BUN LAB
Broadband Ubiquitous Networking Lab

通訊網路實驗

REST API
112學年度 第一學期

Dept. of Electrical and Computer Engineering (ECE)
National Yang Ming Chiao Tung University

REST API

- API（application programming interface）是應用程式的介面/接口

- Representational State Transfer，簡稱 REST (RESTful)，它是一種網路架構風格
  - 使用 HTTP 的協定完整定義 Web Service 在HTTP Request 的各種流程

RESTful 例子

BUN LAB
Broadband Ubiquitous Networking Lab

一般

- 所有使用者資料 /getAllUsers
- 獲取使用者資料 /getUser/1
- 新增使用者資料 /createUser
- 更新使用者資料 /updateUser/1
- 刪除使用者資料 /deleteUser/1

- 不夠直觀簡潔
- 使用拓展子分頁來達到 HTTP Verb（一個子分頁一個Verb)

REST 風格

- 所有使用者資料 /GET/users
- 獲取使用者資料 /GET/user/1
- 新增使用者資料 /POST/user
- 更新使用者資料 /PUT/user/1
- 刪除使用者資料 /DELETE/user/1

- 直觀簡潔的資源 URI
- 善用 HTTP Verb 達到對資源的操作
- 使用 Web 所接受的資料類型: JSON, XML, YAML 等，最常見的是 JSON

HTTP Request Method

- 常用的有 GET / PUT / POST / DELETE

- **GET** : 向指定的資源要求取得資料，並不會動到內部資源
- PUT : 向指定資源位置提交更新內容
- **POST** : 向指定資源提交資料
- DELETE : 向指定資源位置請求刪除內容

HTTP Request Example

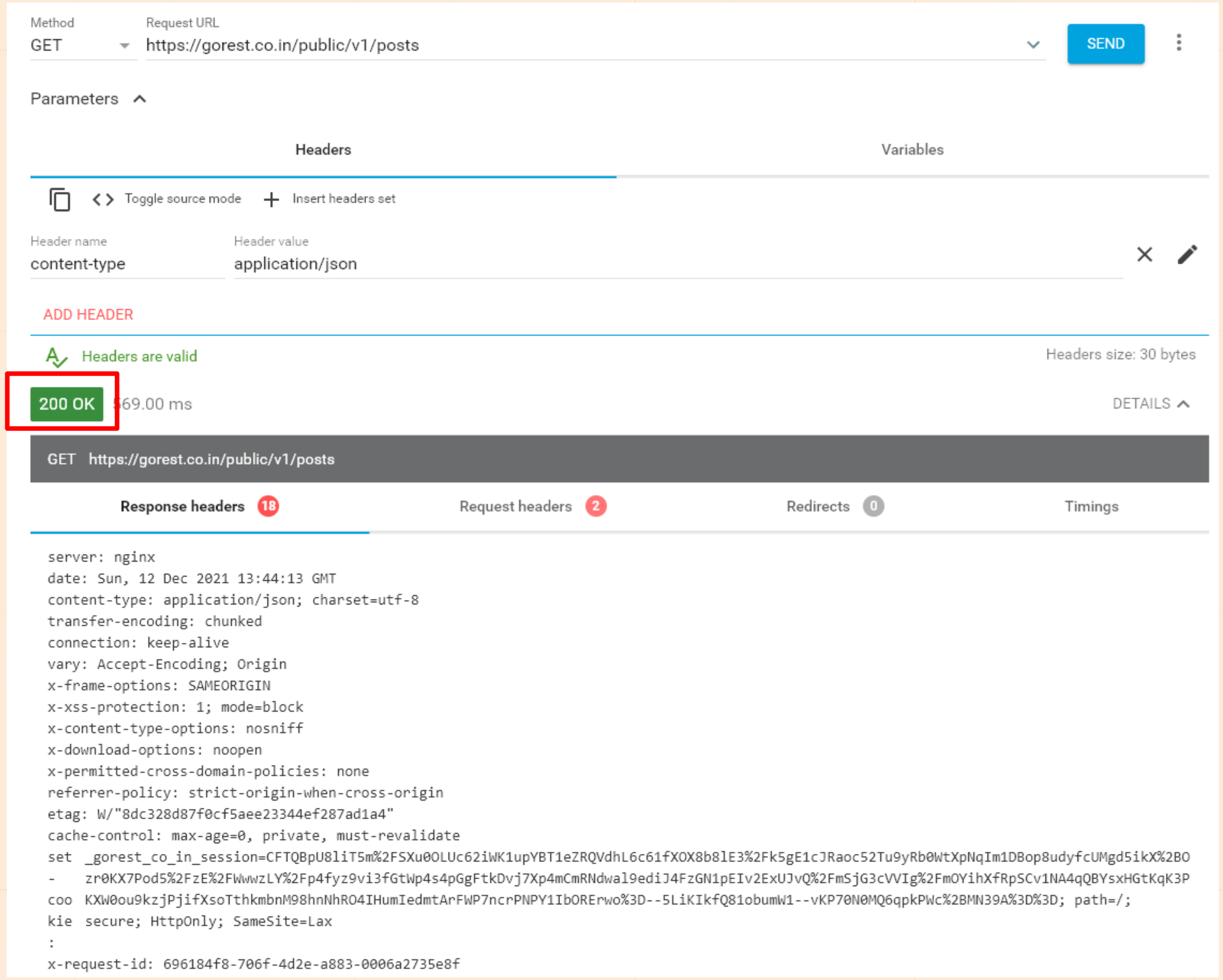
```
POST /task/ HTTP/1.1
Host: localhost:9090
Content-Type: application/json;

{
  "text" : "AAA",
  "tags" : ["BBB", "BBB"],
  "due" : "2020-05-23T18:25:43.511Z"
}
```

Json 格式

```
{
      "String"   :   "This is a string."  ,
      "Value"   : 123456,
      "Boolean"   : True,
      "Object"   :
    {
            "var1"   :   "AAA"  ,
            "var2"   :   "BBB"
    },
      "Array"   :
    [
            "var3"   : 0,
            "var4"   : False
    ]
}
```

HTTP Response Example

BUN LAB
Broadband Ubiquitous Networking Lab



Method
GET

Request URL
https://gorest.co.in/public/v1/posts

SEND

Parameters

Headers | Variables

Toggle source mode | Insert headers set

Header name: content-type
Header value: application/json

ADD HEADER

Headers are valid

Headers size: 30 bytes

200 OK 569.00 ms

DETAILS

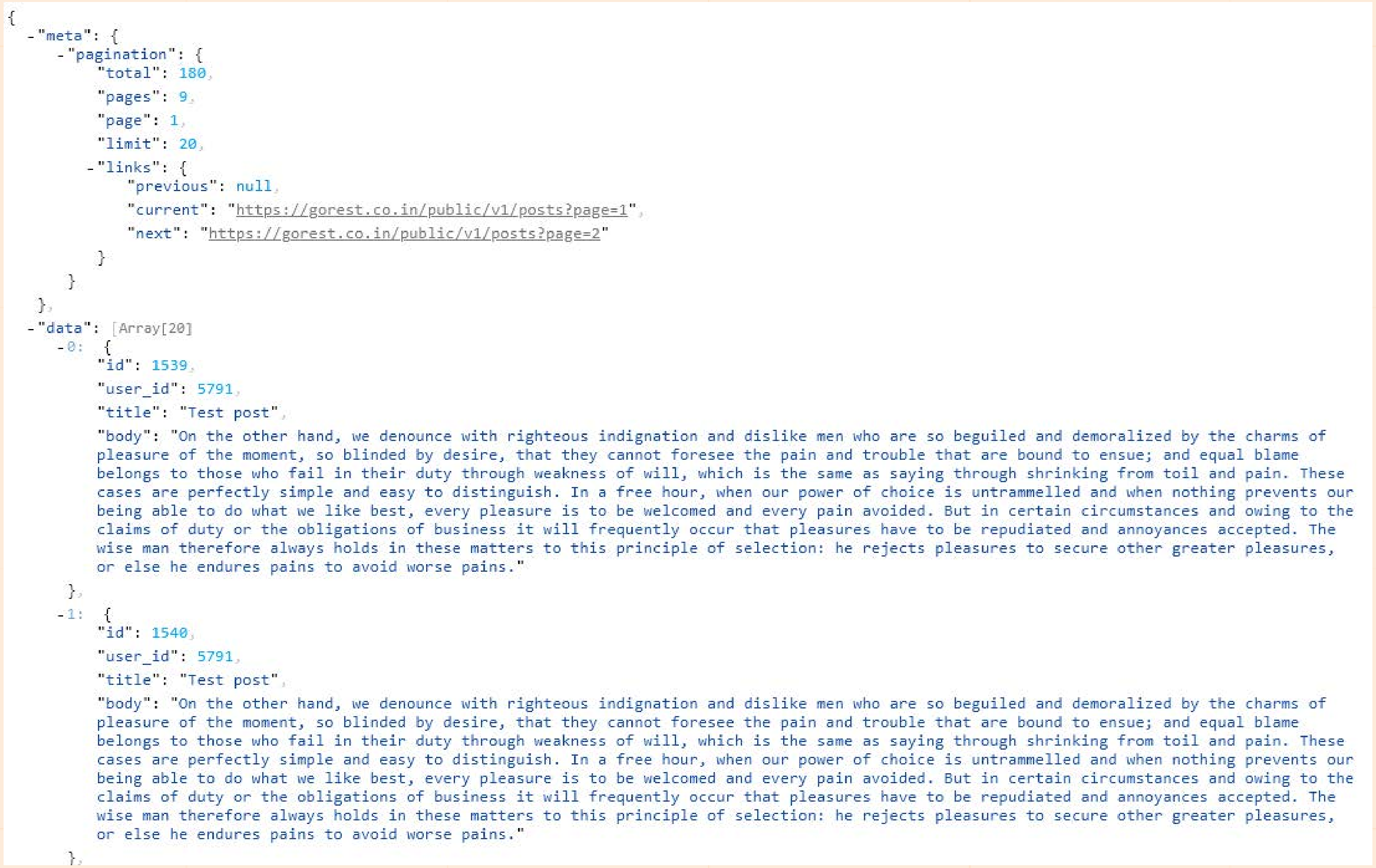
GET https://gorest.co.in/public/v1/posts

Response headers 18 | Request headers 2 | Redirects 0 | Timings

```
server: nginx
date: Sun, 12 Dec 2021 13:44:13 GMT
content-type: application/json; charset=utf-8
transfer-encoding: chunked
connection: keep-alive
vary: Accept-Encoding; Origin
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
x-download-options: noopen
x-permitted-cross-domain-policies: none
referrer-policy: strict-origin-when-cross-origin
etag: W/"8dc328d87f0cf5aee23344ef287ad1a4"
cache-control: max-age=0, private, must-revalidate
set  _gorest_co_in_session=CFTQBpU8liT5m%2FSXu0OLUc62iWK1upYBT1eZRQVdhL6c61fXOX8b8lE3%2Fk5gE1cJRaoc52Tu9yRb0WtXpNqIm1DBop8udyfcUMgd5ikX%2BO
-    zr0KX7Pod5%2FzE%2FWwwzLY%2Fp4fyz9vi3fGtWp4s4pGgFtkDvj7Xp4mCmRNdwal9ediJ4FzGN1pEIv2ExUJvQ%2FmSjG3cVVIg%2FmOYihXfRpSCv1NA4qQBYsxHGtKqK3P
coo  KXW0ou9kzjPjifXsoTthkmbnM98hnNhRO4IHumIedmtArFWP7ncrPNPY1IbORErwo%3D--5LiKIkfQ81obumW1--vKP70N0MQ6qpkPWc%2BMN39A%3D%3D; path=/;
kie  secure; HttpOnly; SameSite=Lax
:
x-request-id: 696184f8-706f-4d2e-a883-0006a2735e8f
```

HTTP Response Example

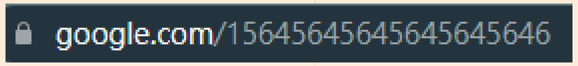
BUN LAB
Broadband Ubiquitous Networking Lab



