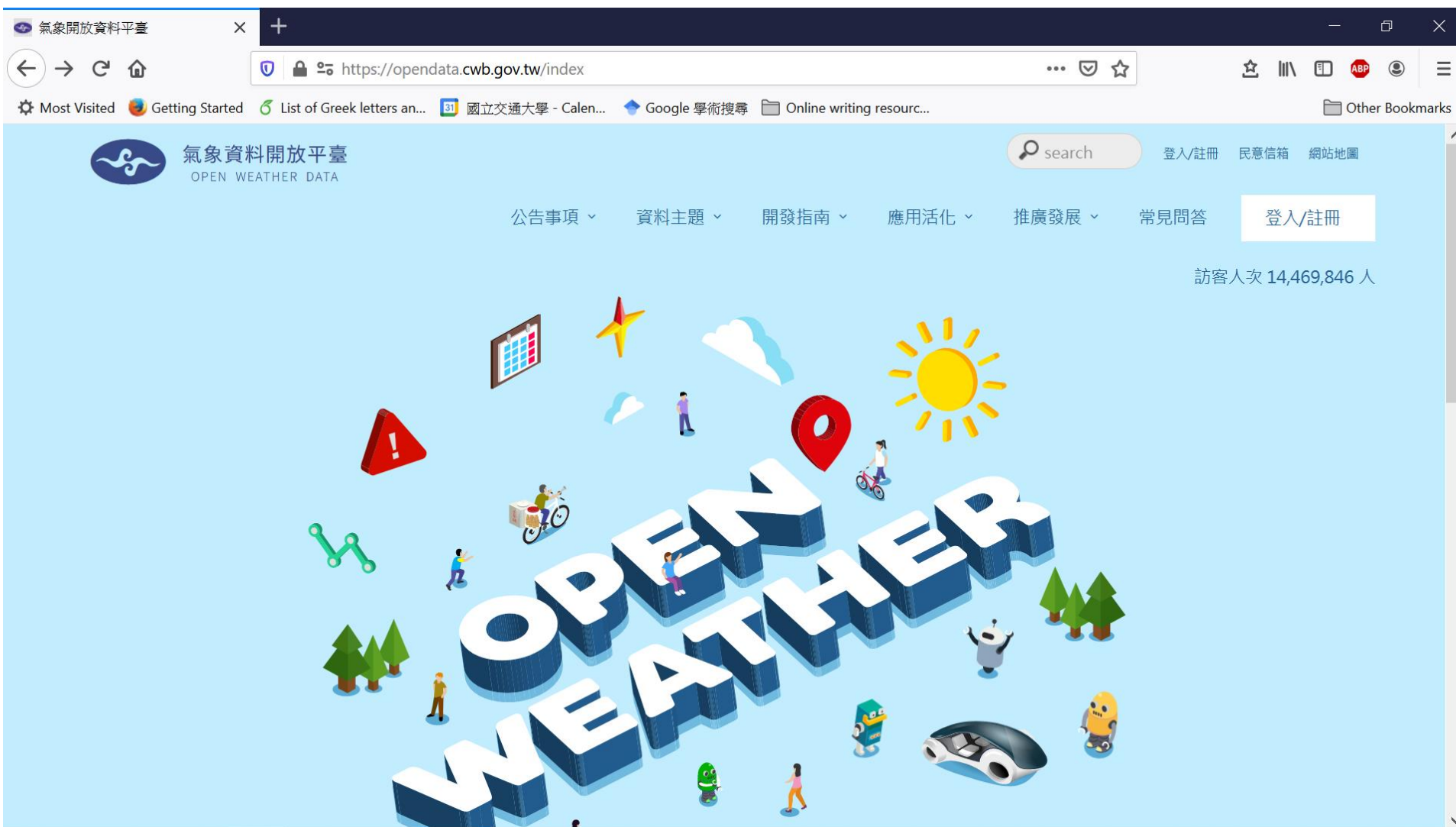


JSON 格式 (JavaScript Object Notation)

