



# PHP+Mysql 互動式網頁程式設計班

php Web API - 網站實例

鄔明學

The background of the slide is a soft, teal-colored image of a misty landscape. It features rolling hills or mountains in the distance, partially obscured by a light fog or mist. In the foreground, there's a calm body of water that reflects the surrounding scenery. The overall tone is serene and atmospheric, with a monochromatic teal palette.

先來探討我們目前所學的php

跟你平常上網，體驗到差異

# 目前待解的兩個議題

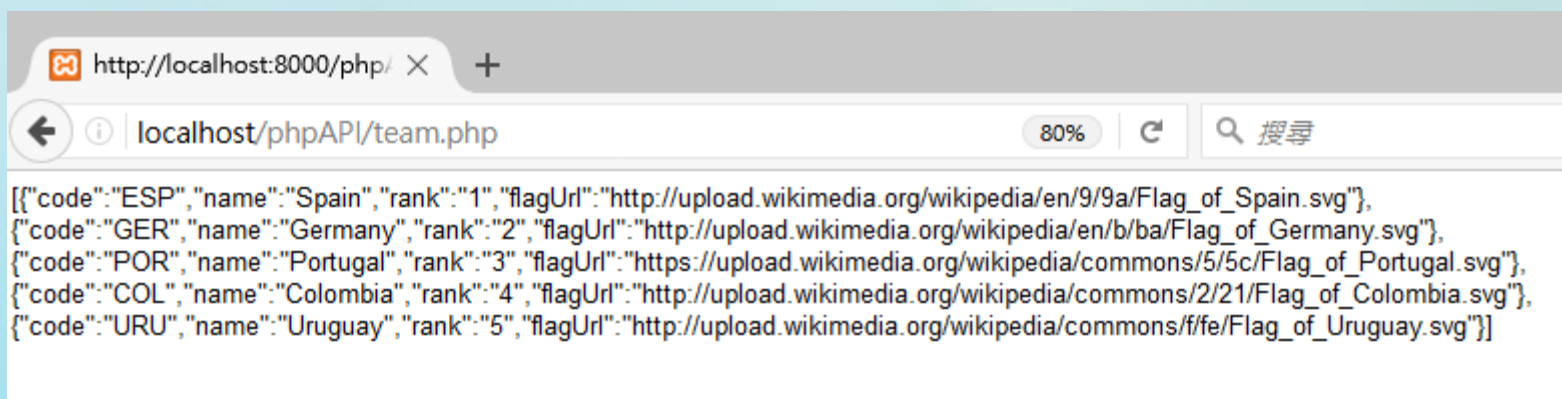
- 傳送表單的時候需要跳頁，跳到下一個頁面再由那支php處理資料。
  - 不跳頁，才潮阿
- 常常需要將html標籤 echo 到瀏覽器上，能不能php只做使用者與資料庫邏輯處理就好，不要再管任何html的標籤
  - 交給更專業的javascript 與css 來處理就好

# 讓 php 直接與 javascript 溝通

- 即所謂的 AJAX 技術，可用jquery、angularjs....等來輔助你完成
- 與過去一樣使用GET、POST傳送資料，但是要使用特定格式(JSON)
- php 透過GET、POST拿到資料後，處理完也要用JSON格式傳回資料
- 全部都靠JSON 格式的 字串來做溝通

# 幸與不幸

- php處理的事情單純化，最後只要把JSON格式印在瀏覽器就好



- Javascript透過網址，拿到JSON資料後再做處理呈現絢爛效果
  - 透過網址拿到資料後，就是有前端工程師搞定所有的畫面

# 前後端完全分割

- 後端: php 的程式可獨立放在另一個資料夾甚至是另一台網站伺服器
- 前端: 只含html、css、javascript，只專注在瀏覽器呈現的效果，需要資料就透過網址去要JSON
- SPA (Single Page Access) 用一個html檔案不跳頁來完成工作

