網路程式設計
Web API

Instructor: 馬豪尚

RESTful WebAPI

- REST → Representational State Transfer
  - 表現層狀態轉換，是一種應用於全球資訊網軟體的架構
  - 它定義了幾項基本原則和架構，並非是一種協定或標準
    - 資源（Resources）
    - 表現層（Repersontation）
    - 狀態轉換（State Transfer）
  - 只要一個WebAPI的設計符合這些理念就稱為是RESTful WebAPI

RESTful WebAPI

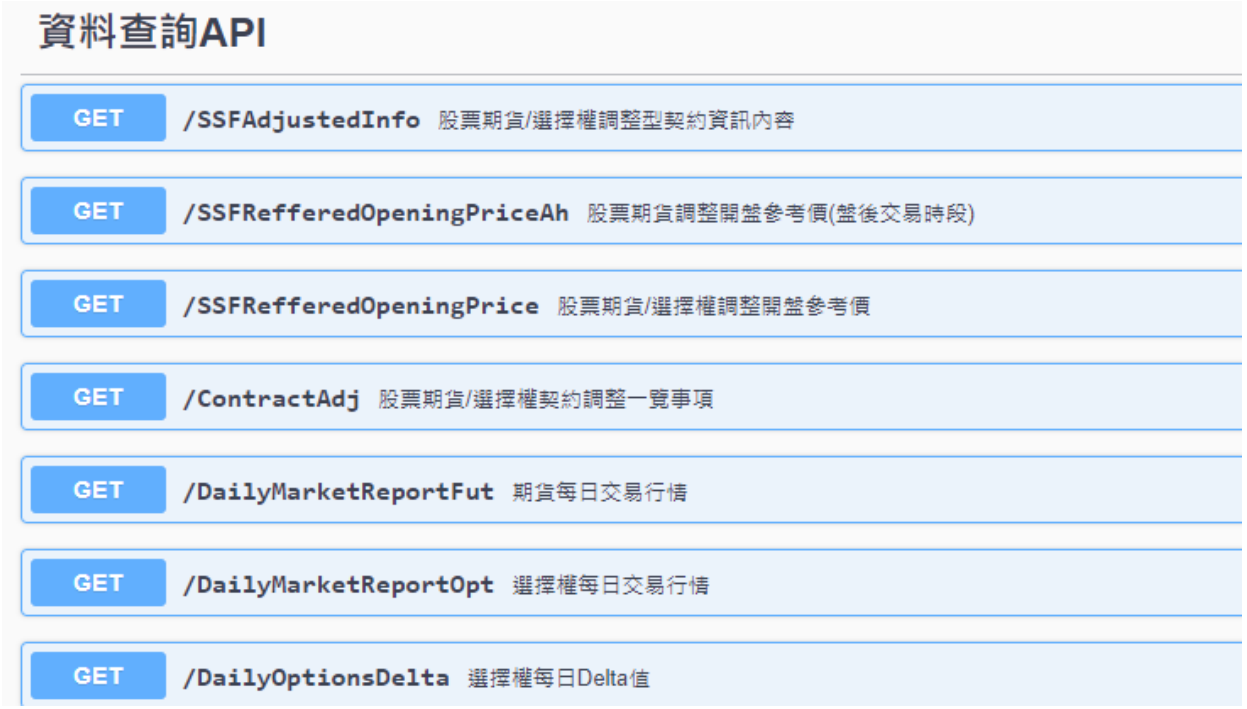
- 資源→ 指的是網路上的一個實體，或是一個具體的訊息
  - 可以是一段文字、圖片、歌曲或是服務
  - 可以透過統一資源標識符（URL）指向資源並取得資源
  - 每一種資源對應一個特定的URL。
- 表現層 → 定義了資源的呈現方式
  - 一段字串可以使用txt格式表現
  - 頁可以使用HTML格式來呈現，或XML、JSON的格式
- 狀態轉換 → 代表的是客戶端與伺服器的一個互動中的資料與狀態的變化
  - 使用者請求與伺服器回應的狀態(例如伺服器回應的狀態碼)

RESTful WebAPI 總結要點

- 這個網路服務提供使用者發出以URL存取和操作網路資源的請求
  - 資源是由URL來進行指定
  - 資源的操作可包括：取得、建立、修改以及刪除，對應到HTTP Request Method中的GET、POST、PUT與DELETE等方法
- 通過定義好的表現形式來操作資源，取決於不同的讀取者
- 以無狀態的方式回應使用者的請求，無狀態是指伺服器獨立於所有之前的請求，完成每個用戶端請求的通訊方法

