

HTTP通訊協定

關於網址Web Address (URL)

觀念說明

我們就是利用網址，組成資訊後，連線到後端特定的網路伺服器，取得那個網址所提供的服務與內容

而網站服務，其實是由各式各樣的網址所組成的

一個網址，代表的就是這個網頁的其中一個內容，或是其中一個API

網址的組成

通訊協定://主機名稱/路徑?參數名稱=資料&參數名稱=資料&....

組成介紹

- 通訊協定Protocol
  - http 或 https
- 主機名稱Hostname
  - www.youtube.com
- 路徑Path
  - 代表一個主機底下的不同服務
  - /watch
- 參數名稱

即網站前後端的互動基礎

互動邏輯

瀏覽器前端輸入網址後

透過通訊協定、主機名稱，連到對應的後端伺服器

後端伺服器根據路徑，去跑一些動作

將最後的結果回傳到我們的前端瀏覽器，並顯示出來

透過HTML：form標籤

特色說明

幫你收集使用者的輸入內容，並發一個連線到你指定的網址

會將整個畫面都換掉，重新刷新，進入新網頁

連線所提供的資料通常是完整的HTML程式碼

使用方法

搭配action屬性

輸入要接收連線的網址

`<form action="https://tw.search.yahoo.com/search">`

❗ 網址那邊要寫到路徑的部分，也就是？之前，不含？

涵括在form標籤時，會產生特殊意義/動作的標籤

- input
  - 搭配屬性
  - name
  - 設定網址中的「參數名稱」
  - `<input type="text" name="p"/>`
  - ❗ 「參數名稱」通常都是由後端決定的，沒有為什麼
- button
  - 點擊按鈕，會有「表單送出」的特別意義/動作在

```
<form action="https://tw.search.yahoo.com/search">  
<input type="text" name="p"/>  
<button>送出</button>  
</form>
```

如何建立前後端互動？

透過JS：AJAX技術

特色說明

非同步技術，不會整個畫面都需要刷新，而是在背景與後端做連線

只提供單純的資料而已，且大家習慣用JSON格式

實際操作上會使用fetch函式，來執行網路連線功能

使用方法

fetch函式

- 為JS內建的連線用函式
- 基本語法
  - fetch ("要連線的網址")
- 完整程式碼 (暫且用背的不論背後原理)
  - async function 函式名稱(){  
let 變數1=await fetch("要連線的網址")  
let 變數2=await response.json()  
}
  - async function getData(){  
let response=await fetch("https://cwpeng.github.io/live-records-samples/data/products.json")  
let data=await response.json()  
}

這是彭彭的網路課程學的，沒去特別查，但推估是偏舊的寫法，反正上面筆記的寫法更乾淨利落，用上面的就好

基本語法

fetch("要連線的網址").then(function(回應物件){console.log(回應物件)})

[程式邏輯細部說明]

純文字資料

fetch("網址").then(function(response){return response.text()}).then(function(data){console.log(data)})

JSON格式資料

fetch("網址").then(function(response){return response.json()}).then(function(data){console.log(data)})

伺服器會回應不同格式的資料，處理方式不同

[程式邏輯細部說明]