# 前端世界

- 程式在使用者的瀏覽器上執行，那些程式會先去跟網站下載一份再到使用者的瀏覽器執行。 Html 、 css 、 javascript
- 變化非常快速，套件與工具玲朗滿目
- 建議靠 JQuery來輔助你完成一些工作吧 – 要先懂css selector
- 推薦框架： angularJS 、 Vue.js

# 後端世界

- 程式在伺服器上執行，透過GET、POST拿到資料後，想辦法生出正確的JSON字串在瀏覽器。
- \$\_COOKIE快沒人用了，現前端的世界都可以搞定
- PHP的工作: 想辦法跟資料庫溝通、檔案上傳到伺服器，檔案處理，登入帳號密碼的SESSION驗證，網路資料傳輸。
- 反正最後能要給正確的JSON字串



# JSON格式

- 在javascript 上其實就是javascript 的物件
- 而在php的世界裡比較像是平常在使用的php “陣列”
- JSON，物件用 {}，陣列用[]
- 一個名稱對應一個值
- 名稱記得要加 ""
- 名稱與值使用：來分隔對應
- 多組資料用，來區隔

```
1  {  
2      "code": "ESP",  
3      "name": "Spain",  
4      "rank": 1,  
5      "flagUrl": "http://www.google.com.tw"  
6  }
```

# JSON格式

- Javascript 取得物件{} 內的東西
  - 使用 .
- 取得 陣列裡面的東西用 [0]
- data.code 就會得到 ESP
- data.rank 就會得到 1
- data.flagUrl[0] 得到
  - <http://www.google.com.tw>
- data.flagUrl[2] 得到
  - <http://www.msn.com.tw>

```
1  var data = {  
2      "code": "ESP",  
3      "name": "Spain",  
4      "rank": 1,  
5      "flagUrl": "http://www.google.com.tw"  
6  }
```

```
1  var data = {  
2      "code": "ESP",  
3      "name": "Spain",  
4      "rank": 1,  
5      "flagUrl": [ "http://www.google.com.tw" ,  
6                  "http://www.yahoo.com.tw" ,  
7                  "http://www.msn.com.tw" ]  
8  }
```

# JSON格式

```
[{"code": "ESP", "name": "Spain", "rank": "1", "flagUrl": "http://upload.wikimedia.org/wikipedia/en/9/9a/Flag_of_Spain.svg"},  
{"code": "GER", "name": "Germany", "rank": "2", "flagUrl": "http://upload.wikimedia.org/wikipedia/en/b/ba/Flag_of_Germany.svg"},  
{"code": "POR", "name": "Portugal", "rank": "3", "flagUrl": "https://upload.wikimedia.org/wikipedia/commons/5/5c/Flag_of_Portugal.svg"},  
{"code": "COL", "name": "Colombia", "rank": "4", "flagUrl": "http://upload.wikimedia.org/wikipedia/commons/2/21/Flag_of_Colombia.svg"},  
{"code": "URU", "name": "Uruguay", "rank": "5", "flagUrl": "http://upload.wikimedia.org/wikipedia/commons/f/fe/Flag_of_Uruguay.svg"}]
```

- 假設現在PHP送回JSON在畫面，javascript 接到後用變數 data 儲存這些資料，在javascript 裡面要怎麼使用這些資料？
- data[0].code 得到 ESP ； data[0].rank 得到 1
- data[1].name 得到Germany ； data[1].rank 得到 2

# php 處理 json

- 回憶php的陣列

```
<?php
$data = [
    'dragon' => "恐龍遠古生物",
    'money' => "錢，拿來買東西",
    'car' => "車子，交通工具",
    'gun' => "武器具有殺傷力"
]
?>
```

- 若php的陣列轉換為JSON

```
{
    "dragon" : "恐龍遠古生物",
    "money" : "錢，拿來買東西",
    "car" : "車子，交通工具",
    "gun" : "武器具有殺傷力"
}
```

# Php轉換函數 - json\_decode()

- json\_decode( )      把接收到的json字串轉換成 php 陣列

```
<?php

$jsonString = '{
    "dragon" : "恐龍遠古生物",
    "money" : "錢，拿來買東西",
    "car" : "車子，交通工具",
    "gun" : "武器具有殺傷力"
}';

$dictionary = json_decode($jsonString , true);

var_dump($dictionary);

?>
```