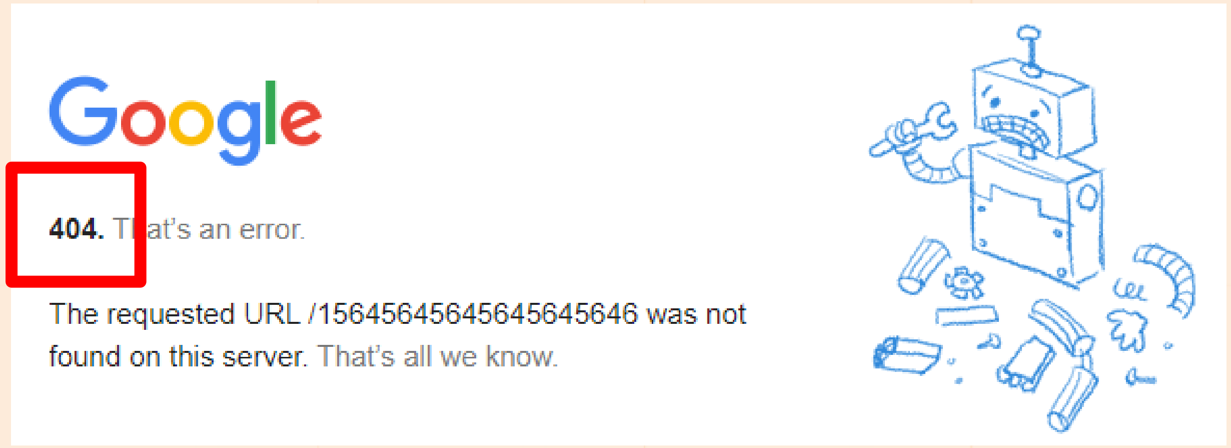
```
{
 - "meta": {
    - "pagination": {
          "total": 180,
          "pages": 9,
          "page": 1,
          "limit": 20,
        - "links": {
              "previous": null,
              "current": "https://gorest.co.in/public/v1/posts?page=1",
              "next": "https://gorest.co.in/public/v1/posts?page=2"
          }
      }
  },
 - "data": [Array[20]
    - 0:  {
          "id": 1539,
          "user_id": 5791,
          "title": "Test post",
          "body": "On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of
          pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame
          belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These
          cases are perfectly simple and easy to distinguish. In a free hour, when our power of choice is untrammelled and when nothing prevents our
          being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the
          claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted. The
          wise man therefore always holds in these matters to this principle of selection: he rejects pleasures to secure other greater pleasures,
          or else he endures pains to avoid worse pains."
      },
    - 1:  {
          "id": 1540,
          "user_id": 5791,
          "title": "Test post",
          "body": "On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of
          pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame
          belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These
          cases are perfectly simple and easy to distinguish. In a free hour, when our power of choice is untrammelled and when nothing prevents our
          being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the
          claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted. The
          wise man therefore always holds in these matters to this principle of selection: he rejects pleasures to secure other greater pleasures,
          or else he endures pains to avoid worse pains."
      },
```

HTTP Status Codes

- 為從伺服器端回應的狀態，夾帶在 HTTP Response 中，由三個數字組成



google.com/15645645645645645646



404. That's an error.

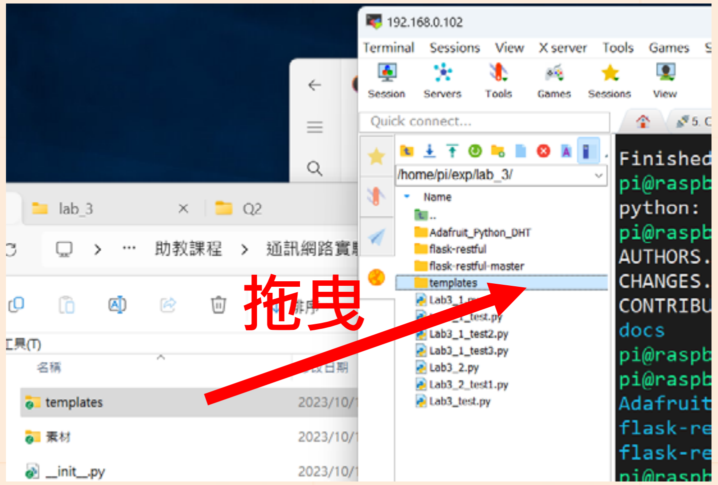
The requested URL /15645645645645645646 was not found on this server. That's all we know.

下載本次實驗函式庫

- sudo pip install flask

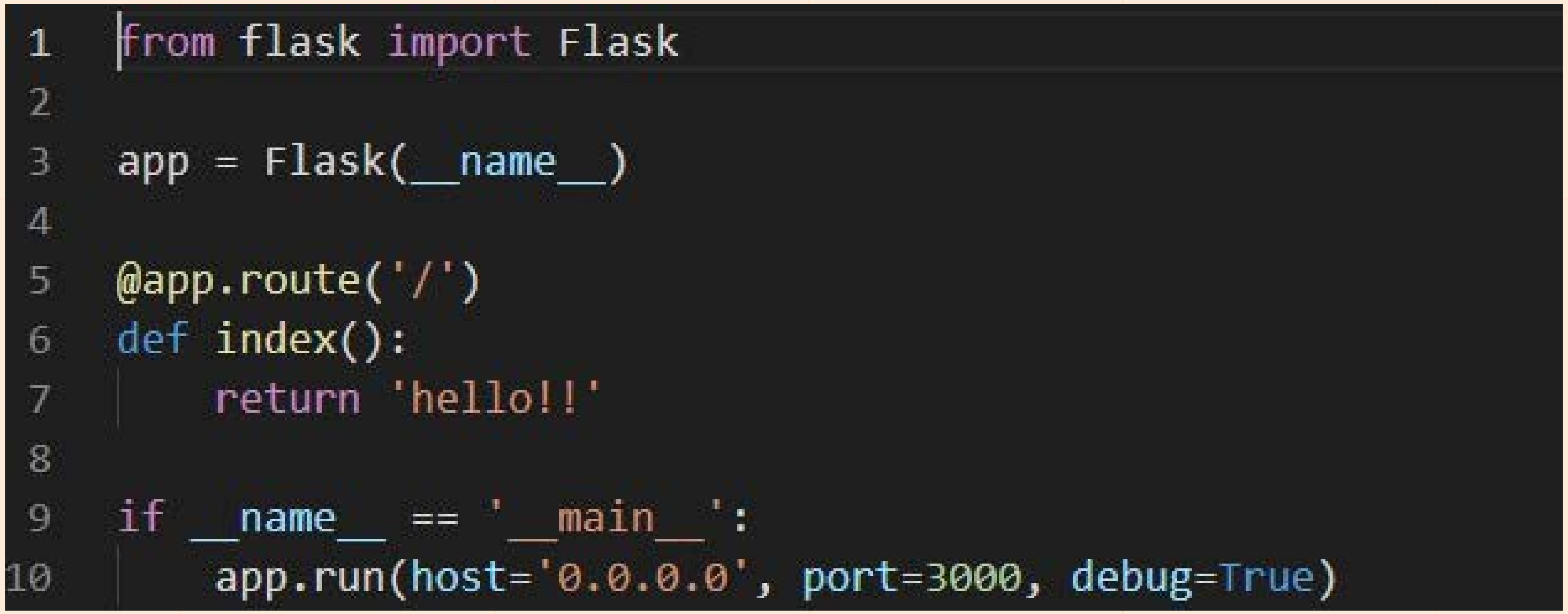
- git clone https://github.com/flask-restful/flask-restful.git
  - ◆ cd flask-restful
  - ◆ sudo python setup.py develop

- 程式碼已放在 E3
  - ◆ templates 資料夾也需要下載，
    並連同整個資料夾上傳至Rpi



```
Finished
pi@raspb
python:
pi@raspb
AUTHORS.
CHANGES.
CONTRIBU
docs
pi@raspb
pi@raspb
Adafruit
flask-re
flask-re
```

Code 解釋



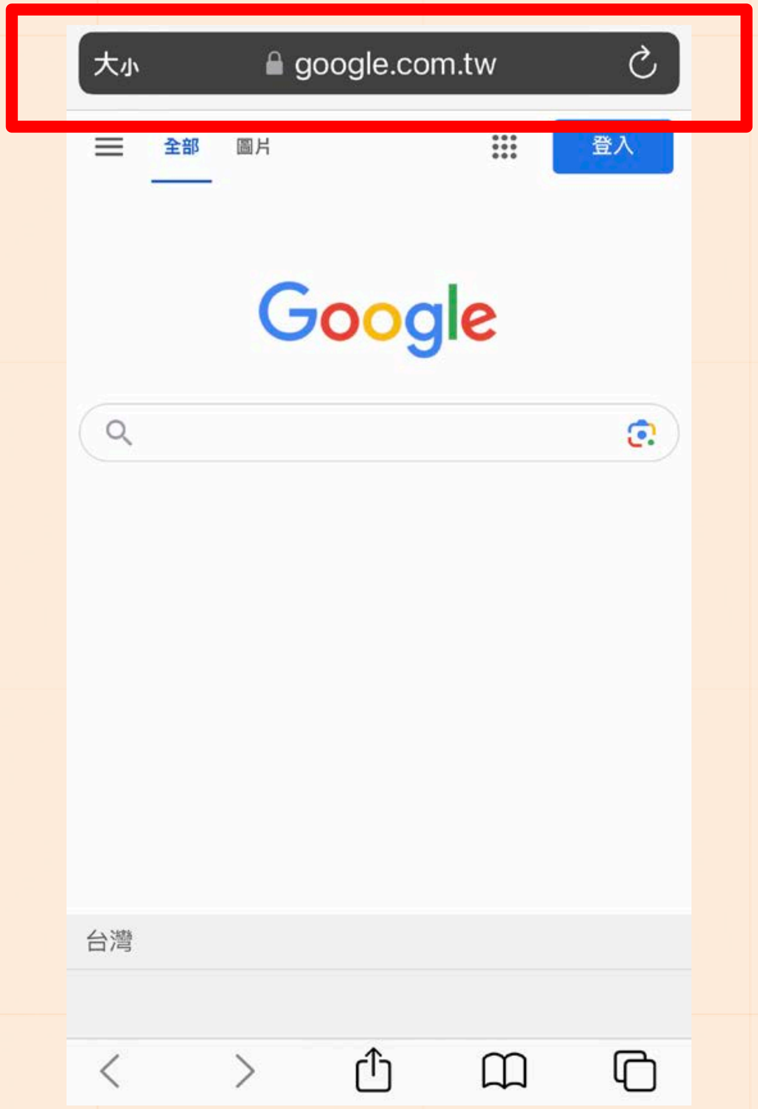
```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return 'hello!!'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=3000, debug=True)
```

- from flask import Flask
    - 引入相關 Flask 函式庫
- app = Flask(__name__)
    - 初始化 Flask 物件，並貼上 app 這個 flag，之後就可以方便使用這個物件裡面的功能
- @app.route('/')
    - 創造出主網域下 "/" 這個網址，並定義 Request 該網址時可用的手段

- def index():
  - 定義一個函數，只要使用者請求相應的網址，就執行該函數
- return
  - 使用者請求該網址之後，根據函數運算之後回傳給使用者的資料，也就是網頁的資料
- app.run()
  - 使整個 Flask Web service 運作
  - 定義 host 地址與 port

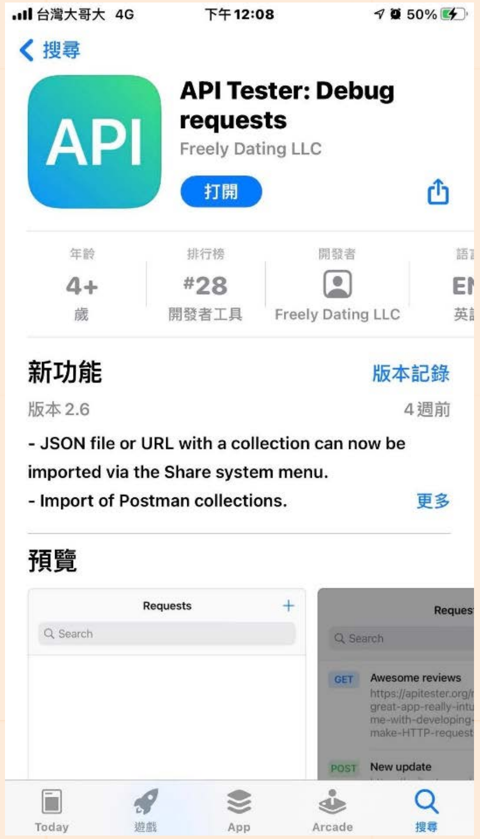
如何發送 HTTP Request

- 1. 使用手機或電腦網頁瀏覽器
  - 輸入 Server IP 地址或是域名即可使用
  - 最容易使用的方式



如何發送 HTTP Request

- 2. 使用手機 APP
  - 能根據需求進行 GET、POST、等 HTTP Request
  - 能自己設定夾帶在 HTTP Request 中的 Json



