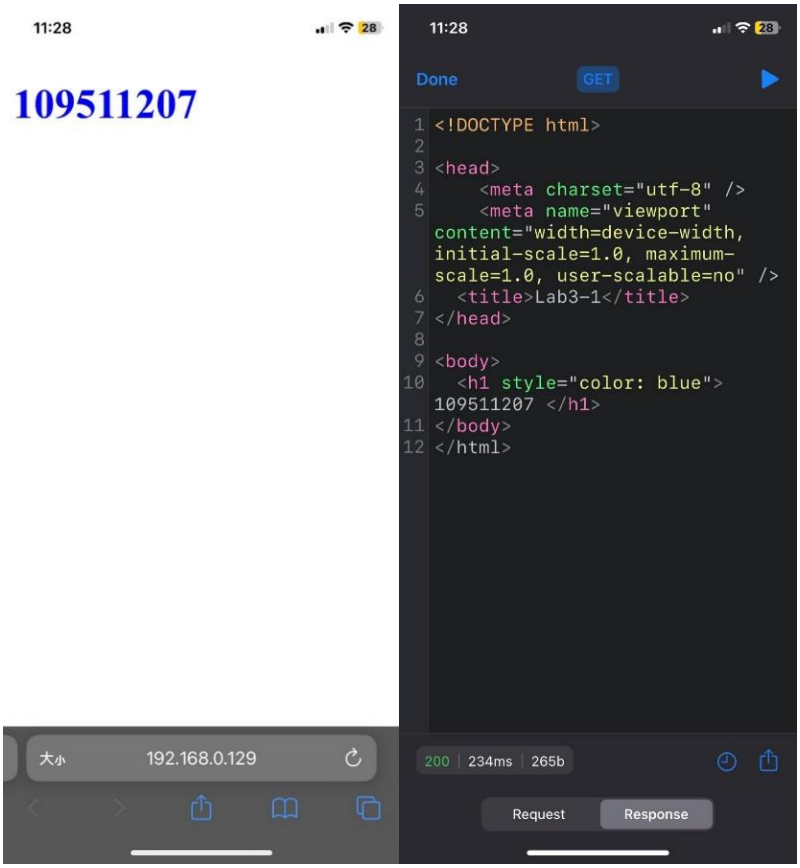
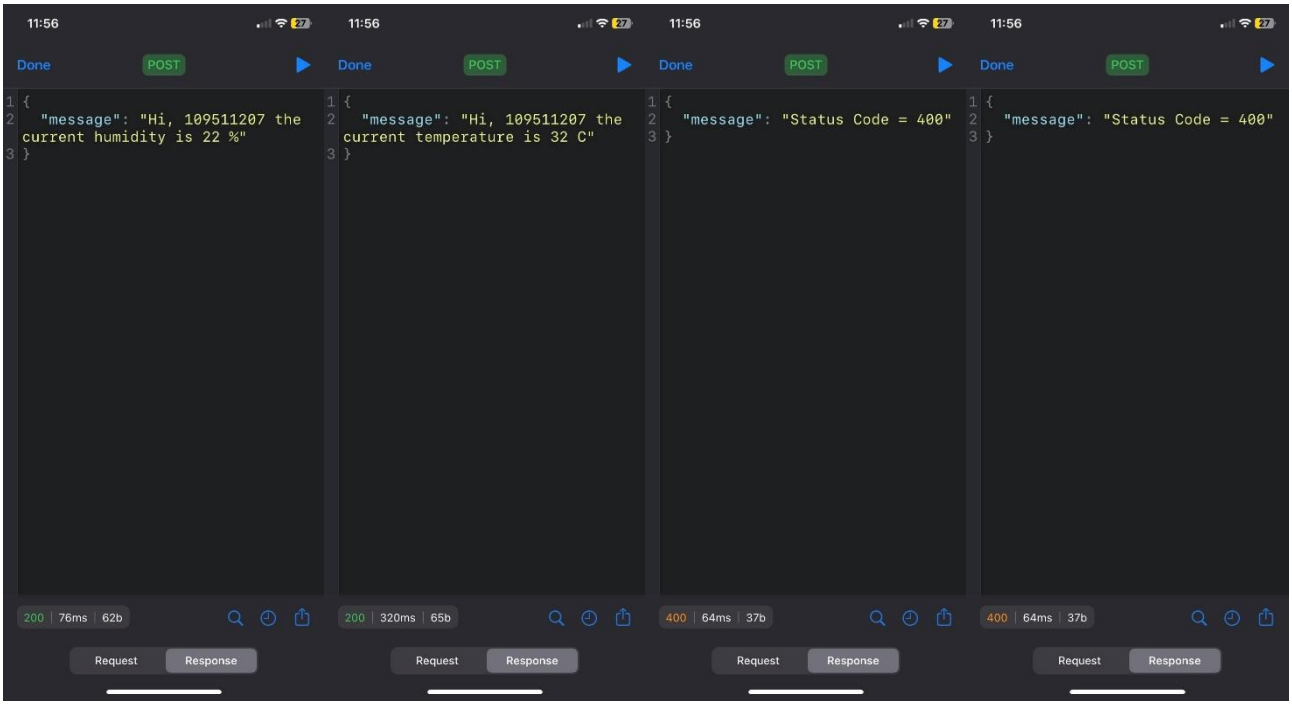
**1. 請附上本次實驗 Q1 的手機(或 PC 端)瀏覽器截圖、APP 內容的截圖，以及 Q2 APP 內容的截圖，並詳細說明為何在 Q1 中兩種 Demo 方式顯示的文字內容不太一樣。另外，在 PC 瀏覽器中進行何項操作可使兩者顯示相同文字內容(即 APP 中的顯示方式)？**

