

API簡介

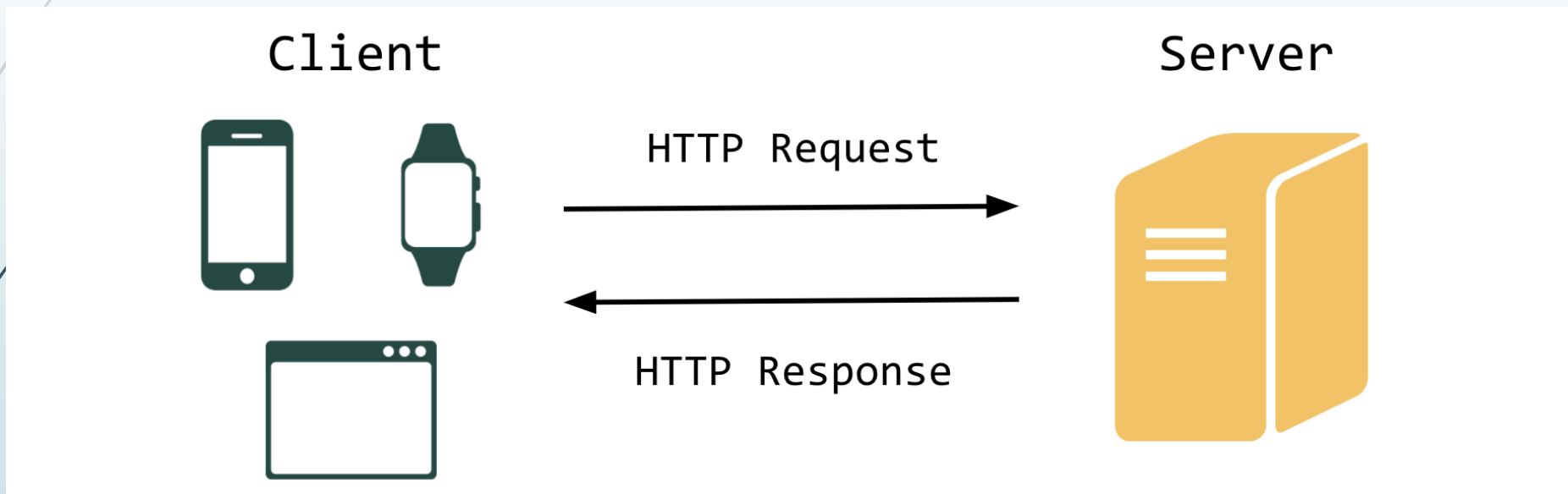
API簡介

1

- 應用程式介面（Application Programming Interface）
- 應用程式和應用程式之間，交換資訊的溝通橋樑
- 使用 API 時，不需要知道其內部程式運作的邏輯或演算法，只需提供 API 所需的參數，它就會回傳結果
- 使用 API 的好處
 - OpenSource 的時代，公家機關或大型公司往往會將手上的資料庫開放成 API
 - 從軟體開發的角度來看，近年來由於軟體開發日益複雜，將複雜的系統劃分成多個組成元件成為趨勢，讓工程師可能會分頭開發，再通過 API 的介面讓元件之間可以溝通，組合出完整的功能。
 - 在移動設備當道的時代趨勢中，API 也日益重要。一個軟體服務可能會在多種設備上被人使用，例如手機、瀏覽器、穿戴式設備等，這些裝置必須要和同一個後端資料庫互動，才可能呈現出一致的內容。

Web API

- 在 Web Application 的開發情境下的 API 被稱為 Web API，在 Web API 作用時，客戶端和伺服器端會透過 HTTP 通訊協定來進行請求與回應。



API – Client (客戶端)

- 串接他人開發的 Web API
- 作為開發者，你會常常需要使用第三方開發的 Web API：



API – Server (伺服器端)