# Php轉換函數 - json\_encode() 常用

- json\_encode()      php 陣列轉換成JSON字串

```
<?php

$jsonString = '{
    "dragon" : "恐龍遠古生物",
    "money" : "錢，拿來買東西",
    "car" : "車子，交通工具",
    "gun" : "武器具有殺傷力"
}';

$dictionary = json_decode($jsonString , true);

var_dump($dictionary);

?>
```


# Php轉換函數 - 解決中文亂碼

- `json_encode()` `JSON_UNESCAPED_UNICODE`

```
<?php

$dictionary = [
    "dragon" => "恐龍遠古生物",
    "money" => "錢，拿來買東西",
    "car" => "車子，交通工具",
    "gun" => "武器具有殺傷力"
];

echo json_encode($dictionary , JSON_UNESCAPED_UNICODE);
?>
```



# 深入php傳送資料議題



# \$\_SERVER

- `$_SERVER` 這個php 的陣列，裡面存放了瀏覽器與伺服器的資訊
- `$_SERVER[ 'REQUEST_METHOD' ]` 用什麼方式取得資料 GET 或是 POST
- `$_SERVER[ 'PATH_INFO' ]` 網址上的其他訊息

# `$_SERVER[ 'PATH_INFO' ]`

- 平常我們所看到的網址 `192.168.152.129/phpAPI/teamDetails.php`
- restful 網站的網址可以再傳參數到php檔 `192.168.152.129/phpAPI/teamDetails.php/GER`
- 透過 `$_SERVER[ 'PATH_INFO' ]` 可以取得 **/GER** 字串

# 傳送錯誤

- 如果使用者沒有登入或是SESSION 沒有紀錄，卻還是跑來需要登入的頁面
- 我們還是需要回送一個json到瀏覽器，同時在傳送的同時在背後告訴JavaScript 發生錯誤
- 故發生錯誤的時候必須要 加入 header()

```
header('HTTP/1.1 500 Internal Server');
```



帳密驗證

# 密碼安全性

- 帳號密碼與使用者資料都會存在於資料庫中
- 加強密碼安全性: 先將密碼經過編碼後再存到資料庫
- 編碼演算法: 輸入密碼會得到編碼後的暗碼，要從暗碼轉換回密碼近乎不可能，
- student 透過 sha256 編碼後可得  
264c8c381bf16c982a4e59b0dd4c6f7808c51a05f64c35db42cc78  
a2a72875bb

# 編碼演算法

- md5 早期使用的編碼演算法，現依舊還是有人使用
- sha256 : 比md5 更安全但複雜，個人偏愛
- sha512 : 又比sha512 更安全複雜
- 不同的編碼演算法對student編出的結果就會不一樣，選定一種使用就好
- php 函數
  - md5()
  - hash()
  - password\_hash()

```
<?php
md5($_POST['password']);           // 早期使用

hash('md5' , $_POST['password'])   // 速度較快

hash('sha256' , $_POST['password']) ;


password_hash($_POST['password'], PASSWORD_DEFAULT );

?>
```

# 登入程式

- 資料庫的密碼已經經過編碼，且轉換不回來。
- 此時就把使用者表單送來的密碼編碼後，在跟資料庫密碼做比較

```
// $userInputPwd = $_POST['password'];  
// $userInputPwd = md5($_POST['password']);  
// $userInputPwd = hash('md5' , $_POST['password']);  
$userInputPwd = hash('sha256' , $_POST['password']);  
// $userInputPwd = password_hash($_POST['password'] , PASSWORD_DEFAULT );
```

The background is a soft, teal-colored landscape. It features a calm body of water in the foreground, with gentle ripples. In the distance, there are misty mountains and hills, their details softened by the haze. The overall mood is serene and contemplative.