台灣大哥大 4G 下午12:08 50%

搜尋

API Tester: Debug requests

Freely Dating LLC

打開

年齡 4+ 歲 | 排行榜 #28 開發者工具 | 開發者 Freely Dating LLC | 語言 EN 英

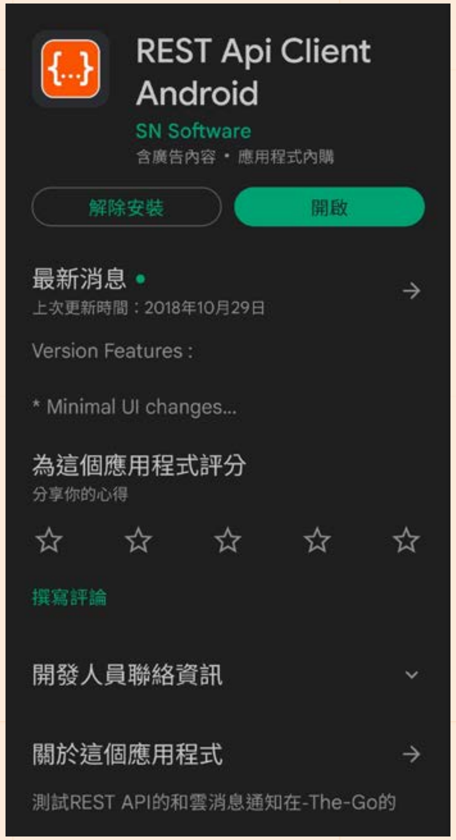
新功能 版本記錄

版本 2.6 4 週前

- JSON file or URL with a collection can now be imported via the Share system menu.
- Import of Postman collections. 更多

預覽

Requests
Search
GET Awesome reviews
https://apitester.org/great-app-really-intu me-with-developing-make-HTTP-request
POST New update
Today
遊戲
App
Arcade
搜尋



REST Api Client Android

SN Software

含廣告內容・應用程式內購

解除安裝 開啟

最新消息

上次更新時間：2018年10月29日

Version Features :

* Minimal UI changes...

為這個應用程式評分

分享你的心得

撰寫評論

開發人員聯絡資訊

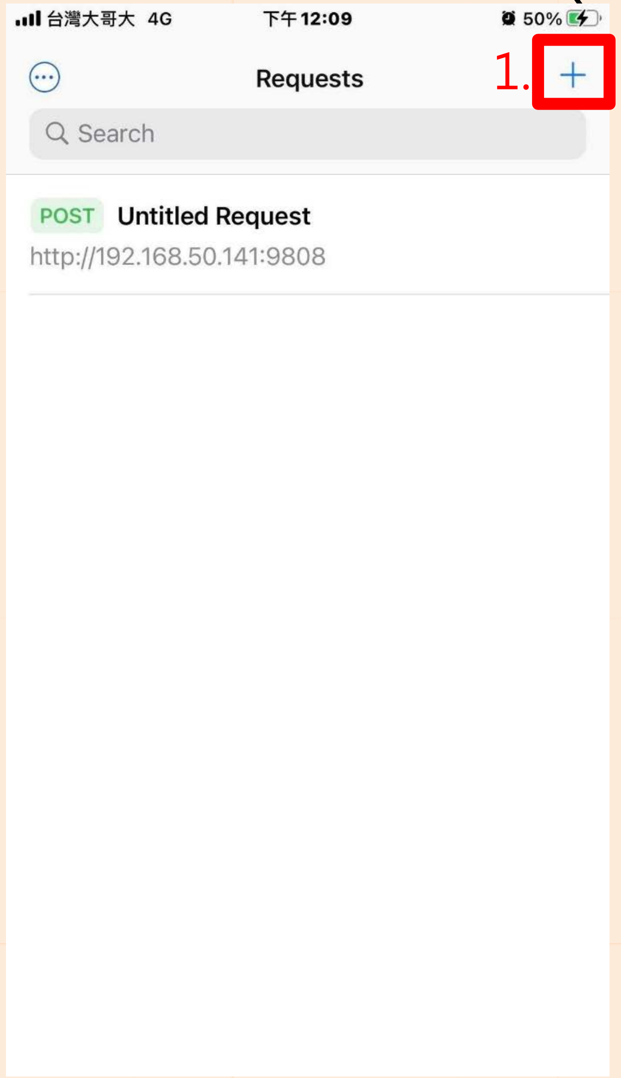
關於這個應用程式

測試REST API的和雲消息通知在-The-Go的

※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 (AP)

如何發送 HTTP Request

- 2. 使用手機 APP (IOS、 Android)

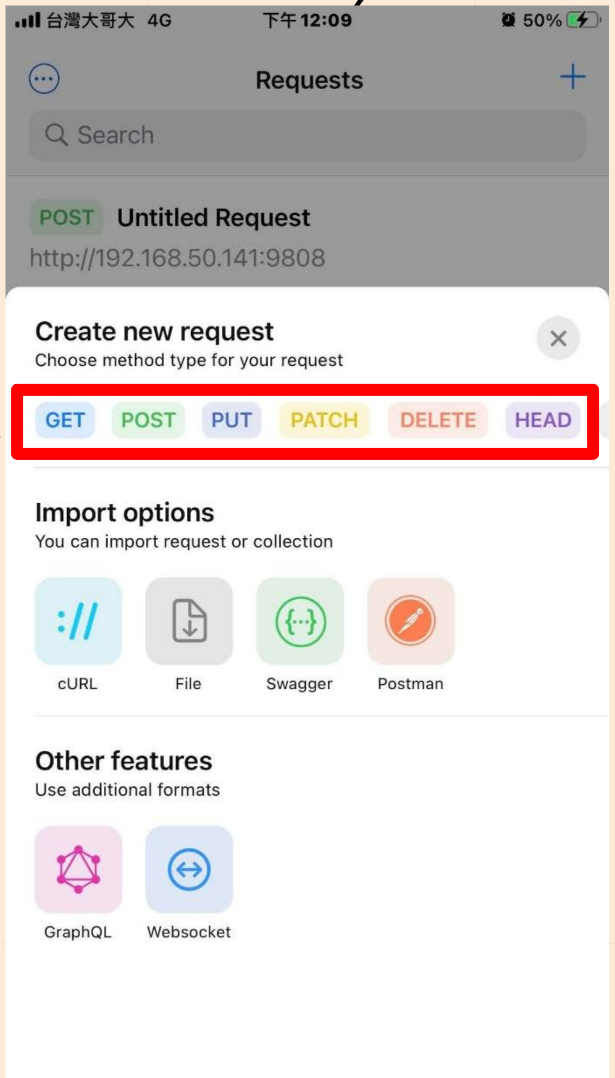

1.

台灣大哥大 4G 下午12:09 50%

Requests

Search

POST Untitled Request
http://192.168.50.141:9808

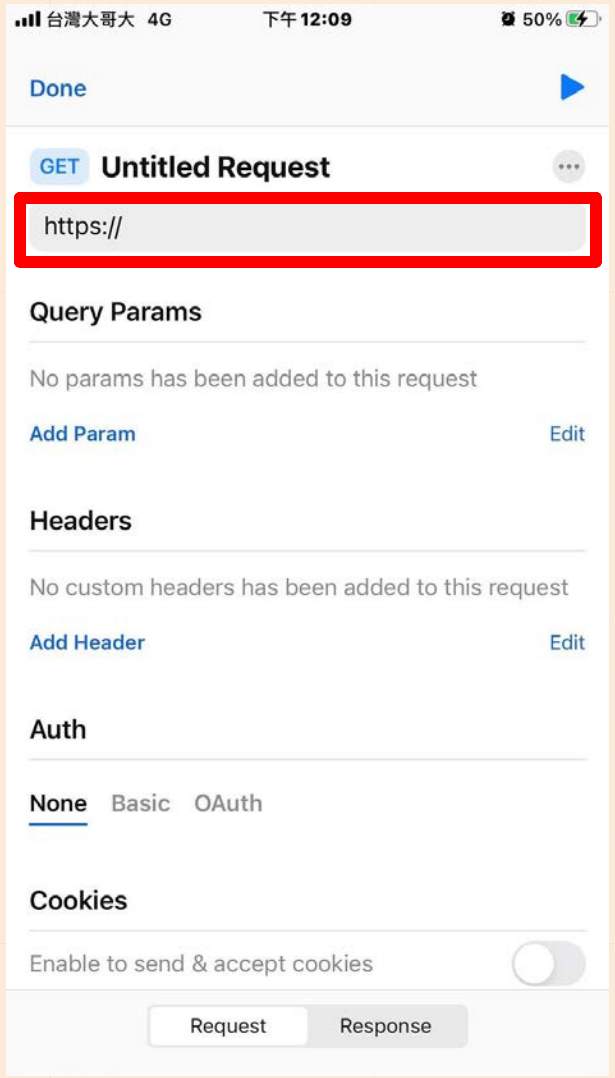

2.

Create new request
Choose method type for your request

GET POST PUT PATCH DELETE HEAD

Import options
You can import request or collection

Other features
Use additional formats


3.

Done

GET Untitled Request

https://

Query Params

No params has been added to this request

Add Param Edit

Headers

No custom headers has been added to this request

Add Header Edit

Auth

None Basic OAuth

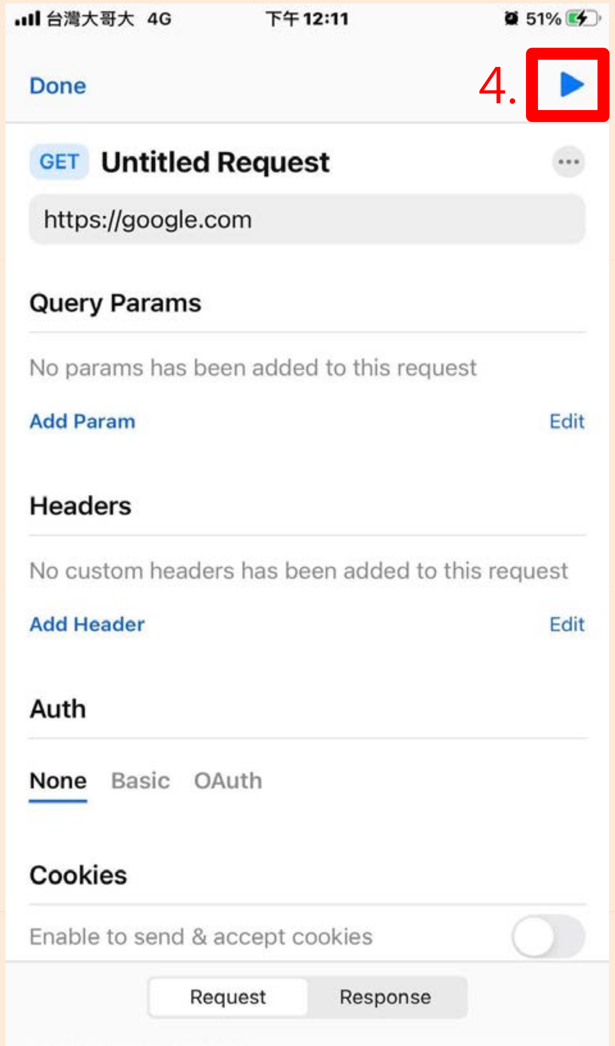
Cookies

Enable to send & accept cookies

Request Response

如何發送 HTTP Request

- 2. 使用手機 APP (IOS 、 Android)