- ➡ 開發 Web API 給他人使用
- ➡ 將資料庫的資料開放給授權的使用者查詢



API 開發者

定義 API 規格與撰寫文件

撰寫方法並開發 Web API



需要 API 的工程師

HTTP

- HTTP (HyperText Transfer Protocol) 網際網路上應用廣泛的一種網路協議
- 設計 HTTP 最初的目的是為了提供一種發佈和接收 HTML 頁面的方法
- 透過統一資源識別元 (Uniform Resource Identifiers, URI)，標識並識別了所有可以透過 HTTP 協定取得的資源位置。

`http://www.example.com/photos?page=1`

HTTP
通訊協定

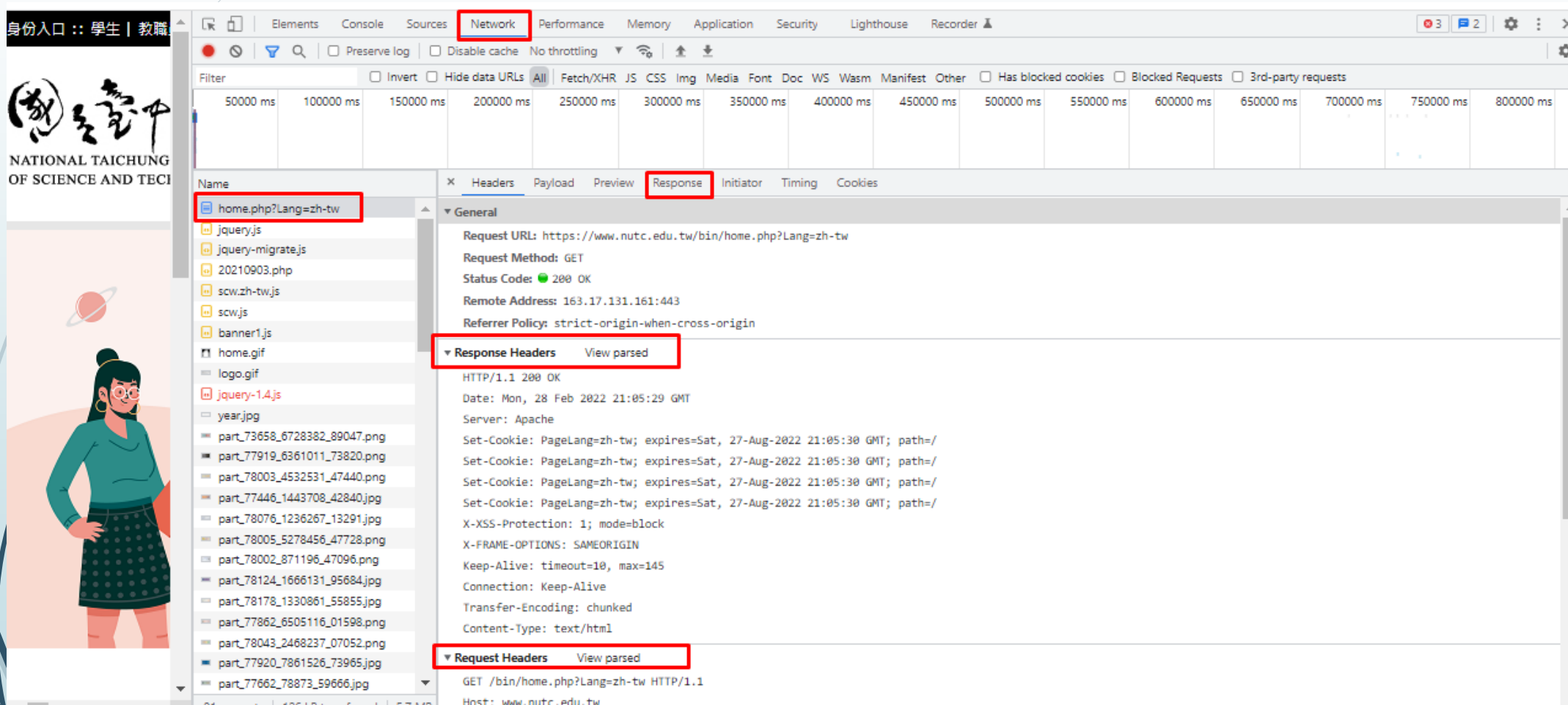
DNS 網域名稱

Path
路徑

Parameter
參數

觀察瀏覽器動作

➡ 右鍵 → 檢查 → Network(需要重新整理)



