

軟體工程師扶持計畫訓練營
Assignment - Week 7

延續上週作業建立的會員系統，運用前後端分離的思維，在會員頁中增添更多的功能。

要求一：後端建立查詢會員資料的 API

在資料庫中已經包含一些會員的資料，透過 Python Flask 後端建立 API 供前端使用。

查詢會員資料 API :<http://127.0.0.1:3000/api/users>

連線方法: GET

要求字串: username=要查詢的會員帳號

呼叫範例: <http://127.0.0.1:3000/api/users?username=ply>

回應格式: 按照以下規定的 JSON 格式，回應查詢結果

成功取得資料:

```
{
  "data": {
    "id": 3,
    "name": "強強",
    "username": "strong"
  }
}
```

沒有取得資料:

```
{
  "data": null
}
```

要求二：前端透過 **AJAX** 連接 **API** 查詢會員資料，並顯示在畫面上

會員頁面網址：<http://127.0.0.1:3000/member/>

參考設計如下：



歡迎光臨，這是會員頁

彭彭，歡迎登入系統

[登出系統](#)

查詢會員姓名

丁滿 (tintin)

必須滿足的運作流程：

1. 完成要求一，建立【查詢會員資料 API】
2. 使用者登入成功，在會員頁面中輸入要查詢的會員帳號，透過 JavaScript 的網路連線技術 (AJAX) 串接後端【查詢會員資料 API】，後端程式必須確實從資料庫中根據輸入的帳號查詢使用者姓名，並把查詢結果回應給前端。
3. JavaScript 程式取得查詢結果後，將資料顯示在會員頁面中。

要求三:完成修改姓名的功能 (Optional)

使用前後端分離的架構完成一個功能, 必須建立後端 API 以及前端的互動介面, 請按照以下要求完成修改會員姓名的功能。

修改會員姓名 API: <http://127.0.0.1:3000/api/user>

連線方法: POST

請求標頭:

標頭名稱	標頭內容
Content-Type	application/json

請求資料, 使用 **JSON** 格式處理:

```
{  
  "name": "新的使用者姓名"  
}
```

回應格式: 按照以下規定的 JSON 格式, 回應更新狀況

成功更新資料:

```
{  
  "ok": true  
}
```

更新資料失敗:

```
{  
  "error": true  
}
```

會員頁面網址: <http://127.0.0.1:3000/member/>

參考設計如下:

<p>彭彭, 歡迎登入系統</p> <p>登出系統</p> <p>查詢會員姓名</p> <div><input type="text" value="tintin"/> <input type="button" value="查詢"/></div> <p>丁滿 (tintin)</p> <p>更新我的姓名</p> <div><input type="text" value="彭彭彭"/> <input type="button" value="更新"/></div>	<p>彭彭彭, 歡迎登入系統</p> <p>登出系統</p> <p>查詢會員姓名</p> <div><input type="text" value="tintin"/> <input type="button" value="查詢"/></div> <p>丁滿 (tintin)</p> <p>更新我的姓名</p> <div><input type="text" value="輸入新的姓名"/> <input type="button" value="更新"/></div> <p>更新成功</p>
---	---

必須滿足的運作流程:

4. 建立【修改會員姓名 API】
5. 使用者登入成功, 在會員頁面中輸入新的姓名, 透過 JavaScript 的網路連線技術 (AJAX) 串接後端【修改會員姓名 API】, 後端程式必須確實根據前端的輸入, 更新資料庫中的使用者姓名, 並把更新狀況回應給前端。
6. JavaScript 程式取得更新結果後, 將結果顯示在會員頁面中。