https://google.com

Query Params

No params has been added to this request

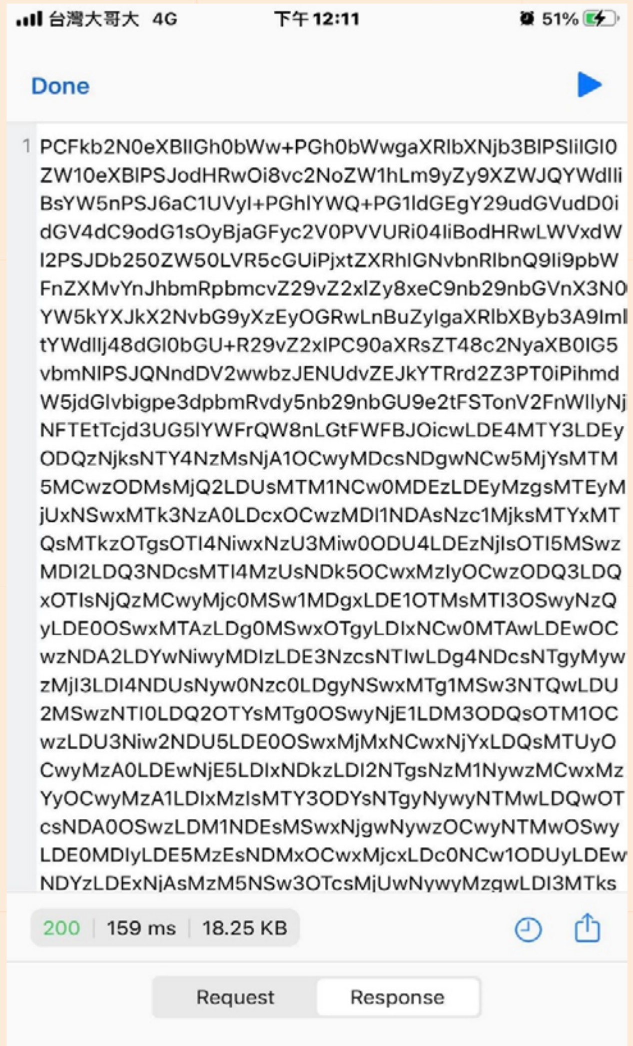
Headers

No custom headers has been added to this request

Auth

Cookies

Enable to send & accept cookies

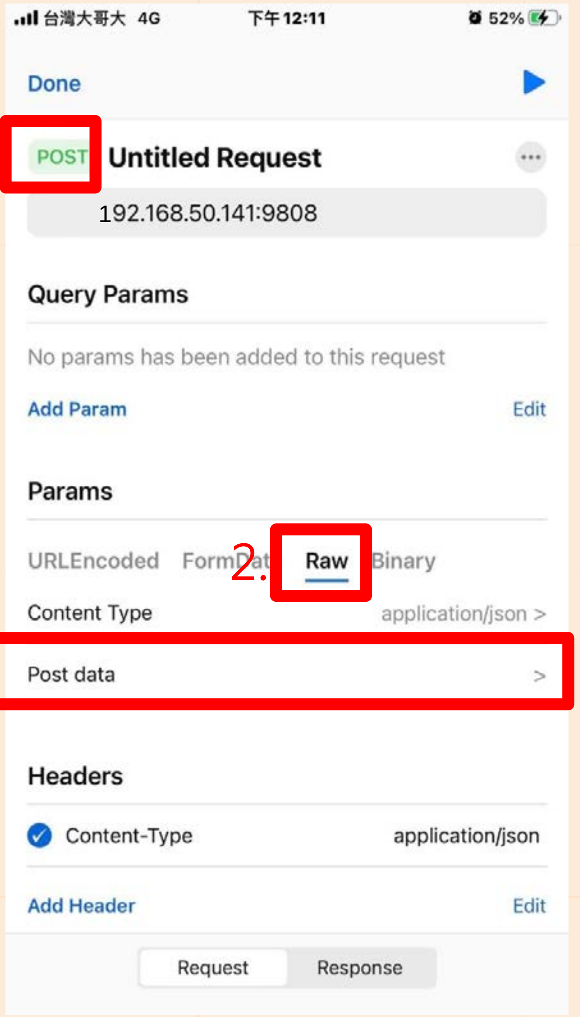


PCFkb2N0eXBlIGh0bWw+PGh0bWwgaXRlbXNjb3BlPSIiIGl0
ZW10eXBlPSJodHRwOi8vc2NoZW1hLm9yZy9XZWJQYWdlIi
BsYW5nPSJ6aC1UVyI+PGhlYWQ+PG1ldGEgY29udGVudD0i
dGV4dC9odG1sOyBjaGFyc2V0PVVURi04IiBodHRwLWVxdW
l2PSJDb250ZW50LVR5cGUiPjxtZXRhIGNvbnRlbnQ9Ii9pbW
FnZXMvYnJhbmRpbmcvZ29vZ2xlZy8xeC9nb29nbGVnX3N0
YW5kYXJkX2NvbG9yXzEyOGRwLnBuZyIgaXRlbXByb3A9Iml
tYWdlIj48dGl0bGU+R29vZ2xlPC90aXRsZT48c2NyaXB0IG5
vbmNlPSJQNndDV2wwbzJENUdvZEJkYTRrd2Z3PT0iPihmd
W5jdGlvbigpe3dpbmRvdy5nb29nbGU9e2tFSTonV2FnWllyNj
NFTEtTcjd3UG5lYWFrQW8nLGtFWFBJOicwLDE4MTY3LDEy
ODQzNjksNTY4NzMsNjA1OCwyMDcsNDgwNCw5MjYsMTM
5MCwzODMsMjQ2LDUsMTM1NCw0MDEzLDEyMzgsMTEyM
jUxNSwxMTk3NzA0LDcxOCwzMDI1NDAsNzc1MjksMTYxMT
QsMTkzOTgsOTI4NiwxNzU3Miw0ODU4LDEzNjIsOTI5MSwz
MDI2LDQ3NDcsMTI4MzUsNDk5OCwxMzIyOCwzODQ3LDQ
xOTIsNjQzMCwyMjc0MSw1MDgxLDE1OTMsMTI3OSwyNzQ
yLDE0OSwxMTAzLDg0MSwxOTgyLDIxNCw0MTAwLDEwOC
wzNDA2LDYwNiwyMDIzLDE3NzcsNTIwLDg4NDcsNTgyMyw
zMjI3LDI4NDUsNyw0Nzc0LDgyNSwxMTg1MSw3NTQwLDU
2MSwzNTI0LDQ2OTYsMTg0OSwyNjE1LDM3ODQsOTM1OC
wzLDU3Niw2NDU5LDE0OSwxMjMxNCwxNjYxLDQsMTUyO
CwyMzA0LDEwNjE5LDIxNDkzLDI2NTgsNzM1NywzMCwxMz
YyOCwyMzA1LDIxMzIsMTY3ODYsNTgyNywyNTMwLDQwOT
csNDA0OSwzLDM1NDEsMSwxNjgwNywzOCwyNTMwOSwy
LDE0MDIyLDE5MzEsNDMxOCwxMjcxLDc0NCw1ODUyLDEw
NDYzLDExNjAsMzM5NSw3OTcsMjUwNywyMzgwLDI3MTks

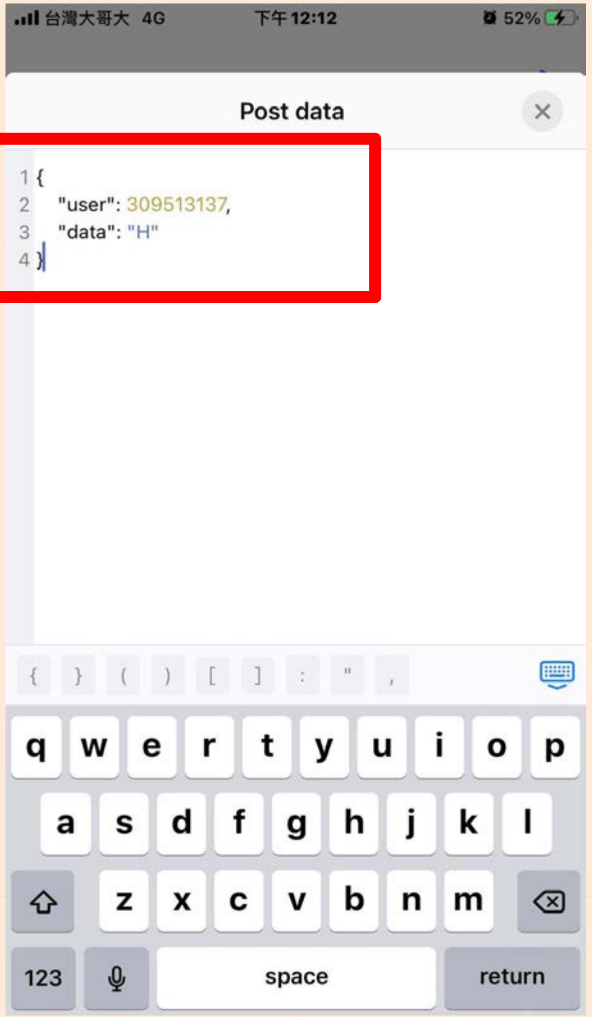
200 | 159 ms | 18.25 KB

如何發送 HTTP Request

- 2. 使用手機 APP (IOS 、 Android)

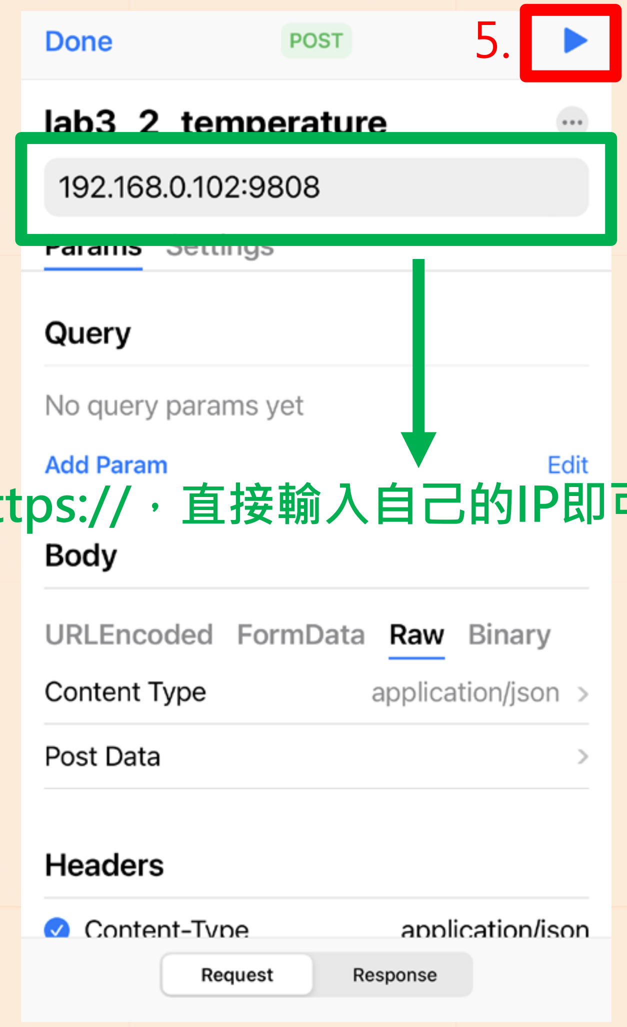


```
POST  Untitled Request
192.168.50.141:9808

Query Params
No params has been added to this request
Add Param                                   Edit

Params
URLEncoded   FormData   Raw   Binary
Content Type                  application/json >
Post data                                      >

Headers
✓ Content-Type                 application/json
Add Header                                  Edit

Request   Response
```