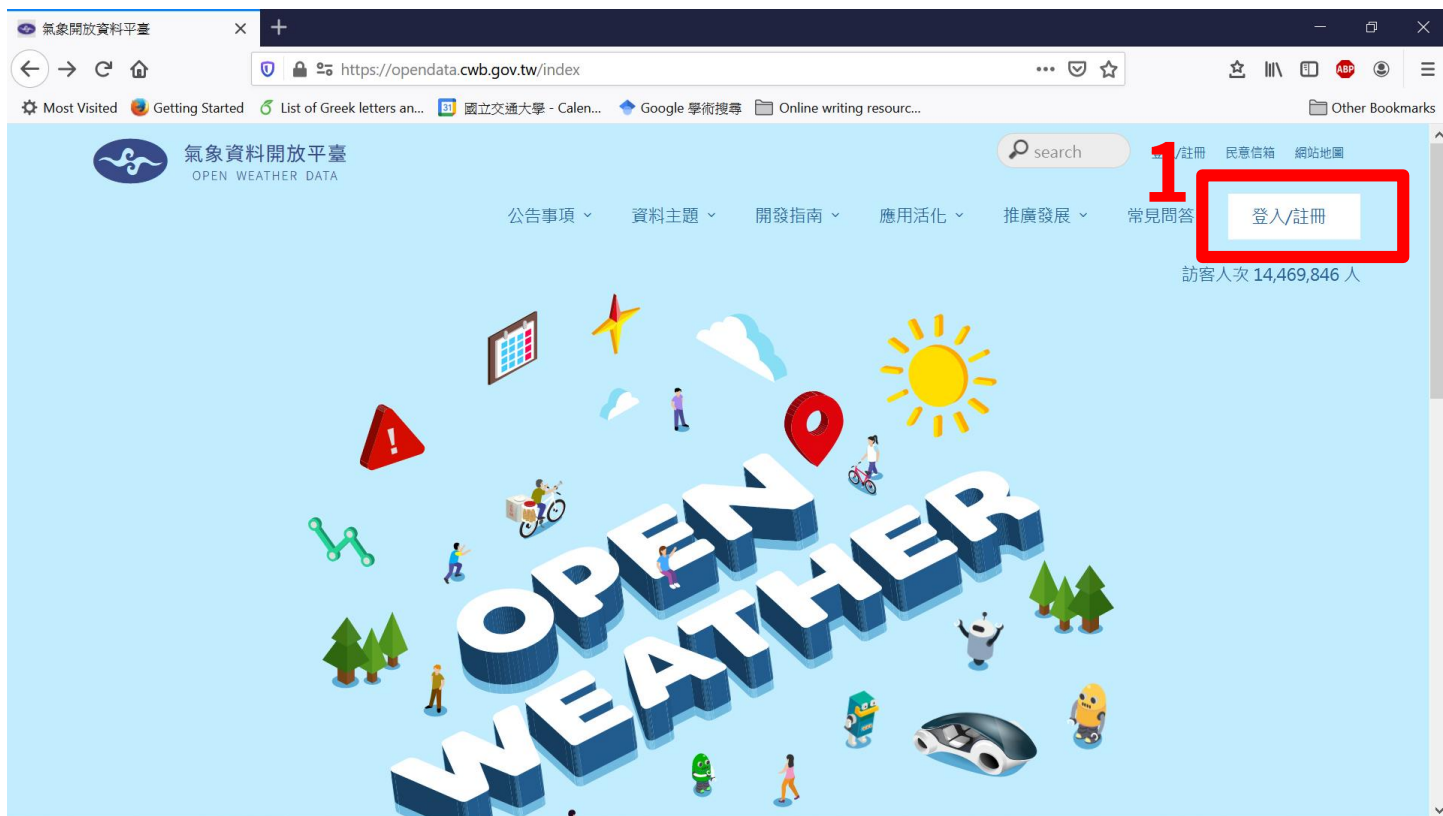
- JavaScript 中用來呈現物件用的格式
- 常被用作網站間資料傳輸和呈現的格式
- Python 中有內建的模組 (json) 可以處理 JSON 格式的資料
 - 會將 JSON 格式的資料轉換成 Python 的 Dictionary

```
1 import json
2
3 # JSON 格式資料 (字串)
4 x = '{ "name": "Winnie Sun", "age" : "18"}'
5
6 # JSON 格式資料 -> Python Dictionary
7 y = json.load(x)
8
9 # 即可使用 Python Dictionary 的方式處理資料
10 print(y['name'])
```

