

# Communication Networks Lab

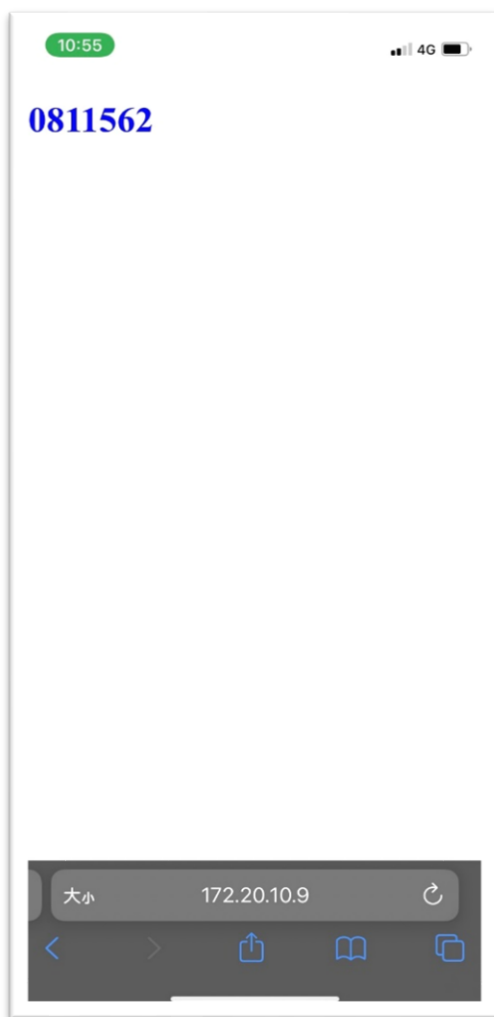
## Topic Sensor Network-Lab3

Q1: 使用GET Method 將自己的學號顯示出來

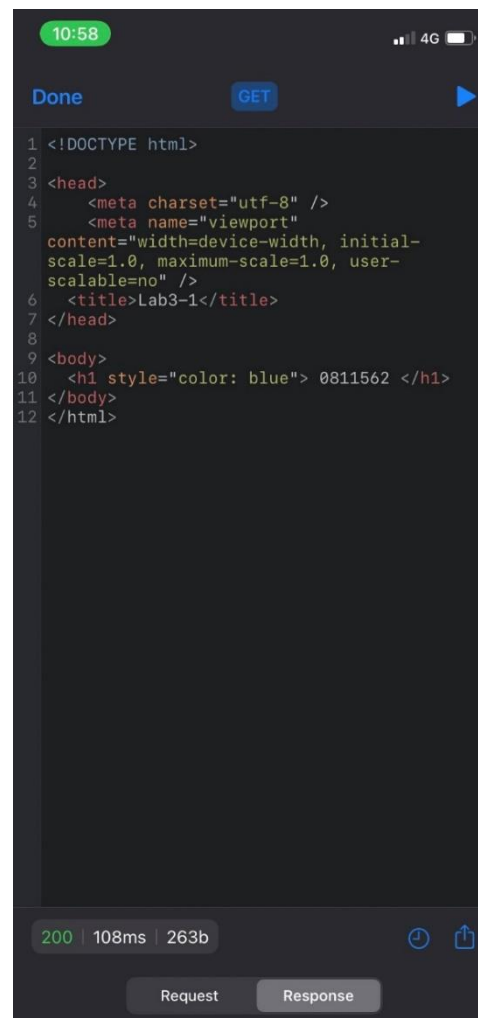
- ◆ 根據助教提供的 `Lab3_1.py`，並自行完成
- ◆ 請在自己手機端利用網頁瀏覽器顯示出 HTTP Request 結果 (即學號)
- ◆ 再使用手機 APP 顯示出 HTTP Request 結果

```
pi@raspberrypi:~/lab3 $ python Lab3_1.py
* Serving Flask app "Lab3_1" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:9808/ (Press CTRL+C to quit)
172.20.10.1 - - [10/Nov/2022 02:54:37] "GET / HTTP/1.1" 200 -
```

網頁瀏覽器結果:



手機 App 結果:



### Lab3-1.py

```
1  # -*- coding: utf-8 -*-
2  from flask import Flask, render_template
3
4  app = Flask(__name__, template_folder = 'templates')
5
6  @app.route('/')
7  def index():
8      # 此處會用到的 function
9      # 1. render_template('index.html', ID = ....)
10     ''' start of you code '''
11     return render_template('index.html', str('0811562'))
12     ''' end of you code '''
13
14 if __name__ == '__main__':
15     app.run(host = '0.0.0.0', port = 9808, debug = False)
```