Post data

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



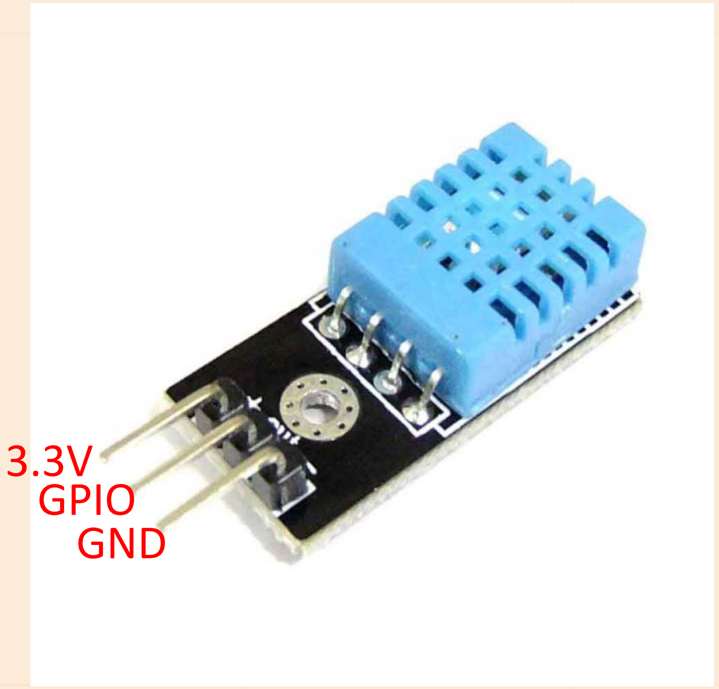
```
Done   POST                                ▶
lab3_2_temperature
192.168.0.102:9808
Params   Settings

Query
No query params yet
Add Param                                   Edit

Body
URLEncoded   FormData   Raw   Binary
Content Type                  application/json >
Post Data                                      >

Headers
✓ Content-Type                 application/json

Request   Response
```

請刪除 https://，直接輸入自己的IP即可

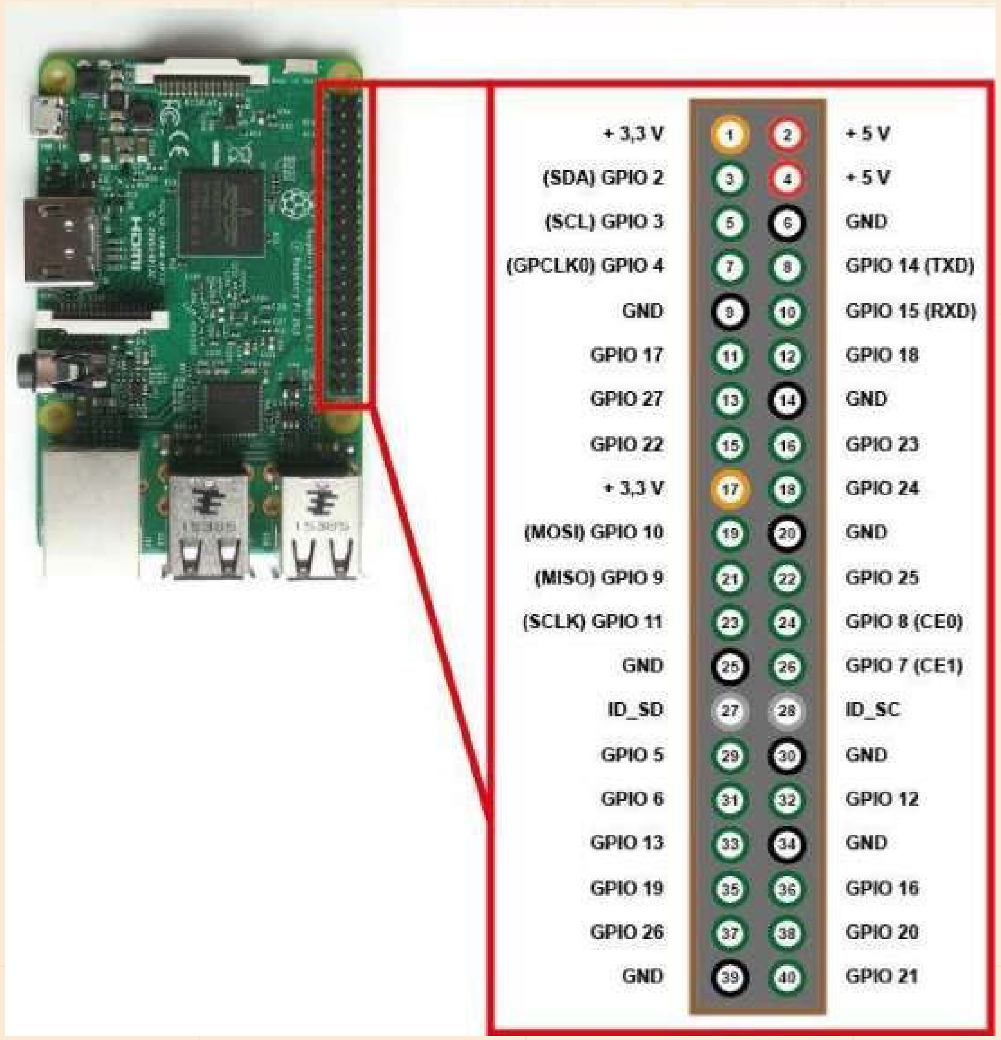
溫溼度感測器

BUN LAB
Broadband Ubiquitous Networking Lab

- DHT11 溫溼度感測器
  - 溫度: 0 ~ 50 ℃，誤差 ±2 ℃
  - 濕度: 20 ~ 90 %，誤差 ±5 %
  - 使用三個腳位: Data , VCC , GND (out、+ 、-)
  - Data 腳位統一連接到 RPi 板上的 GPIO4 (Pin 7)
  - VCC 連接到 RPi 板上的 3.3 V 位置
  - GND → 接地



腳位參考圖



+ 3,3 V	1	2	+ 5 V
(SDA) GPIO 2	3	4	+ 5 V
(SCL) GPIO 3	5	6	GND
(GPCLK0) GPIO 4	7	8	GPIO 14 (TXD)
GND	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
+ 3,3 V	17	18	GPIO 24
(MOSI) GPIO 10	19	20	GND
(MISO) GPIO 9	21	22	GPIO 25
(SCLK) GPIO 11	23	24	GPIO 8 (CE0)
GND	25	26	GPIO 7 (CE1)
ID_SD	27	28	ID_SC
GPIO 5	29	30	GND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GND
GPIO 19	35	36	GPIO 16
GPIO 26	37	38	GPIO 20
GND	39	40	GPIO 21

下載本次實驗函式庫

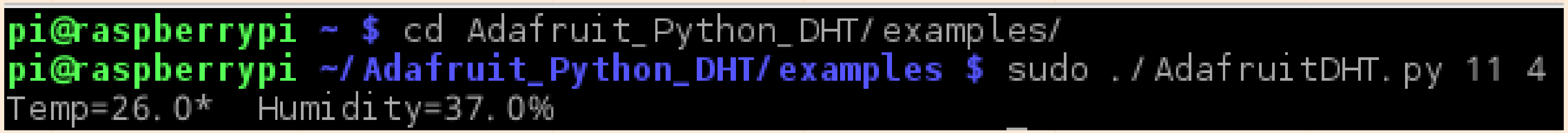
BUN LAB
Broadband Ubiquitous Networking Lab

- DHT11
  - git clone https://github.com/adafruit/Adafruit_Python_DHT.git
  - cd Adafruit_Python_DHT
  - sudo python setup.py install

- GPIO
  - sudo pip install rpi.gpio

溫溼度感測器功能測試

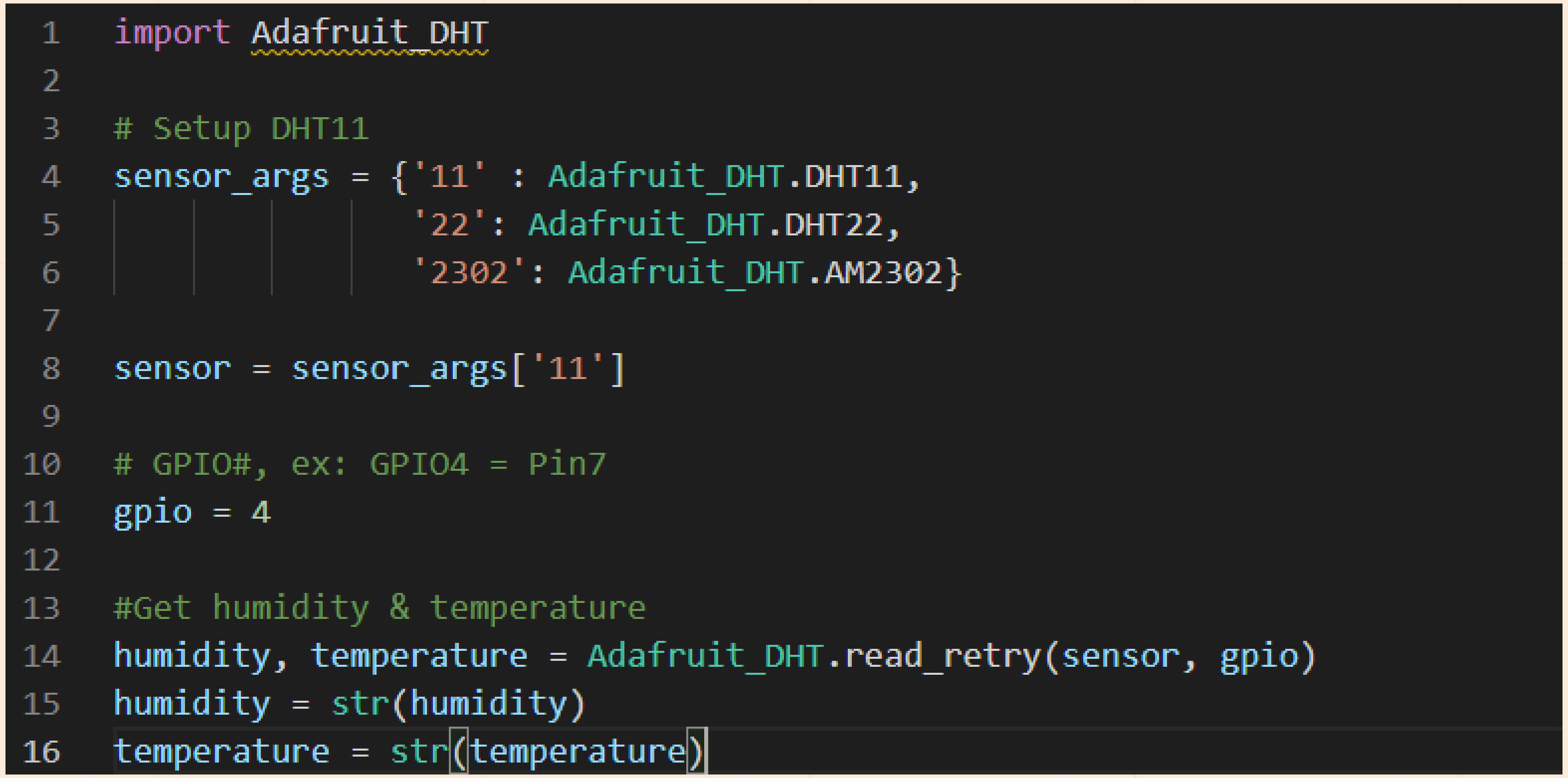
- 執行函式庫提供的測試檔
  - cd Adafruit_Python_DHT/examples
  - sudo ./AdafruitDHT.py 11 4
    - 11 為 DHT11 (也有 22 的型號)
    - 4 為 GPIO 4 (也就是Pin 7)



```
pi@raspberrypi ~ $ cd Adafruit_Python_DHT/examples/
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 11 4
Temp=26.0*  Humidity=37.0%
```

※若測試結果有任何錯誤或是無結果請先自行檢查溫濕度計模組是否有接線錯誤，待確認後再跟助教反應需更換材料或其他處理

Code 解釋



```python
import Adafruit_DHT

# Setup DHT11
sensor_args = {'11' : Adafruit_DHT.DHT11,
               '22': Adafruit_DHT.DHT22,
               '2302': Adafruit_DHT.AM2302}

sensor = sensor_args['11']

# GPIO#, ex: GPIO4 = Pin7
gpio = 4

#Get humidity & temperature
humidity, temperature = Adafruit_DHT.read_retry(sensor, gpio)
humidity = str(humidity)
temperature = str(temperature)
```