RESTful WebAPI實例

- 台灣期貨交易所API
- 台灣證券交易所API
- 台灣氣象資料開放平台API



自己建立一個RESTful架構的 WebAPI

- 架設伺服器
  - Flask 函式庫 (模組) 是一個輕量級的 Web 應用框架
  - 提供了包括路由 (Routes)、樣板 (templates) 和權限 (authorization) 等功能
  - 可以提供架設網站或建構網路服務
  - 用本機當作伺服器
- 載入模組
  - pip install Flask
  - from flask import Flask

Flask建立一個網頁服務

- 建立一個Flask 物件
  - app = Flask(__name__)
  - __name__表示目前執行的程式
- 建立一個路由 (Routes)，定義主網域 / 請求資源的url路徑
  - @app.route("/", methods)
  - "/" 代表請求資源的url路徑，沒有指定代表主網域的路徑
  - methods為指定請求操作資源的方式(get、post等)

Flask建立一個網頁服務

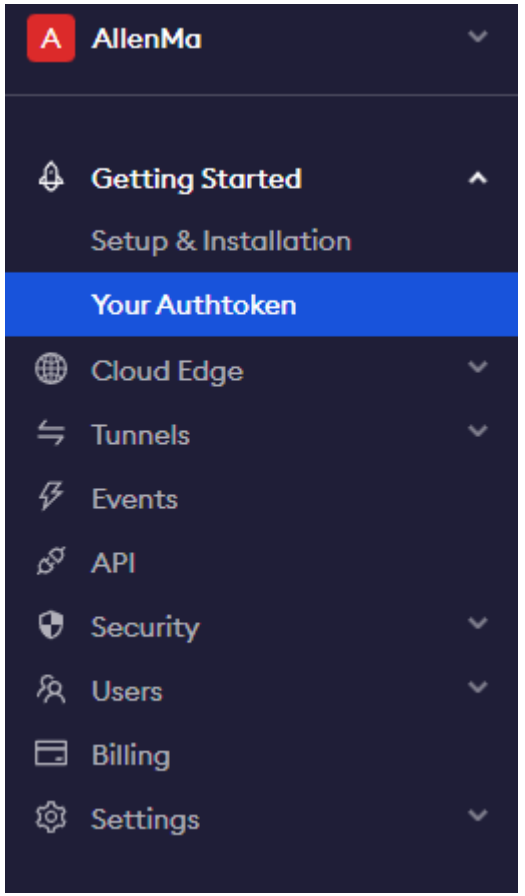
- 定義發出請求後執行的函式
  - def function():
  - 在函式需要中指定回傳給使用者的內容
    - return 內容
- 執行這個flask物件 → 啟動網路服務
  - app.run()
  - 設定連線埠號 port
    - app.run(host="0.0.0.0", port=5555)
    - 預設是127.0.0.1, port=5000

Ngrok服務

- 在開發網頁應用時，通常是使用本機的伺服器，無法真正在外界進行測試
- ngrok 服務，能夠將本機環境對應到一個 ngrok 網址
- 這個網址能夠在整個網際網路中提供服務，就能真正從外部連結進行測試

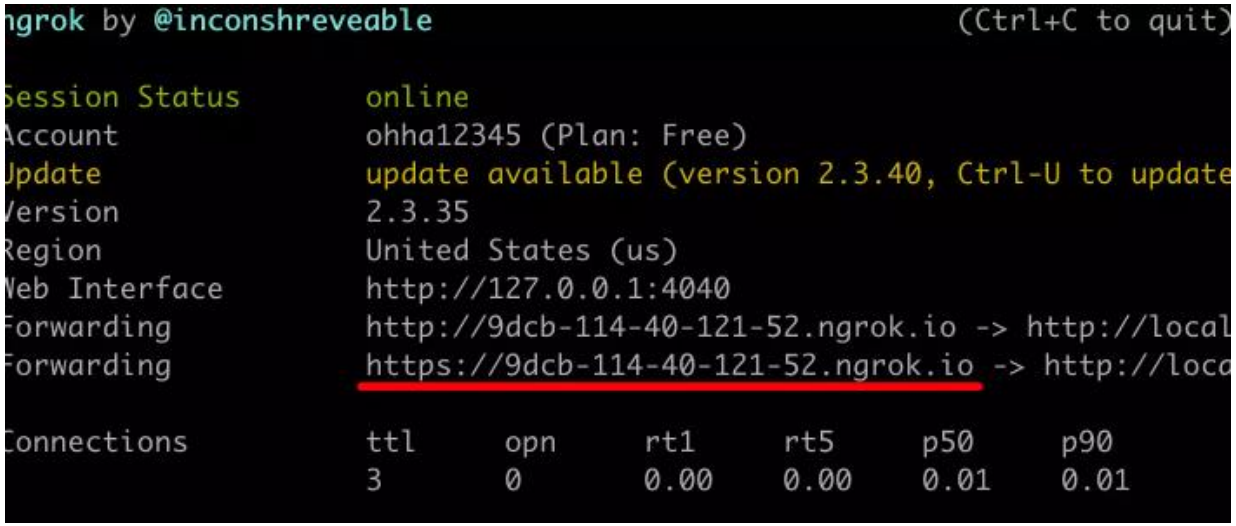
Ngrok服務

- 註冊 ngrok 取得 token
  - https://ngrok.com/
- 串接 ngrok 服務所使用的 token
  - Your Authtoken
- 本機環境使用 ngrok
  - 下載符合作業系統的ngork並安裝