中央氣象局 Open Weather Data



使用流程



2. 擇一使用

氣象會員登入

氣象會員登入

其他方式登入

Facebook會員登入

中央氣象局 | 隱私權保護政策 | 資訊安全政策

地址：100006臺北市中正區公園路64號 總機：(02)2349-1000(代表號)

使用流程 – 取得 API 授權碼

The screenshot shows a web browser window with the URL <https://opendata.cwb.gov.tw/user/authkey>. The page title is "中央氣象局-開放資料平臺". The navigation bar includes links for "公告事項", "資料主題", "開發指南", "應用活化", "推廣發展", "常見問答", and "會員資訊". The breadcrumb trail is "首頁 > 會員資訊 > API授權碼". On the left sidebar, "會員基本資料" is selected, and "API授權碼" is highlighted. The main content area is titled "API授權碼" and contains the following text: "本平臺提供透過URL下載檔案以及 RESTful API 資料擷取方法取用資料，惟因本平臺採用會員服務機制，需帶入資料項目代碼以及有效會員之授權碼，方可取得各式開放資料。其中，資料項目代碼可至資料清單列表查詢。"

一、取得授權碼

會員之授權碼可於下方按鈕取得

取得授權碼

二、更新授權碼

一旦更新授權碼後，舊的授權碼將永久失效，並且更新授權碼後七日內無法再進行更新。

更新授權碼

取得授權碼

中央氣象局 | 隱私權保護政策 | 資訊安全政策

地址：100006臺北市中正區公園路64號 總機：(02)2349-1000(代表號)

產生擷取 API

- 步驟1. 到中央氣象局的資料擷取 API 頁面: <https://opendata.cwb.gov.tw/dist/opendata-swagger.html>
- 步驟2. 選擇想要擷取的資料種類



產生擷取 API

- 步驟 3. 按下「Try it out」按鈕
- 步驟 4. 填入所要求之資料欄位
- 步驟 5. 按下「Execute」按鈕

可參照「timeTo」之格式，若使用「startTime」，則參數「timeTo」的篩選資料則會失效，預設為全部回傳

timeTo - 時間區段，根據內容可篩選需要之時間

Execute

Responses Response content type **application/json**

Code	Description
200	OK

Opendata API 線上說明文件

https://opendata.cwb.gov.tw/dist/

Most Visited Getting Started List of Greek letters an... Other Bookmarks

預報

GET /v1/rest/datastore/F-C0032-001 一般天氣預報-今明 36 小時天氣預報

臺灣各縣市天氣預報資料及國際都市天氣預報

Parameters

3 **Try it out**

Name	Description
Authorization * required	氣象開放資料平台會員授權碼
string (query)	Authorization - 氣象開放資料平台會員授權碼
limit	限制最多回傳的資料，預設為回傳全部筆數
number (\$int) (query)	limit - 限制最多回傳的資料，預設為回傳全部
offset	指定從第幾筆後開始回傳，預設為第 0 筆開始回傳
number (\$int) (query)	offset - 指定從第幾筆後開始回傳，預設為第 0
format	回傳資料格式，預設為 json 格式

解讀擷取 API

API

送該API會得到的
回傳資料範例

The screenshot shows a web browser window with the URL `https://opendata.cwb.gov.tw/dist/`. The page title is "Opendata API 線上說明文件". The browser's address bar shows the URL and several icons. Below the address bar, there are tabs for "Most Visited", "Getting Started", and "List of Greek letters an...". The main content area is titled "Responses" and shows the "Response content type" as "application/json".