溫溼度感測器 替代方案

- 若溫溼度感測器故障，即執行測試檔未出現溫溼度資訊，可嘗試以下方法
- 導入 random 函式庫

```
import random
```

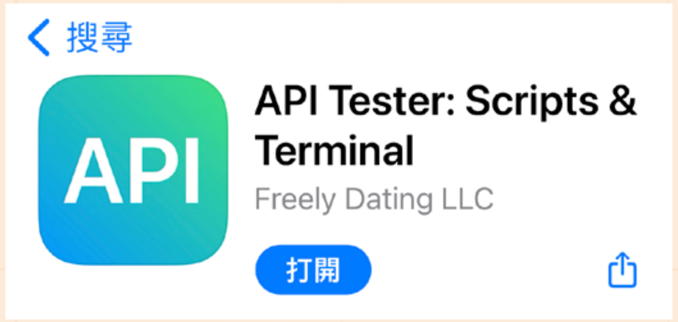
- 以隨機數代替溫、濕度

```
humidity = random.randint(20, 30)
temperature = random.randint(30, 40)
```

本次實驗 Demo

- 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
  - 請在自己手機或PC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

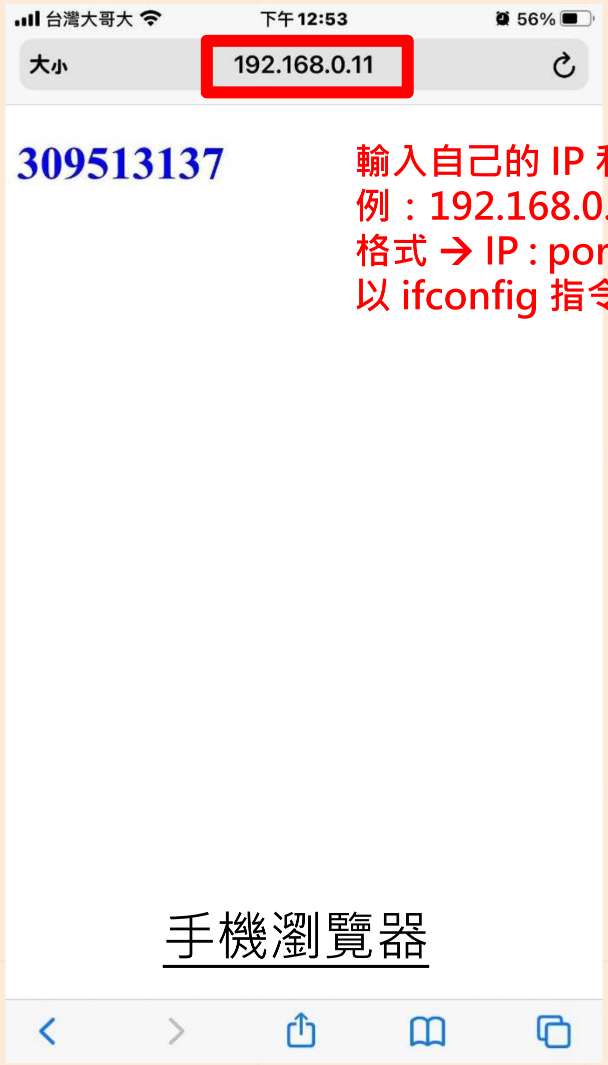
  - IOS : 於 App Store 直接下載 API Tester，或搜尋 HTTP request 並下載相關 APP
  - Android : 於 Google Play 商店直接下載 API Tester，或搜尋 HTTP request 並下載相關 APP



※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 (AP)

本次實驗 Demo

- Q1: 使用GET Method 將自己的學號顯示出來



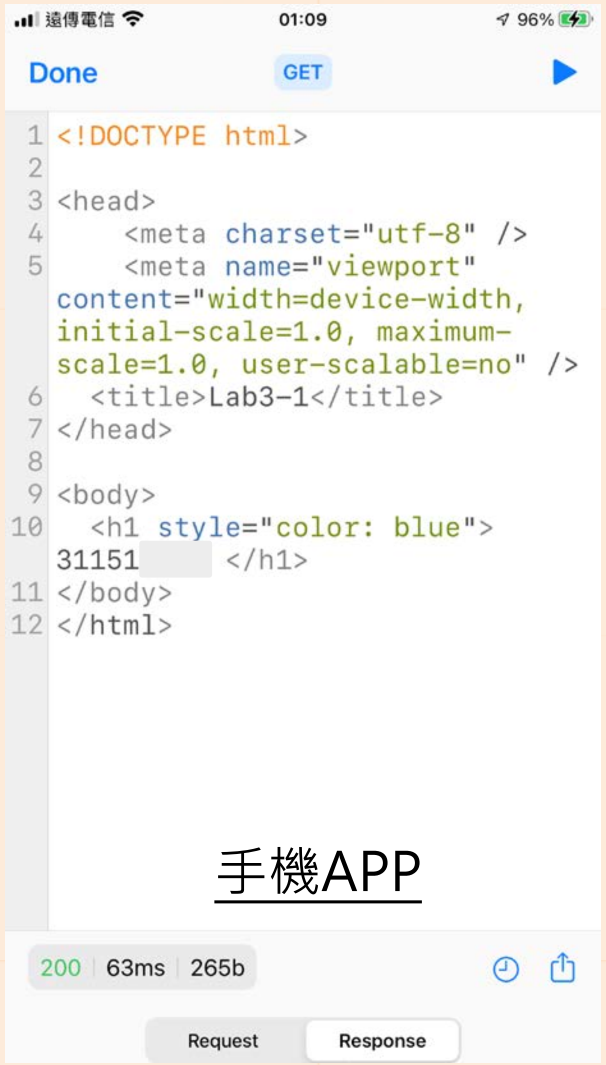
台灣大哥大 下午 12:53 56%

大小 192.168.0.11

309513137

輸入自己的 IP 和 port number
例：192.168.0.11:9808
格式 → IP : port
以 ifconfig 指令查詢 Rpi IP位址

手機瀏覽器



遠傳電信 01:09 96%

Done GET

```
<!DOCTYPE html>

<head>
    <meta charset="utf-8" />
    <meta name="viewport"
content="width=device-width,
initial-scale=1.0, maximum-
scale=1.0, user-scalable=no" />
  <title>Lab3-1</title>
</head>

<body>
  <h1 style="color: blue">
31151[redacted] </h1>
</body>
</html>
```

200 63ms 265b

Request Response

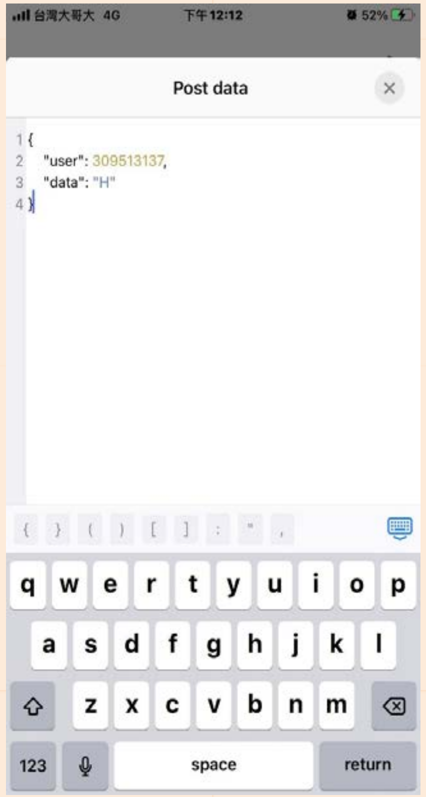
手機APP

※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 (AP)

本次實驗 Demo

- 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**或**濕度**
  - Json 格式　(Post data)

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



```
{
  "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 (AP)

本次實驗 Demo

- Q2: 完成一個能查閱 RPi 溫溼度模組數值 REST API
  - GET Method 部分已經完成，請參考 POST 部分完成剩下的 POST Method
    - GET Method 的內容有關於 POST Method 的實作提示
    - POST Method 溫溼度選擇以 T 、 H 代表即可，實作內容需包含下列4種情形：
      1. 如果傳入 **T**，應回傳 **溫度**、**學號**及指定格式的網頁內容
      2. 如果傳入 **H**，應回傳 **濕度**、**學號**及指定格式的網頁內容
      3. 使用者進行 POST Request 時並沒有提供 json 內容時，應回傳 Status Code = 400 的網頁內容
      4. 如果傳入非 H 、 T 變數時，應回傳 Status Code = 400 的網頁內容

本次實驗 Demo

- Q2 會使用到 jsonify 、 make_response 兩個 Flask 函式，使用方式請參考 Flask 函式，請點擊下列 API 網址參考使用方法： 1. jsonify 2. make_response

- jsonify (**資料**)

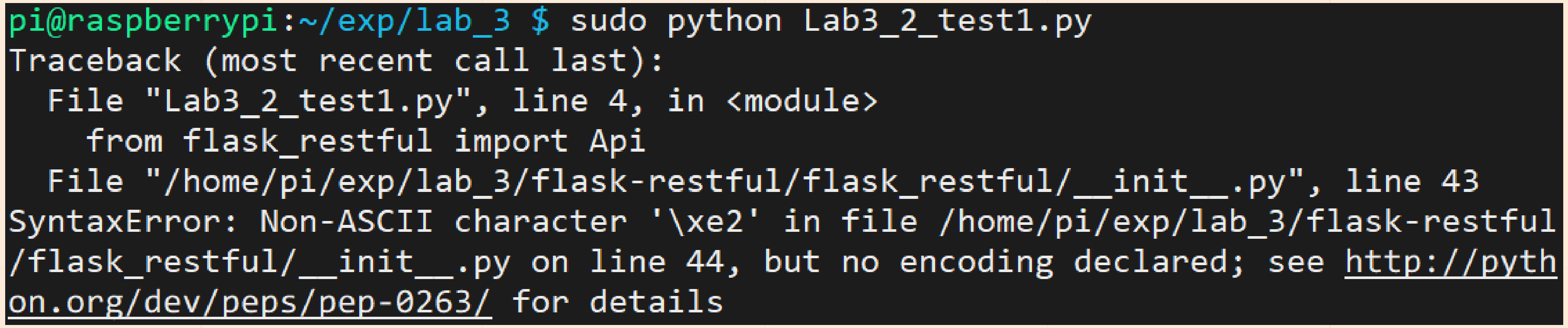
◆ 將資料轉換為 json 格式

- return make_response (**欲回傳的內容**, status code)

◆ 回傳指定內容及狀態碼

本次實驗 Demo

- 若出現下列錯誤訊息，請執行以下步驟以排除編碼問題

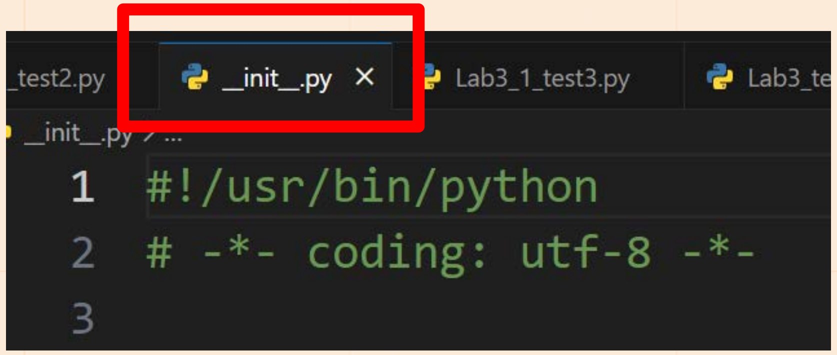