Request

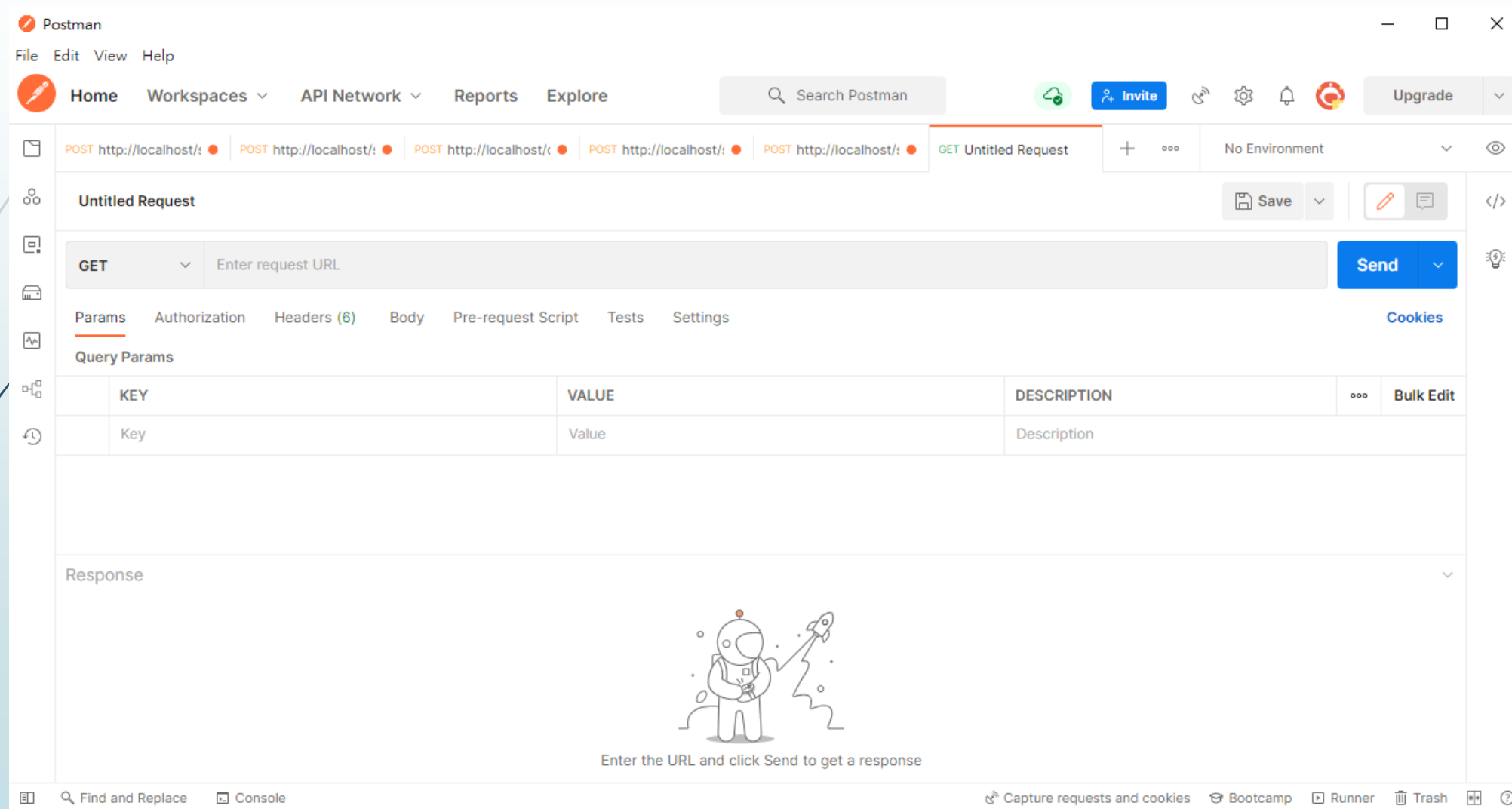
- Request 客戶端需要使用特定的 HTTP 動詞，對 URL 發出請求。

```
▼ Request Headers    View parsed
GET /bin/home.php?Lang=zh-tw HTTP/1.1
Host: www.nutc.edu.tw
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36
```


