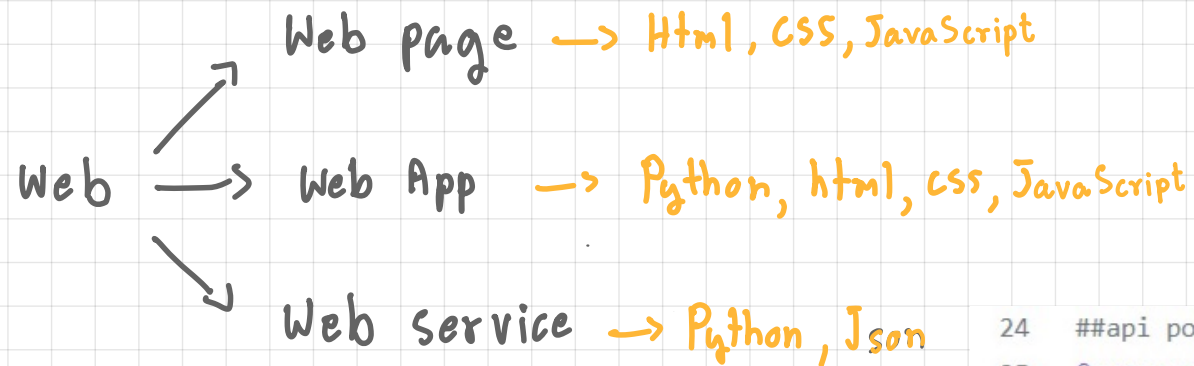


flask



- ระวัง 200, error 404
- pip install flask ท้าหัว
- <http://127.0.0.1:5000> link

```
1 from flask import Flask, request, render_template, make_response
2 import json
3 # import pandas as pd
4
5 app = Flask(__name__)
6
7 @app.route("/")
8 def helloworld():
9     return "Hello, World!"
10
11 @app.route("/name")
12 def hellotohn():
13     return "Hello, Tohn!"
14
15 ##api get
16 @app.route('/getrequest', methods=['GET']) #get?
17 def get_request_detail():
18
19     getmsg = request.args.get('msg1')
20     print(getmsg)
21     return "recieved"
22
```

ฟังก์ชันที่ 1

ฟังก์ชันที่ 2

Method เป็น http protocol

Protocol คือ ข้อ
กำหนดหรือข้อตกลง
ในการสื่อสารระหว่าง
อุปกรณ์ต่างๆ

```
24 ##api post
25 @app.route('/postrequest', methods=['POST'])
26 def post_request_detail():
27     payload = request.data.decode("utf-8")
28     inmessage = json.loads(payload)
29
30     print(inmessage)
31
32
33     json_data = json.dumps({'y': 'received!'})
34     return json_data
35
```

รับของที่ส่งมาให้เรา เช่น ข้อความ

ข้อความ เป็น Json

```
1 import requests
2 import json
3
4 if __name__ == "__main__":
5
6     url = 'http://127.0.0.1:5000/postrequest' #ของเรา
7     myobj = {'message_key': 'message_val'}
8
9     # url = 'http://10.71.160.33:5001/postrequest' #ของอาจารย์
10    # myobj = {'name': 'Athitiya'}
11
12    # url = 'http://192.168.205.233:8080/postrequest' #ของเพื่อนแต่ติดไฟลัวอร์
13    # myobj = {'Hi': 'Wanita'}
14
15    x = requests.post(url, data = json.dumps(myobj))
16
17    print(x.text)
```

GET

เมธอดนี้ Client ทำการร้องขอข้อมูล (Request) เข้ามา โดยสามารถอ่านหรือดูข้อมูลใน Resources นั้นได้อย่างเดียว จะไม่สามารถทำอะไรให้ Resources เกิดการเปลี่ยนแปลงของข้อมูลได้ ถ้ามีการร้องขอข้อมูลเข้ามา แล้ว Match กับ Resources ที่มีอยู่ ตัวเว็บเซิร์ฟเวอร์ก็จะรีเทิร์น STATUS 200 เพื่อบอกว่ามีการเรียกดูข้อมูลใน Resources สำเร็จ หรือถ้าข้อมูลใน Resources นี้ไม่มีหรือถูกลบออกไปแล้ว ก็จะรีเทิร์น STATUS 404 Not Found เป็นต้น

POST

เมธอดนี้ Client สามารถบันทึกหรือทำให้ Resource ของผมเกิดการเปลี่ยนแปลงได้ เช่นเพิ่มข้อมูล ตัวอย่างง่าย ๆ เช่นอย่างการบันทึกข้อมูลในฟอร์ม เมื่อเราเข้าไปเว็บของ Lazada แล้วต้องการสั่งซื้อสินค้า จากนั้นเราก็กรอกข้อมูลการชำระเงินต่าง ๆ นั้นแหละคือเมธอด POST เมื่อข้อมูลที่เรากรอกถูก Submit ตัวเว็บเซิร์ฟเวอร์จะรีเทิร์น STATUS 201 เพื่อบอกว่ามีการสร้างข้อมูลใหม่เข้ามาใน Resources ซึ่งเมื่อสังเกตดูที่ url ที่เว็บเบราว์เซอร์ เราจะไม่เห็นข้อมูลหรือพารามิเตอร์ต่าง ๆ ก็จะถูกซ่อนไว้ซึ่งถ้าเป็นเมธอด GET ข้อมูลเหล่านั้นจะแสดงออกมาให้เห็นหมด ดังนั้นถ้ามีการทำ Transaction ต่าง ๆ ที่ต้องมีการบันทึกข้อมูล จึงควรต้องใช้เมธอด POST

PUT

เมธอดนี้ Clients สามารถแก้ไขหรืออัปเดตข้อมูล ใน Resources ได้ เมื่ออัปเดตข้อมูลสำเร็จ เว็บเซิร์ฟเวอร์ก็จะรีเทิร์น STATUS 200

DELETE

เมธอดนี้ Clients สามารถลบข้อมูลใน Resources ได้ ซึ่งสมมติว่ามีการลบข้อมูลออกไปแล้วโดยเมธอด DELETE นี้ เมื่อเข้ามาที่ url เดิมก็จะพบกับ STATUS 404 Not Found

เมื่อสังเกตให้ดีจะพบว่า Methods ทั้ง 4 ที่ได้อธิบายไปนั้นมันค่อนข้างที่จะสัมพันธ์กันกับ CRUD (Create, Read, Update, Delete) ซึ่งเมธอดที่คุ้นเคยกันดีและถูกใช้บ่อยที่สุดคือ GET และ POST



สู้ๆนะ เป็นกำลังใจ
จ้ายยยย