Q1




Q2



為何在 Q1 中兩種 Demo 方式顯示的文字內容不太一樣

因為手機 app 是直接顯示收到的東西，而我們打的 code 是 return 一個 html 回去，所以 app 顯示的是 html 網頁原始碼，而手機網頁或 PC 瀏覽器會對 html 解碼，所以才會發生兩個顯示的文字內容不一樣的問題。另外我在打 code 的時候發現，如果直接 return ID 的話，app 跟網頁就會都顯示 ID，這應該也是因為回傳的就是 ID，app 就會直接顯示 ID，而非 render_template()呼叫的 html 檔。

PC 瀏覽器中進行何項操作可使兩者顯示相同文字內容

可以在 PC 瀏覽器中按下右鍵，選擇檢視網頁原始碼，如下圖。

**109511207**


就可以得到以下 html 網頁原始碼。

```
<!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"> 109511207 </h1>
</body>
</html>
```

也可以在鍵盤上按下 F12，便可以得到以下 html 網頁原始碼。

**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 -
```

這些數字為 HTTP response 的 status code，用來描述 request 的結果。200 代表 request 成功，request object 稍後會被包在 message 中，status phrase 為 OK；404 代表在 server 上找不到 requested document，status phrase 為 Not Found。其他像是 400 代表 server 看不懂 request，status phrase 為 Bad Request；301 代表 object 被 server 移到其他地方了，server 會透過 location 這個 header line 告訴 client 新的路徑，status phrase 為 Moved Permanently。而要大致區分的話，2xx 代表成功；3xx 代表重定向；4xx 代表 client 端錯誤；5xx 代表 server 端錯誤。

**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。**

網頁瀏覽器截圖

# Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Flask Web Service 訊息截圖

```
pi@raspberrypi:~ $ sudo 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://192.168.0.167:9808/ (Press CTRL+C to quit)
192.168.0.149 - - [20/Oct/2023 08:25:57] "GET /pages HTTP/1.1" 404 -
192.168.0.149 - - [20/Oct/2023 08:25:58] "GET /favicon.ico HTTP/1.1" 404 -
```

瀏覽器會顯示 Not Found 訊息，而終端機會收到 Status Code 404，因為我們設定的 app.route() 並沒有'/pages'，自然也就在 server 上找不到這個 URL。解決的方法是在 code 中再加上一行 @app.route('/pages')，如下圖左。加上這行後網頁瀏覽器便能正常顯示學號了，如下圖右。

```
@app.route('/')
@app.route('/pages')
```

**109511207**

## 4. REST API 有哪些應用？(愈詳細且創新分數越高)

我覺得政府可以利用 REST API 來讓公民在網路上進行線上身分驗證，並讓公民能夠在線上參與總統大選、縣市長選舉或是公投等等公共事務，這樣較能刺激公民行使他們的公民權，不會因為自身忙碌原因而無法參與投票，帶來極大的便利性。

## 5. 本次實驗心得，你學到了什麼東西?

在這次 lab 中學到了 REST API。API 的全名是 Application Programming Interface，中文為應用程式介面，其充當著應用程式之間的橋樑，是由品牌開發出的一種接口，允許第三方可以額外開發、應用在自身產品上的系統溝通介面。API 可以幫助開發者節省精力，並快速達到目的。使用 API 的過程時，我們不需知道其內部程式運作的邏輯或演算法，只需告訴 API 所需的資訊，它就會給我們想要的結果。透過串接 API，公司可以購買或使用其他軟體，再把資料從原本內部的程式互傳到這個軟體上，這樣就可以減少公司開發的時間成本，加快流程的效率，間接的協助公司快速成長。

一個實際應用的例子是，在購物網站上，我們常常可以使用 FB 或 Line 登入，這背後就是透過 API 實現的。這允許購物網站將我們 FB 或 Line 的身份回傳回去，也就間接讓我們無須填寫一堆資料，即可註冊為會員，也就不用擔心會忘記密碼的問題，這對我們要註冊帳號帶來了極大的便利性。

而 REST 的全名是 Representational State Transfer，是一種網路架構風格，但並不是一種標準，REST API 與一般 API 的差異是，REST API 充分使用了 HTTP 的指令，用更簡單的方法操作，且有 HTTP 的優點，像是唯一的 URL 表示資源位置、統一的 API 接口、Stateless、Cacheable、分層系統架構以及 client-server 分離等優點。

這次的 lab 讓我認識了 REST API，並簡單地應用了它的特性(GET、POST)，也簡單複習了 HTTP protocol 的一些特性。

## 6. Reference


https://reurl.cc/kaovl3
https://www.hububble.co/blog/api
https://reurl.cc/WvOWYO
https://ithelp.ithome.com.tw/articles/10157431