HTTP 動詞

- GET
- POST
- PUT
- PATCH
- DELETE

Postman



GET

10

- 讀取資源 (read only)，不應該改變所存取的資料
- 根據需要可被呼叫任意次數
- 透過 query string 來傳遞資料 (URL 可以被收藏與紀錄)

POST

- 建立新資源
- 不應被重複或被紀錄
- 對接收請求的伺服器做出修改
- 只有在請求會改變伺服器上的資料時使用

DEMO 1-1

12

- HTML使用form表單傳送資料到api程式接收並處理程式
- HTML
 - form
 - action
 - method
 - get
 - 網址?方式傳值
 - post
 - 屬性 name
 - input:submit
 - PHP
 - \$_SERVER['REQUEST_METHOD']
 - \$_GET / \$_POST

PUT

- 以新上傳的資源，來覆蓋掉本來存在的資源
- 視伺服器端的實作方式，有些開發者選擇用PUT 同時支援 PATCH 的動作

PATCH

- ➡ 修改資源的部份內容
- ➡ 視伺服器端的實作方式，有些開發者選擇用PATCH同時支援PUT的動作

➡ 刪除特定資源

DELETE

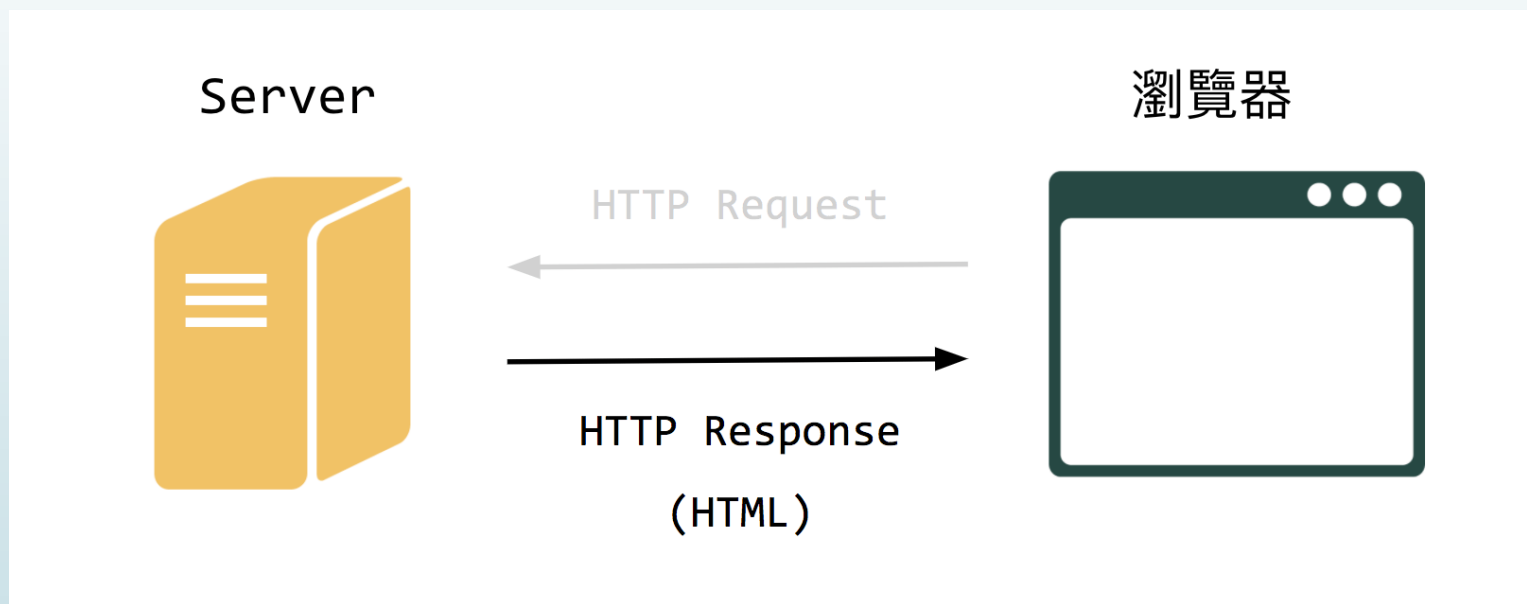
DEMO 1-2

16

- ➡ 因HTML form表單方法只能夠get及post方法
- ➡ 所以改用jQuery AJAX方式來傳送put、patch及delete方法
- ➡ HTML
 - ➡ jQuery Ajax
 - ➡ method: put、patch、delete
 - ➡ url: "api路徑"
 - ➡ data: data
 - ➡ PHP
 - ➡ file_get_contents('php://input')
 - ➡ parse_str

Response

- Response 伺服器端回應時採用的資料格式，可能會是：XML、JSON、Protocol Buffers、Thrift、YAML 等等。



Response Header

▼ Response Headers View parsed

```
HTTP/1.1 200 OK
Date: Mon, 28 Feb 2022 21:05:29 GMT
Server: Apache
Set-Cookie: PageLang=zh-tw; expires=Sat, 27-Aug-2022 21:05:30 GMT; path=/
Set-Cookie: PageLang=zh-tw; expires=Sat, 27-Aug-2022 21:05:30 GMT; path=/
Set-Cookie: PageLang=zh-tw; expires=Sat, 27-Aug-2022 21:05:30 GMT; path=/
Set-Cookie: PageLang=zh-tw; expires=Sat, 27-Aug-2022 21:05:30 GMT; path=/
X-XSS-Protection: 1; mode=block
X-FRAME-OPTIONS: SAMEORIGIN
Keep-Alive: timeout=10, max=145
Connection: Keep-Alive
```

Response 交換資料

- 在使用 Web API 交換資料時，目前最常見的回應格式有二
 - XML：Extensible Markup Language，是一種標記式 (markup) 語言，使用 <> 定義標記，標記的頭尾之間夾帶著內容，可以想像成是 HTML 標籤語法的延伸。