### template/index.html

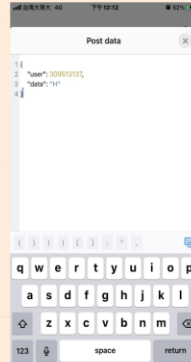
```
1  <!DOCTYPE html>
2
3  <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6      <title>Lab3-1</title>
7  </head>
8
9  <body>
10     <h1 style="color: blue"> {{ID}} </h1>
11 </body>
12 </html>
```

將我的學號置換到 template 中 index.html 中 ID 這個變數的位置，而後開始執行，等待其他裝置連接到這個 server。透過手機網頁瀏覽器及 API tester 這個 app 發送 get 的請求，以取得內部的資料，並不會更動到內部的資源。

## Q2: 完成一個能查閱 RPi 溫溼度模組數值 REST API

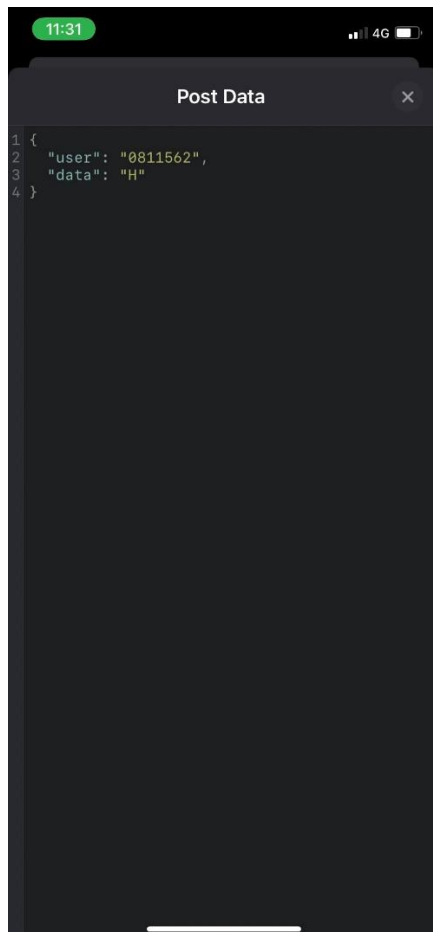
- ◆ 根據助教提供的 [Lab3-2.py](#)，並自行完成
- ◆ 使用者須使用 POST Request 提供使用者的學號與查詢溫度還是濕度兩個 Json fields

```
{  
  
  "user": "309513137",  
  "data": "H"  
}
```



※本次實驗使用 private IP 來讓手機與 Flask Web Service 互相連線，所以需要讓手機與 Rpi 板同時連接同一個無線網路分享器

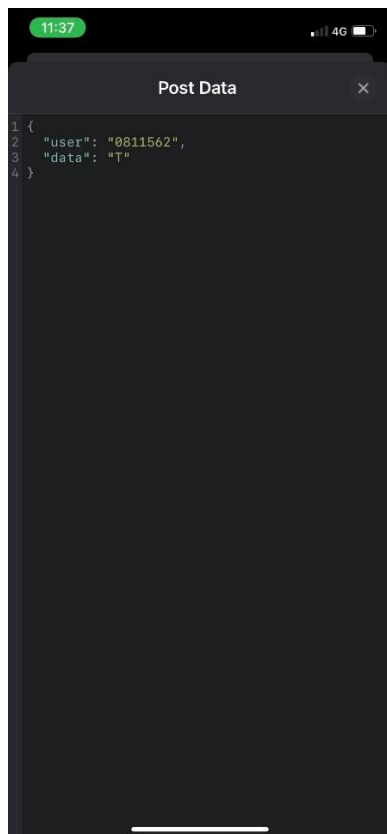
### Post request for Humidity:



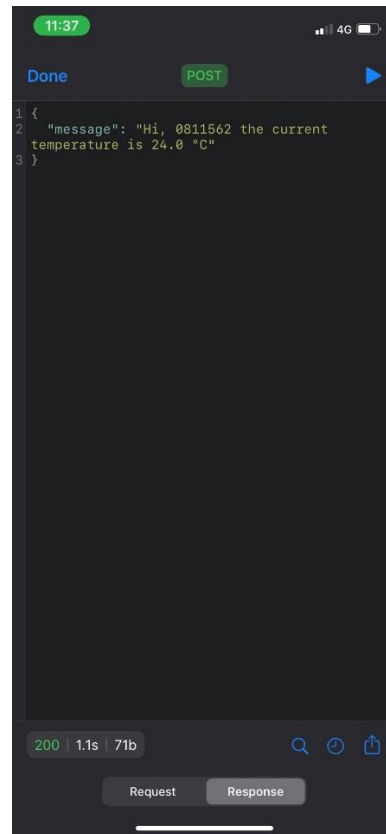
### Response for humidity:



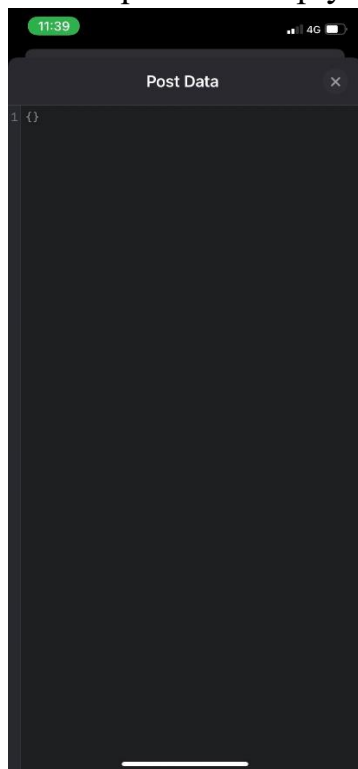
Post request for Temperature:



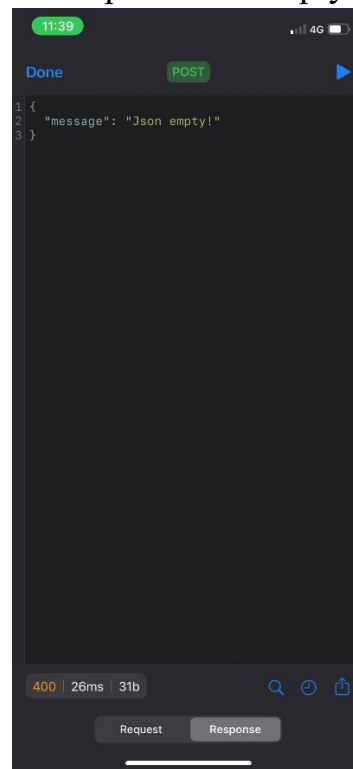
Response for temperature:



Post request for empty Json:



Response for empty Json:



172.20.10.1 - - [10/Nov/2022 03:39:27] "POST / HTTP/1.1" 400 -

## Lab3-2.py

Deal with empty Json.

```
if len(json) == 0:
    # 若使用者傳入的 Json 為空白的話，需回傳 HTTP Status Code = 400 的網頁內容
    # 此處會用到的 functions
    # 1. jsonify(message = ....)
    # 2. make_response(..., 400)

    ''' start of you code '''
    message = jsonify(message = 'Json empty!')
    response = make_response(message, 400)
    return response

    ''' end of you code '''
```

Return the response of the request:

```
if data_type == 'T':
    ''' start of you code '''
    response = jsonify(message="Hi, 0811562 the current temperature is " + temperature + " °C")
    return response

    ''' end of you code '''

elif data_type == 'H':
    ''' start of you code '''
    response = jsonify(message="Hi, 0811562 the current humidity is " + humidity + " %")
    return response

    ''' end of you code '''

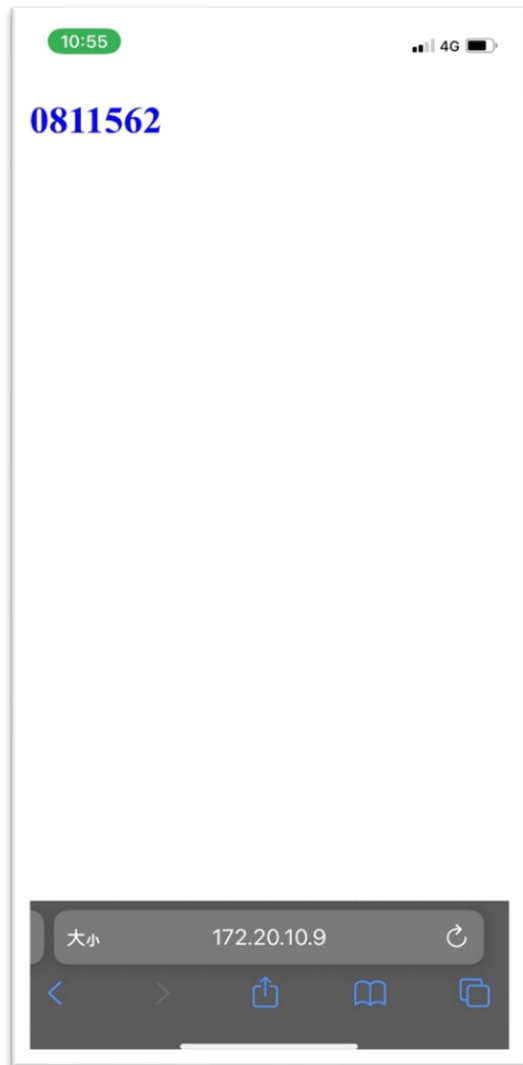
else:
    ''' start of you code '''
    response = make_response(jsonify(message= 'Error'), 400)
    return response

    ''' end of you code '''
```