php -遺珠之憾

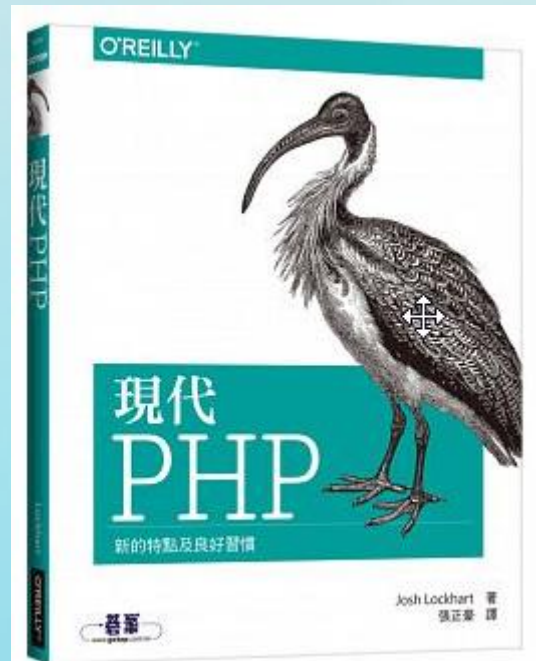


# 可繼續深耕的議題

- php 物件導向、元件、例外處理
- **composer** : 像手機上的appStore 一樣可以去網路取得並管理，別人寫好的php應用程式(元件)
- php測試: PHPUnit
- 學一套php MVC 框架：codeigniter ， **Laravel**

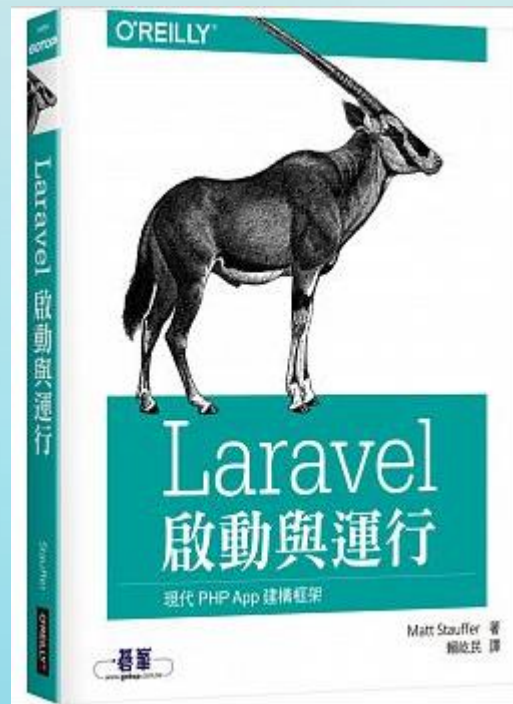
# php 推薦書籍

- 現代PHP



# php 推薦書籍

- Laravel



# 前端推薦書籍

- [AngularJS](#)
- [Html](#)
- JQuery (可不看)
  - [JavaScript & jQuery : The Missing Manual國際中文版\(第三版\)](#)
  - [JavaScript & JQuery：網站互動設計程式進化之道](#)

# 架站系統更新

- [Ngnix](#) : 比 apache 的 Http 網站伺服器要更快的 web 伺服器
- php7 : php5 直接改版後稱為7 速度大衍進，語法上細節也有些更動
- [Mariadb](#): 在mysql 被 Oracle 公司買走後，原團隊又再出來開發一套
- Xampp: 有整合php7 的版本包。

# 小叮嚀

- 若是覺得是站台環境問題程式無法好好執行
    - 確認php 版本、apache or nginx 、 mysql or mariadb
    - php.ini 檔案設定
    - phpinfo() 頁面來確認環境組態
    - 檔案與目錄是否有權限 (linux , mac)
  - 程式出錯
    - var\_dump() 、 print\_r() 確認陣列格式
    - 在前後行加 echo 看一下
    - sql 要先在 mysql 的工具執行過，再移過來php改寫
    - Debug：付費工具 phpStorm 、或是vscode 的 php debug套件
  - Google 是你的好朋友
- 記得要歸還門禁卡



thx