 - JSON：JavaScript Object Notation 的縮寫，它參考了 JavaScript 中物件結構的表示方式。

XML格式

- XML與HTML都屬於標記語言(Markup language)
- XML主要是「儲存和傳輸」資料
- XML的標記一定要兩兩相對(開始和結束)
- XML要自己定義標記
- XML標記中的屬性一定要用雙引號括起來
- XML可以透過標記，對傳輸的資料做描述
- XML是W3C組織制定的國際標準，不是某一家廠商的私有格式

```
<?xml version="1.0" ?>
- <ipo:purchaseOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema
  ipo.xsd">
-   <shipTo export-code="1" xsi:type="ipo:EU-Address">
       <name>Helen Zoe</name>
       <street>47 Eden Street</street>
       <city>Cambridge</city>
       <postcode>126</postcode>
     </shipTo>
-   <billTo xsi:type="ipo:US-Address">
       <name>Robert Smith</name>
       <street>8 Oak Avenue</street>
       <city>Old Town</city>
       <state>AK</state>
       <zip>95819</zip>
     </billTo>
-   <Items>
       - <item partNum="811-AA">
           <productName>Delicious Crab</productName>
           <quantity>10</quantity>
           <price>200</price>
           <shipDate>2011-10-10</shipDate>
         </item>
       - <item partNum="822-AA">
           <productName>Funny Toys</productName>
           <quantity>20</quantity>
           <price>300</price>
           <ipo:comment>Want this for the holidays!</ipo:comment>
           <shipDate>2011-10-12</shipDate>
         </item>
       - <item partNum="833-AA">
           <productName>Lapis necklace</productName>
           <quantity>30</quantity>
           <price>300</price>
           <ipo:comment>Go ahead. Make my day.</ipo:comment>
           <shipDate>2011-10-15</shipDate>
         </item>
     </Items>
  </ipo:purchaseOrder>
```