```
pi@raspberrypi:~/exp/lab_3 $ sudo python Lab3_2_test1.py
Traceback (most recent call last):
  File "Lab3_2_test1.py", line 4, in <module>
    from flask_restful import Api
  File "/home/pi/exp/lab_3/flask-restful/flask_restful/__init__.py", line 43
SyntaxError: Non-ASCII character '\xe2' in file /home/pi/exp/lab_3/flask-restful
/flask_restful/__init__.py on line 44, but no encoding declared; see http://pyth
on.org/dev/peps/pep-0263/ for details
```

- 於執行 *git clone https://github.com/flask-restful/flask-restful.git* 的路徑

1. 輸入 cd flask-restful/flask_restful
2. 輸入 nano __init__.py
3. 務必於 __init__.py 最上方加上以下兩行註解：

   #! /usr/bin/python
   # -*- coding: utf-8 -*-

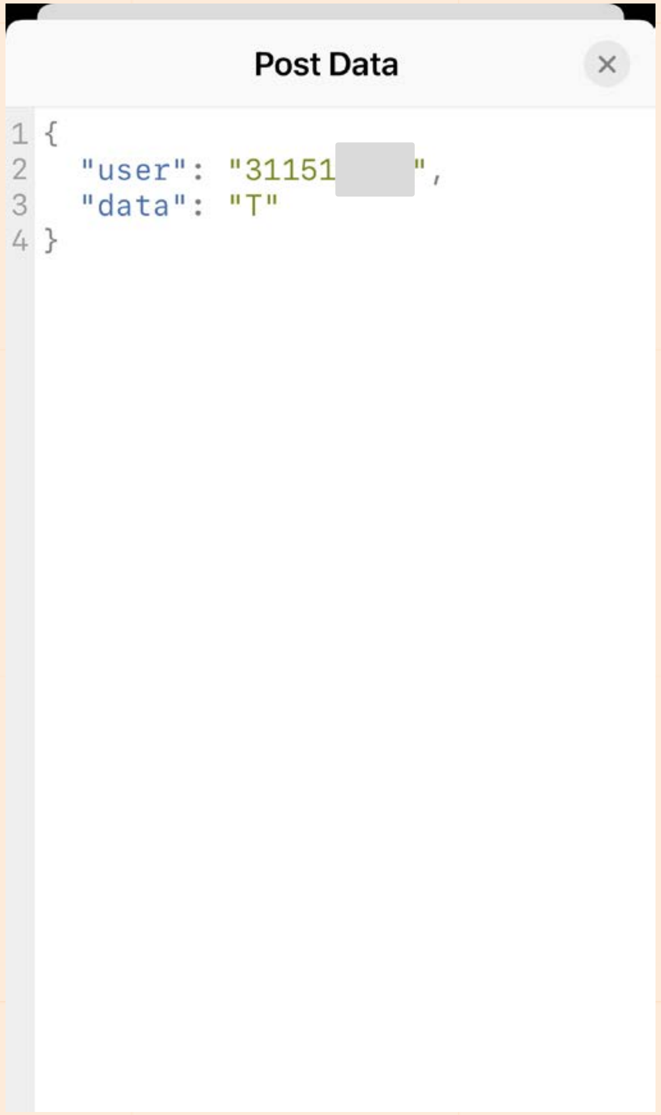


```
#!/usr/bin/python
# -*- coding: utf-8 -*-
```

- 詳情可參考[此網站]

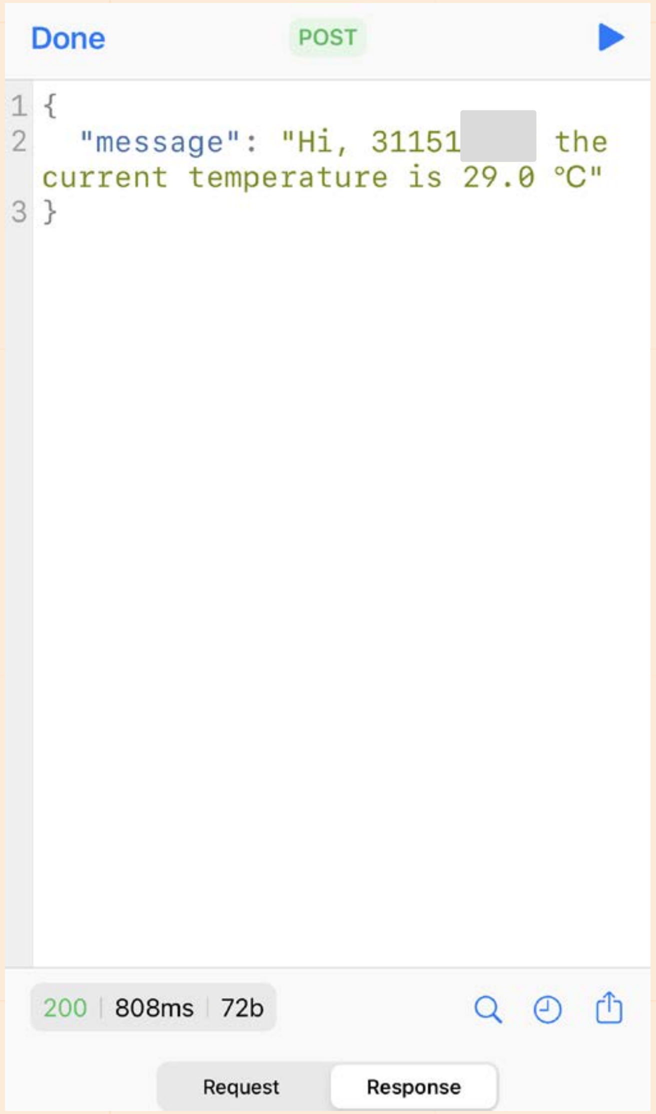
本次實驗 Demo

1. 溫度



Post Data

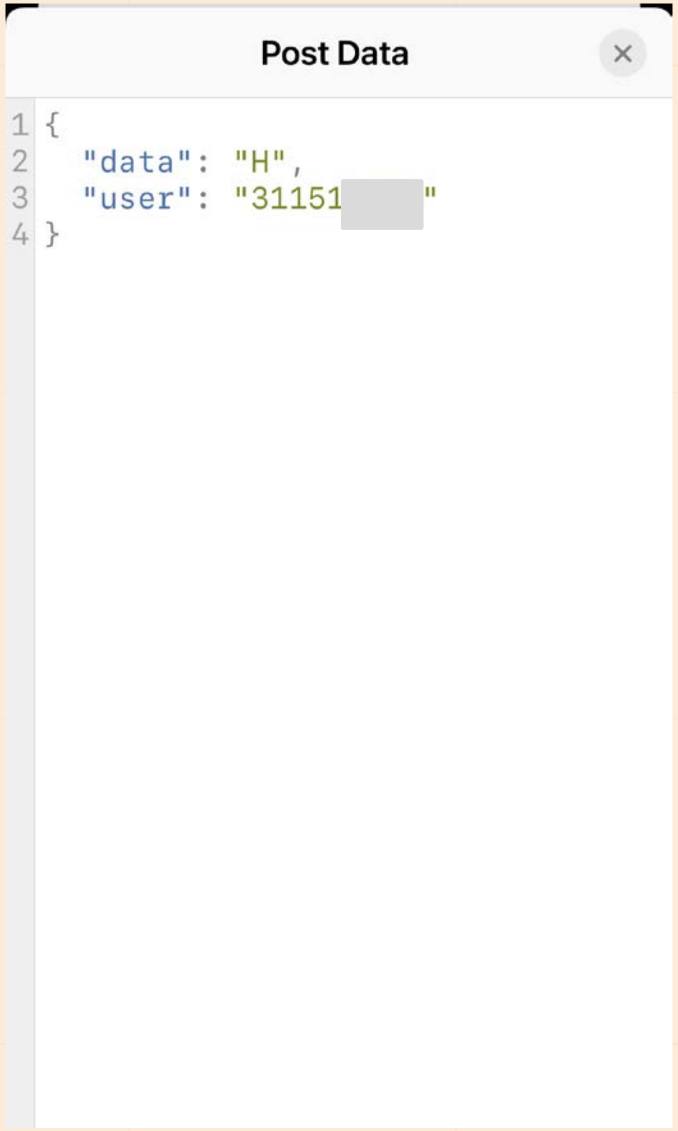
```
{
  "user": "31151[redacted]",
  "data": "T"
}
```



```
{
  "message": "Hi, 31151[redacted] the current temperature is 29.0 °C"
}
```

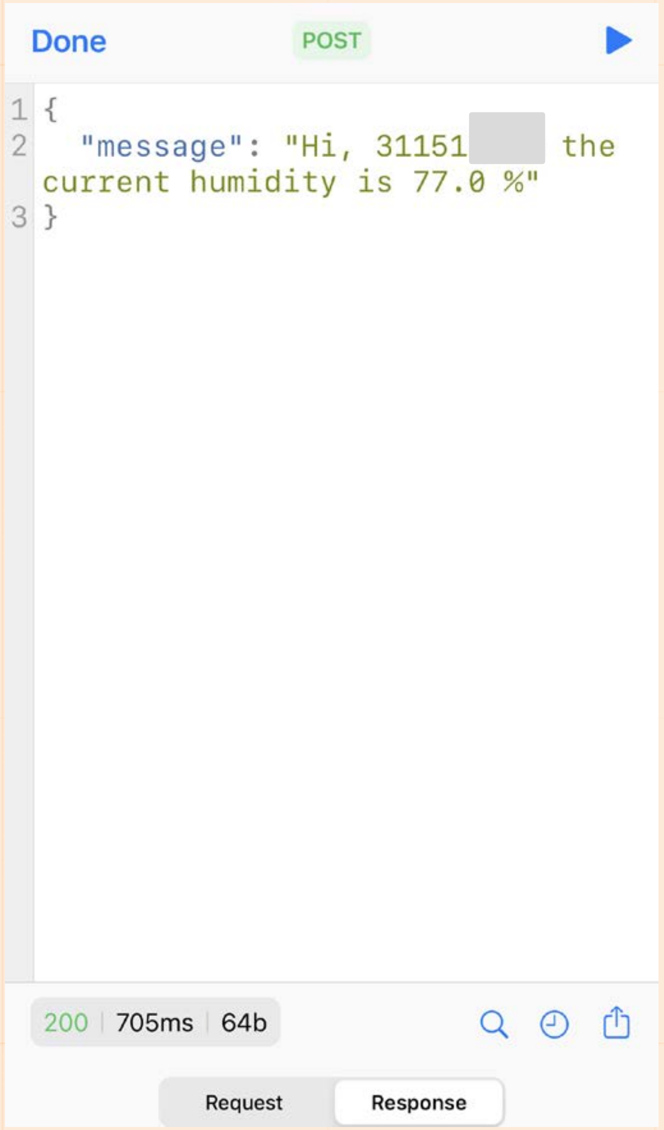
本次實驗 Demo

2. 濕度



Post Data