Ngrok服務

- 安裝後，開啟終端機，註冊ngrok的 token
  - ngrok authtoken <token>
- 啟用ngrok的服務
  - ngrok http <port>
  - <port>為本機伺服器服務的port
  - 使用 Flask 建構的服務，port 預設是 5000



在colab上使用ngrok

- 如果你使用colab來創建flask等於是將colab的虛擬機當作你的本機，但是這個本機又無法與外界溝通
- 我們要在colab的虛擬機上安裝ngrok才可以將本機服務公開到網際網路中

在colab上安裝ngrok

- 連接到google雲端硬碟
- 建立並設定ngrok要安裝的路徑

```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)

!mkdir -p /drive
!mount --bind /content/drive/MyDrive /drive
!mkdir -p /drive/ngrok-ssh
```

在colab上安裝ngrok

- 下載ngrok並解壓縮ngrok到colab虛擬機上
- 註冊ngrok的 token

```
!mkdir -p /drive/ngrok-ssh
%cd /drive/ngrok-ssh
!wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip -
O ngrok-stable-linux-amd64.zip
!unzip -u ngrok-stable-linux-amd64.zip
!cp /drive/ngrok-ssh/ngrok /ngrok
!chmod +x /ngrok

!/ngrok authtoken <token>
```

在colab上使用ngrok

- 安裝 flask_ngrok 函式庫
- 載入flask_ngrok
- 使用run_with_ngrok(app)

```
from flask import Flask
from flask_ngrok import run_with_ngrok

app = Flask(__name__)
run_with_ngrok(app)
@app.route("/")
def home():
    return f"<h1>hello world</h1>"

app.run()
```

Ngrok注意事項

- 使用本機環境開發時，如果電腦關機（或網路斷線），服務也會跟著中斷。
- ngrok 免費版同時間內只能串連一個服務。
- Colab 程式有運行時間限制，無法作為正式的伺服器使用。

Flask 路由設定

- @app.route，可透過 method 設定 GET 或 POST
- 預設是GET，GET 方法可以透過網址進行溝通，也就是透過網址列傳送所有的參數內容
  - @app.route("/", methods=['GET'])
  - 在所對應執行的函式中取得參數
    - request.args

Example

用url傳遞參數→https://127.0.0.1:5000?name=abcd&age=18

request.args→會得到{[('name', 'abcd'), ('age', '18')]}

取得name的參數request.args.get('name')

Flask 路由設定

- POST 方法則是將資料放在 message-body 進行傳送，無法單純透過網址列傳送
  - @app.route("/", methods=['POST'])
  - 在所對應執行的函式中取得參數
    - request.form

Example

使用者請求時夾帶post的參數

data = {'name': 'abcd', 'age': '18'}

requests.post('http://127.0.0.1:5000/', data=data)

網路服務取得使用者傳遞的參數

request.form →會得到{[('name', 'abcd'), ('age', '18')]}

取得name的參數request.form.get('name')

定義回傳資料格式

- 為了符合API的架構，我們需要定義回傳資料的格式
  - String、List、Tuple、JSON、XML等
- 將回傳資料包裝成該格式並定義schema
  - Flask內有提供包裝成JSON的函式
    - jsonify(key=value)
  - 使用json模組的dumps函式將字典轉json格式
    - json.dumps(dict)

練習

- 用flask和ngrok在colab建立一個網路服務
- 在伺服器上放置一個資料庫
  - Stock1101.csv
- 定義一個api
  - 讓使用者取得某一年份的資料
    - 例如: 使用者想取得民國100年的資料，傳入參數為year=100
    - request('ngrok產生的網址/api的網址?year=100')
  - 回傳給使用者資料時使用json格式
    - 在資料內的欄位名稱即為json格式裡物件的key
    - 資料的值為json格式裡物件的value
    - 回傳範例: {'成交股數': '4,585,345,607', '成交金額': '170,209,855,530', '成交筆數': '1,342,223', '最高價': '49.45', '最高價日期': '7月22日', '最低價': '29', '最低價日期': '9月26日', '收盤平均價': '36.8'}