The "Curl" section displays the following command:

```
curl -X GET "https://opendata.cwb.gov.tw/api/v1/rest/datastore/F-C0032-001?Authorization=CWB-FFCD9908-C3B6-429F-93AA-A50B235ADC9E&sort=time" -H "accept: application/json"
```

The "Request URL" section, highlighted with a red box, displays the following URL:

```
https://opendata.cwb.gov.tw/api/v1/rest/datastore/F-C0032-001?Authorization=CWB-FFCD9908-C3B6-429F-93AA-A50B235ADC9E&sort=time
```

The "Server response" section shows a status code of 200. The "Response body" section, also highlighted with a red box, displays the following JSON data:

```
{
  "success": "true",
  "result": {
    "resource_id": "F-C0032-001",
    "fields": [
      {
        "id": "datasetDescription",
        "type": "String"
      },
      {
        "id": "locationName",
        "type": "String"
      },
      {
        "id": "parameterName",
        "type": "String"
      }
    ]
  }
}
```


測試資料讀取程式

- 將所得之回傳資料範例做為測試資料，存為檔案 `weather.json`
- 編寫程式來讀取該檔案，檢查是否能得到所期望的資料

測試資料讀取程式

```
2 import json
3
4 # 取得 weather.json 內容
5 f = open('weather.json', encoding="utf-8")
6
7 # JSON 格式資料 -> Python Dictionary
8 data = json.load(f)
9
10 # 取得某縣市當前天氣資料
11 target = '新竹市'
12 res = ''
13 for city in data['records']['location']:
14     if city['locationName'] == target:
15         for element in city['weatherElement']:
16             if element['elementName'] == 'Wx':
17                 res = element['time'][0]['parameter']['parameterName']
18             break
19
20 print('目前天氣:', res)
```

將資料來源從檔案改為中央氣象局

填入API
送API
取得回傳的
JSON資料

```
2 import json
3 import requests
4
5 # 透過中央氣象局 API 取得天氣資料
6 api_url =
  'https://opendata.cwb.gov.tw/api/v1/rest/datastore/F-C0032-001?Authorizati
  5ADC9E&sort=time'
7 resp = requests.get(api_url)
8
9 # 如果擷取資料成功再做解析
10 if resp.status_code == 200:
11     # JSON 格式資料 -> Python Dictionary (兩種方法，擇一使用)
12     # data = resp.json()
13     data = json.loads(resp.content)
14
15     # 取得某縣市當前天氣資料
16     target = '新竹市'
17     res = ''
18     for city in data['records']['location']:
19         if city['locationName'] == target:
20             for element in city['weatherElement']:
21                 if element['elementName'] == 'Wx':
22                     res = element['time'][0]['parameter']['parameterName']
23             break
24
25     print('目前天氣:', res)
26 else:
27     print(resp.status_code)
```

修改 LINE BOT 程式

- 導入所需函式庫
- 修改 message_text 的內容
- 更新到 Heroku 上

```
69 @handler.add(MessageEvent, message=TextMessage)
70 def message_text(event):
71
72     # 透過中央氣象局 API 取得天氣資料
73     api_url = 'https://opendata.cwb.gov.tw/api/v1/rest/datastore/F-C0032-001?A'
74     resp = requests.get(api_url)
75
76     # 如果擷取資料成功再做解析
77     if resp.status_code == 200:
78         # JSON 格式資料 -> Python Dictionary (兩種方法，擇一使用)
79         # data = resp.json()
80         data = json.loads(resp.content)
81
82         # 取得某縣市當前天氣資料
83         target = event.message.text
84         res = ''
85         for city in data['records']['location']:
86             if city['locationName'] == target:
87                 for element in city['weatherElement']:
88                     if element['elementName'] == 'Wx':
89                         res = element['time'][0]['parameter']['parameterName']
90                     break
91             if res != '':
92                 custom_reply = "目前{}的天氣: {}!".format(target, res)
93             else:
94                 custom_reply = "查無資料!"
95         else:
96             custom_reply = "查無資料!"
97
98     # 回覆
99     line_bot_api.reply_message(
100         event.reply_token,
101         TextSendMessage(text=custom_reply)
102     )
```

執行結果