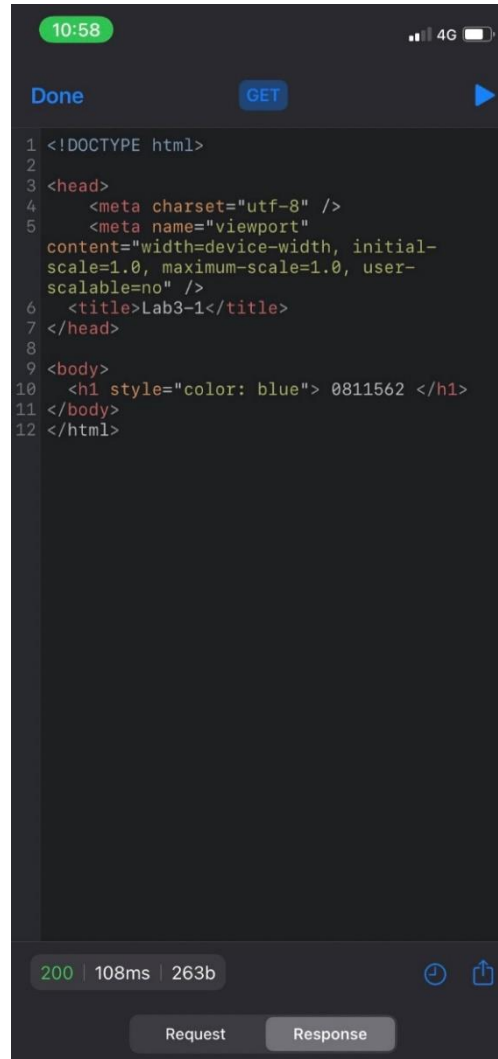
首先，透過 Adafruit\_DHT.read\_retry() 將 sensor 量測到的 data 讀入程式中，然後先判斷 user 傳送的 Json fields 為何。倘若為空的 Json，就回傳 Status Code=400 的網頁內容，裡面寫著 Json empty。倘若為 H，則將溫溼度感測器量測到的濕度回傳給使用者。倘若為 T，則將溫溼度感測器量測到的溫度回傳給使用者，倘若都不是，則回傳 Status Code = 400，裡面寫著 error，以告知使用者沒有傳入適當的變數。

1. 附上本次實驗 Q1 與 Q2 的手機瀏覽器、APP 內容的截圖，並說明為何在 Q1 中兩種 Demo 方式出來的文字內容不太一樣

網頁瀏覽器結果:



手機 App 結果:



透過網頁瀏覽器看到的結果是只有顯示學號，原因是 index.html 會被瀏覽器當成首頁，因此展示的會是 html 在網頁上運行的成果，也就是有著藍色顏色的學號。然而，使用手機 API tester 接收到的 response 為 raw data，並沒有網站去運行這個資料，因此展示出來的內容為 raw index.html 的檔案。

2. 本次實驗內容 Q1，在手機端在進行 HTTP Request 時，Flask Web Service 會產生下列的訊息，請問以紅色方框標示的數字代表著什麼意思？

```
* Serving Flask app 'lab1_1' (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 all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.0.11:9808/ (Press CTRL+C to quit)
192.168.0.11 - - [26/Feb/2022 13:11:55] "GET / HTTP/1.1" 200 -
192.168.0.11 - - [26/Feb/2022 13:11:58] "GET / HTTP/1.1" 200 -
192.168.0.11 - - [26/Feb/2022 13:11:59] "GET / HTTP/1.1" 200 -
```