```
{
  "data": "H",
  "user": "31151[redacted]"
}
```



Done POST

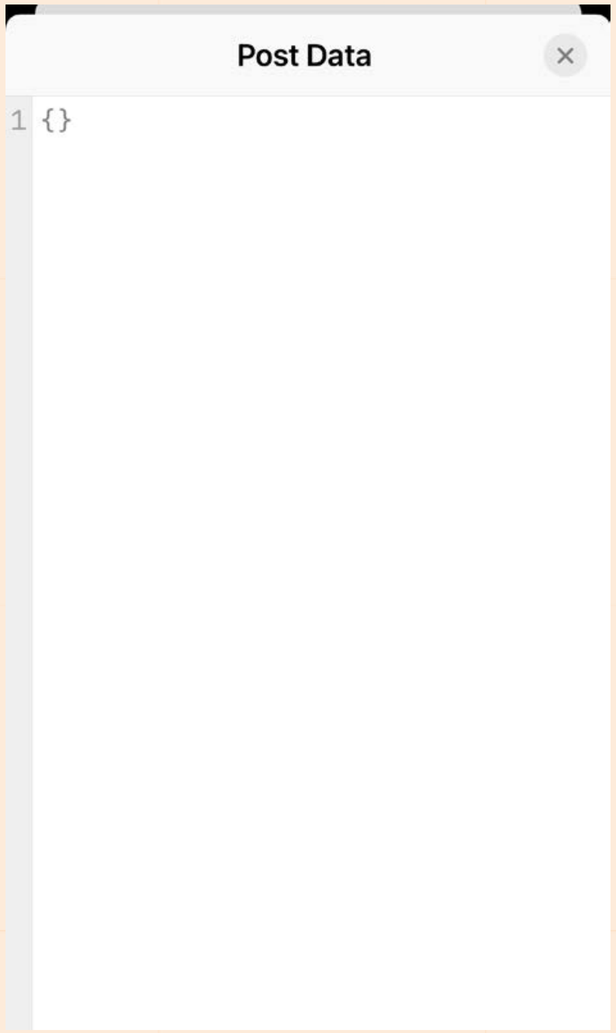
```
{
  "message": "Hi, 31151[redacted] the current humidity is 77.0 %"
}
```

200 | 705ms | 64b

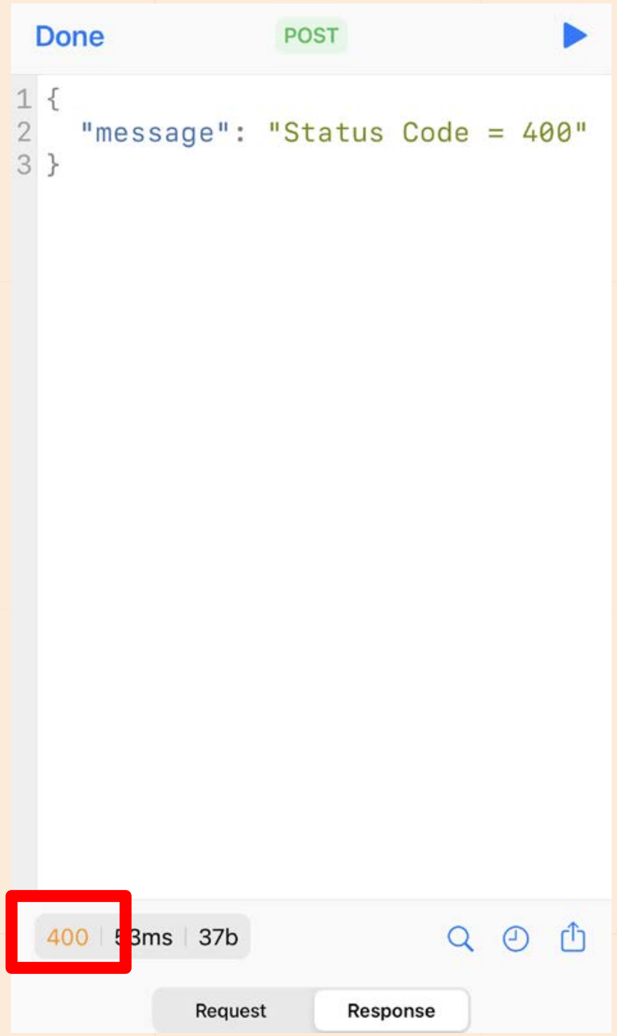
Request Response

本次實驗 Demo

3. POST 時未提供 json 內容



```
{}
```

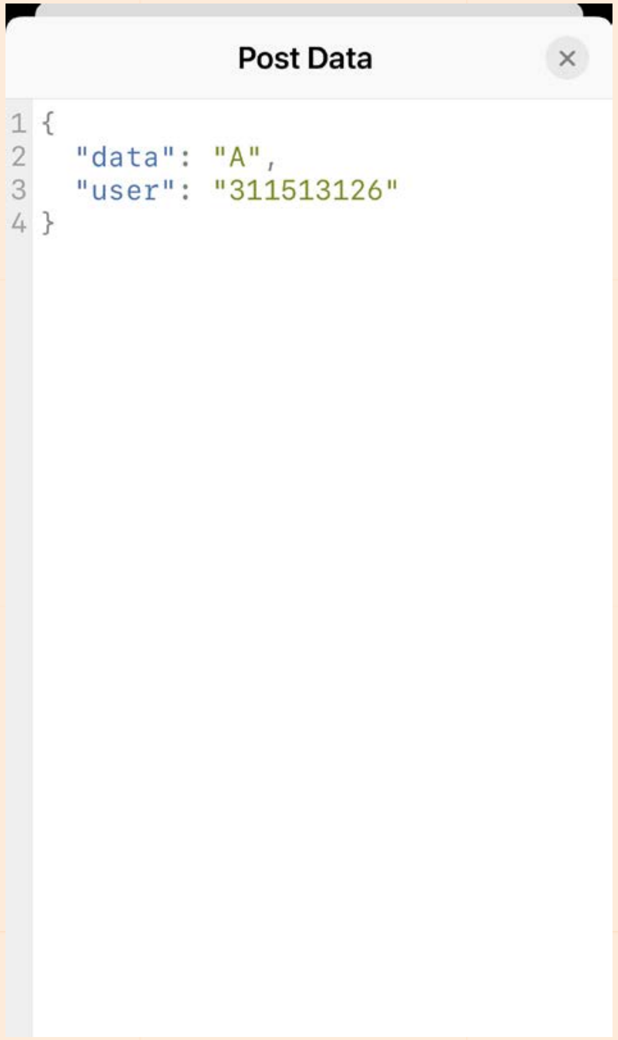


```
{
  "message": "Status Code = 400"
}
```

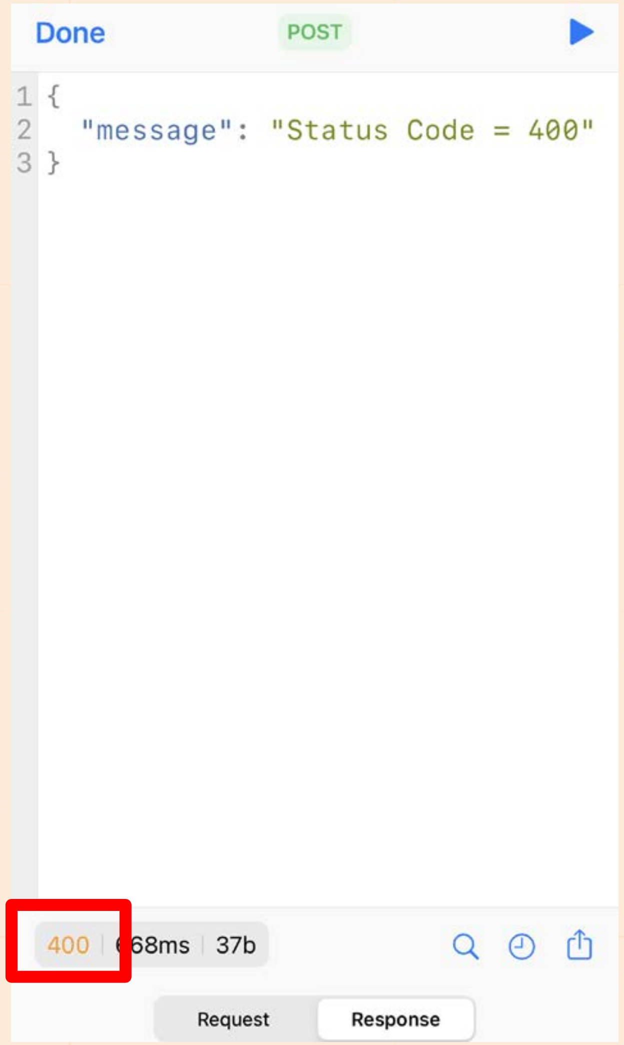
400 | 53ms | 37b

本次實驗 Demo

4. 非溫度或濕度


Post Data

```
{
  "data": "A",
  "user": "311513126"
}
```

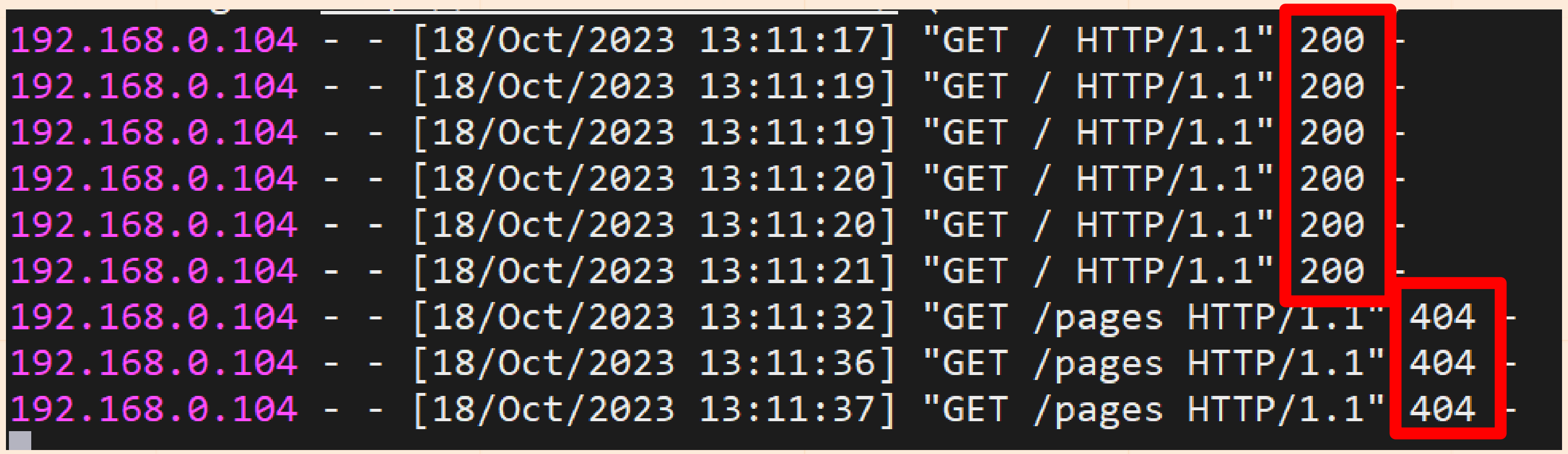


```
{
  "message": "Status Code = 400"
}
```

本次結報內容

- 1. 請附上本次實驗 Q1 的手機 (或PC端) 瀏覽器截圖、APP 內容的截圖，以及 Q2 APP 內容的截圖，並**詳細說明**為何在 Q1 中兩種 Demo 方式顯示的文字內容不太一樣。另外，在 PC 瀏覽器中進行何項操作可使兩者顯示相同文字內容 (即 APP 中的顯示方式) ?

- 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，請問紅色方框標示的數字分別代表著什麼意思?，請**詳細說明**。



```
192.168.0.104 - - [18/Oct/2023 13:11:17] "GET / HTTP/1.1" 200 -
192.168.0.104 - - [18/Oct/2023 13:11:19] "GET / HTTP/1.1" 200 -
192.168.0.104 - - [18/Oct/2023 13:11:19] "GET / HTTP/1.1" 200 -
192.168.0.104 - - [18/Oct/2023 13:11:20] "GET / HTTP/1.1" 200 -
192.168.0.104 - - [18/Oct/2023 13:11:20] "GET / HTTP/1.1" 200 -
192.168.0.104 - - [18/Oct/2023 13:11:21] "GET / HTTP/1.1" 200 -
192.168.0.104 - - [18/Oct/2023 13:11:32] "GET /pages HTTP/1.1" 404 -
192.168.0.104 - - [18/Oct/2023 13:11:36] "GET /pages HTTP/1.1" 404 -
192.168.0.104 - - [18/Oct/2023 13:11:37] "GET /pages HTTP/1.1" 404 -
```

本次結報內容

- 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? 請**詳細說明**原因。

  (請附上網頁瀏覽器截圖 與 Flask Web Service [即Rpi終端機] 訊息截圖)

  另外，請提供解決方法，於該網址恢復至顯示學號的頁面。

- 4. REST API 有哪些應用 ? (愈詳細且創新分數越高)

- 5. 本次實驗心得，你學到了什麼東西 ?

評分標準 & 注意事項

- 出席　　30 %
- Demo　30 %
- 結報　　40 %
- 請繳交 .pdf 檔，檔名為 學號_姓名_Labx.pdf
- 結報遲交一天，分數扣10%，以此類推

Reference

- https://ithelp.ithome.com.tw/articles/10191925
- https://medium.com/%E4%B8%80%E5%80%8B%E4%BA%BA%E7%9A%84%E6%96%87%E8%97%9D%E5%BE%A9%E8%88%88/python-flask-rest-api%E7%AD%86%E8%A8%98-869c3d2fee3
- https://ithelp.ithome.com.tw/articles/10202099?fbclid=IwAR1OSUaSK3yJj2FyyVCJQjfWtA_ucA8rPvpryamqs2E2QgnoQHwJ6nmuPQU
- https://ithelp.ithome.com.tw/articles/10222132
- https://ithelp.ithome.com.tw/articles/10157431