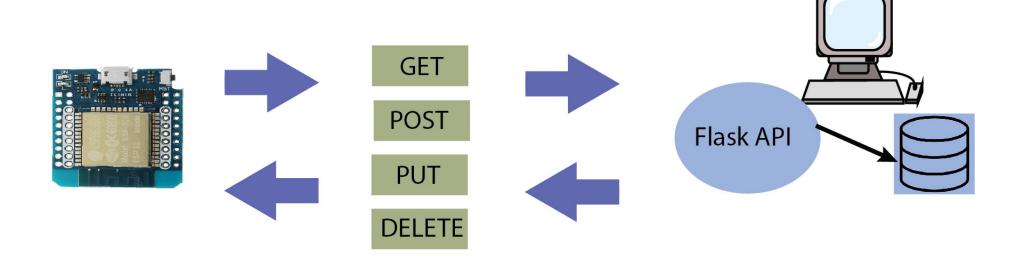
# FLASK API

#### ภาพรวมระบบ

#### เว็บแอปพลิเคชันเฟรมเวิร์กภาษาไพทอน



### การสร้างสภาพแวคล้อมเสมือน (Virtual Environment)

• ติดตั้ง virtualenv

```
PS C:\Users\CK> pip install virtualenv
```

• สร้างโฟลเดอร์สำหรับเวอร์ชวล environment

```
PS C:\Users\CK> mkdir myapi
PS C:\Users\CK> cd myapi
PS C:\Users\CK\myapi> virtualenv venv
```

• เรียกใช้เวอร์ชวล environment และติดตั้ง Flask สังเกตคำว่า (venv) ที่เกิดขึ้นหน้าโฟลเดอร์

```
PS C:\Users\CK\myapi> venv/scripts\activate (venv) PS C:\Users\CK\myapi> pip install Flask
```

### API อย่างง่าย

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def Hello():
    return '<h1>Hello Smart Farm!</h1>'

if __name__ == '__main__':
    app.run()
```

### Run และผลที่ได้

```
(venv) PS C:\Users\CK\myapi> python .\flask1.py
* Serving Flask app "flask1" (lazy loading)
* Environment: production
   WARNING: This is a development server. Do not use it in a
        production deployment.
   Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



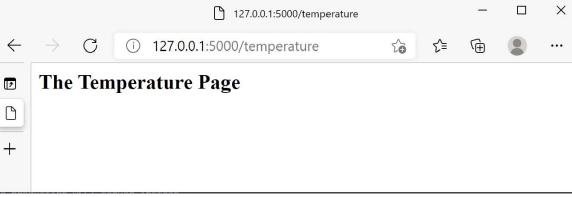
## เพิ่มการเรียกใช้อื่น

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def Hello():
    return '<h1>Hello Smart Farm!</h1>'

@app.route('/temperature')
def temperture():
    return '<h2>The Temperature Page</h2>'

if __name__ == '__main__':
    app.run()
```



## การเชื่อมต่อฐานข้อมูล MySQL

- 1. ติดตั้ง MySQL และสร้างเทเบิลที่ต้องการจัดเก็บข้อมูล
- 2. ติดตั้ง Flask พร้อมโมดูลที่จำเป็น เช่น flask\_mysqldb สำหรับการเชื่อมต่อ MySQL เป็นต้น
- 3. โปรแกรมฟังก์ชันต่าง ๆ ที่ต้องการบน Flask เช่น การเรียกข้อมูล การเพิ่มข้อมูล และอื่น ๆ
- 4. โปรแกรมไมโครไพทอนบน ESP32 พร้อมกำหนดไอพีแอดเดรสของเครื่องที่รัน Flask ให้ถูกต้อง

PS C:\Users\CK> cd .\myapi\
PS C:\Users\CK\myapi> .\venv\Scripts\activate
(venv) PS C:\Users\CK\myapi> pip install flask mysqldb

## สร้างฐานข้อมูล

	Table Name:	sensor_dht					Schema:		myfarm				
	Charset/Collation:	utf8	v utf8_bi	utf8_bin ∨			igine:	InnoDB				~	
	Comments:											<b>^</b>	
Column Name		Datatype	PK	NN	UQ	В	UN	ZF	AI	G	Default/Expression		
💡 id		INT(11)		~					~				
update_tim	ie	DATETIME		~									
temperatur	e	DOUBLE		~									
humidity		DOUBLE		~									

### ติดตั้ง Flask และโมดูลที่จำเป็น

- lask\_mysqldb สำหรับการใช้งานร่วมกับ MySQL
- lask-cors สำหรับรองรับการสื่อสารจากโมดูล ESP32 ซึ่งเป็นการจัดการภายใต้Cross Origin Resource Sharing (CORS) อนุญาตให้สื่อสารข้ามโดเมนได้
- cryptography สำหรับการกำหนดเซอร์ติฟิเกต (Certificate) สำหรับการสื่อสารผ่าน โปรโตคอล HTTPS

#### โปรแกรม Insert Database

```
from flask_mysqldb import MySQL
from flask_cors import CORS
from datetime import datetime
app = Flask(__name__)
CORS(app)
# MySQL Config
app.config['MYSQL HOST'] = 'localhost' #server location
app.config['MYSQL USER'] = 'mysql username'
app.config['MYSQL_PASSWORD'] = 'mysql_password'
app.config['MYSQL_DB'] = 'myfarm'
#app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
# init MySQL
mysql = MySQL(app)
@app.route('/sensor_dht', methods=['POST'])
def add_dht():
    try:
        data = request.json
        temperature = data['temp']
        humidity = data['humi']
        current_time = datetime.now()
        date time = current time.strftime("%Y-%m-%d %H:%M:%S")
        cur = mysql.connection.cursor()
        cur.execute("INSERT INTO sensor_dht (update_time, temperature, humidity) VALUES (%s, %s, %s)",
                 (date_time, temperature, humidity))
        mysql.connection.commit()
        cur.close()
        resp = jsonify(message="Data added successfully!")
        resp.status code = 200
        return resp
    except Exception as e:
        print(e)
        return jsonify(message="adding data error"), 500
if __name__ == '__main__':
    app.run(debug=True, host="0.0.0.0", ssl context='adhoc')
```

#### โปรแกรมไมโครไพทอน

```
from machine import Pin
import urequests as requests
import random, ujson, utime
import dht
import gc
url = "https://192.168.0.76:5000/sensor_dht"
sensor = dht.DHT22(Pin(25))
while True:
    sensor.measure()
    temp = sensor.temperature()
    humi = sensor.humidity()
    print("temperature: %3.1f°C" % temp)
    print("humidity: %3.1f%%" % humi)
    post = ujson.dumps({"temp": temp, "humi": humi})
    try:
        res = requests.post(url, headers = {'content-type': 'application/json'}, data=post)
        res.close() # ปิดการเชื่อมต่อ
        if res.status_code == 200:
            print("Request successful!")
        else:
            print("Request failed with status code {}".format(res.status_code))
    except:
        print("Error: Could not establish connection.")
    gc.collect()
    utime.sleep(10)
```