紅色方框標示的數字代表的是 Status code。當 status code 為 200 時，表示伺服器已經成功接收到請求。常見的 Status code 為 400 或 404，表示 client 端發送了錯誤的請求，因此在 Q2 中，倘若使用者不是傳入 T 或是 H 的變數，我們將使用者帶向 Status code = 400 的網頁內容。

詳細的 Status code 如下：

Informational responses (100 – 199)

Successful responses (200 – 299)

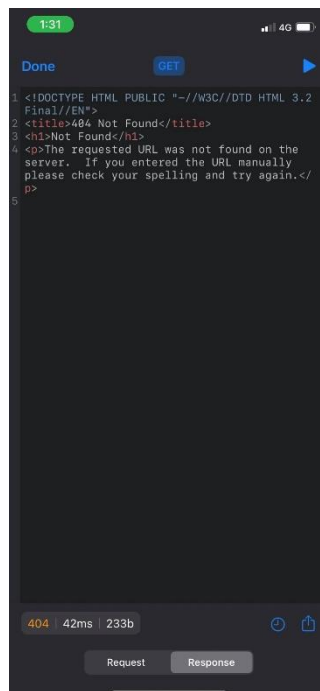
Redirection messages (300 – 399)

Client error responses (400 – 499)

Server error responses (500 – 599)

3. 若將實驗 Q1 中手機瀏覽器的網址內容更改成 **192.168.xxx.xxx:9808/pages** 並提交 GET Request 至 Flask Web Service 會產生什麼樣的訊息？原因是為何呢？(請附上網頁瀏覽器與 Flask Web Service 訊息截圖)

```
172.20.10.1 - - [10/Nov/2022 17:30:53] "GET /pages HTTP/1.1" 404 -
172.20.10.1 - - [10/Nov/2022 17:31:36] "GET /pages HTTP/1.1" 404 -
```



可以觀察到 server 發出的 status code 為 404，表示 client 端發出了無效的請求。因為多加上了 /pages，因此無法正常顯示畫面，被帶到顯示 status code = 404 的 Not found 畫面。

倘若是使用 https 開頭的 url，則 server 端會出現許多亂碼，並且 status code 為 400，將使用著帶到 Status code = 400 的網頁內容。

Using https:

```
pi@raspberrypi:~/lab3 $ python Lab3_1.py
* Serving Flask app "Lab3_1" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:9808/ (Press CTRL+C to quit)
172.20.10.1 - - [10/Nov/2022 03:22:43] code 400, message Bad request version ('\xc6o&\xc7\x042\xf9\xc1\x8f\xeb\x05A\x84')
172.20.10.1 - - [10/Nov/2022 03:22:43] "GET / HTTP/1.1" 400 -
172.20.10.1 - - [10/Nov/2022 03:22:43] code 400, message Bad request syntax ('\x16\x03\x01\x02\x00\x01\x00\x01\xfc\x03\x03\x05,\xc5,\xc6\x07\xfbNy]\x01\x1d\xa9\x1f%\x04\xd4\xb4\xf8"\xc7\x11\xe7\x8e\xb2\x8d|\x8dZ\x04')
172.20.10.1 - - [10/Nov/2022 03:22:43] "GET / HTTP/1.1" 400 -
172.20.10.1 - - [10/Nov/2022 03:22:43] code 400, message Bad HTTP/0.9 request type ('\x16\x03\x01\x00}\x01\x00\x00y\x03\x01\x1da"\x06"\x81\x9b\x95\x86\x00\x17\xcd:\x87\xb1\x08\x907\xad5\xcd}\xf0\xfc\xed')
172.20.10.1 - - [10/Nov/2022 03:22:43] "GET / HTTP/1.1" 400 -
xterm-256color
```

## 4. 本次實驗心得，你學到了什麼東西？

這次的實驗學到了如何透過手機或是網頁發送請求給伺服器，並且也學會伺服器端如何將溫溼度感測器收到的資料透過 Json file 的格式傳給使用者。這算是我第一次接觸到網路類的程式。有多上網去學會一些關於 http 及 https 通訊協定的差別。也了解到平常輸入網址不小心打錯字，造成 404 NOT found 的錯誤的來源是來自於伺服器端將 status code = 404 的網頁畫面顯示給我們所導致的。