- **JSON (JavaScript Object Notation)**
- 以純文字為基底
- 簡單、輕量且可表示結構巢狀化的資料
- 相容性高且格式易於瞭解、閱讀及方便修改
- 幾乎所有程式語言都支援

JSON 格式

- 物件 { }
- 陣列 []
- name / value 是成對的，中間透過：來區隔
- value 可以是：
 - 數字 (整數 或 浮點數)
 - 字串 ("")
 - 布林值 (true 或 false)
 - 陣列 ([])
 - 物件 ({ })
 - NULL

```
1 {  
2   "name": "John",  
3   "age": 29,  
4   "employed": false,  
5   "car": null,  
6   "friends":  
7   [  
8     { "name": "Mary", "status": "close friends" },  
9     { "name": "Robert", "status": "close friends" },  
10    { "name": "Linda", "status": "acquaintance" }  
11  ]  
12 }
```

DEMO 1-3

23

- API端 – 將資料打包成json格式
 - json_encode
- 前端 – 接收JSON格式
 - eval("(" + data + ")");
 - \$.parseJSON()

HTTP 狀態碼

- 表示網頁伺服器回應狀態的 3 位數字代碼
- 狀態碼的第 1 個數字代表回應的 5 種狀態之一：
 - 1xx 訊息
 - 2xx 成功
 - 3xx 重新導向
 - 4xx 用戶端錯誤
 - 5xx 伺服器錯誤

2xx 狀態碼

- ➡ **200 OK**
 - ➡ 一切正常，回傳結果
- ➡ **201 Created**
 - ➡ 建立了新資源，通常 POST 完成後使用
- ➡ **202 Accepted**
 - ➡ 已收到 request 但未處理
- ➡ **204 No Content**
 - ➡ 完成，但沒有內容回傳，通常 DELETE 用

3xx 狀態碼

- ➡ **301 Moved Permanently**

- ➡ 永久轉址

- ➡ **302 Found**

- ➡ 臨時轉址

- ➡ **304 Not Modified**

- ➡ Client 要求該資源時，其內容並沒有變更，此時Client 僅需要使用本地快取 (Local Cache) 的副本即可

4xx 狀態碼

- **400 Bad Request**
 - 錯誤的 Request 內容，伺服器拒絕執行
- **401 Unauthorized**
 - 尚未登入
- **403 Forbidden**
 - 沒有足夠的權限
- **404 Not Found**
 - 查詢的資源不存在
- **405 Method Not Allowed**
 - 不支援該 URL 的存取方法

5xx 狀態碼

- ➡ **500 Internal Server Error**
 - ➡ 伺服器發生未預期的錯誤
- ➡ **502 Bad Gateway**
 - ➡ 伺服器端 Load Balancer 出現錯誤
- ➡ **503 Service Unavailable**
 - ➡ 應用程式暫停服務 (更新中或服務維護中...)

作業

待辦清單			
<input type="text" value="待辦事項"/>			<button>新增</button>
功能	事項	時間	狀態
	asdasdasd	2022-01-07 11:46:02	<button>完成</button>
	asdasd	2022-01-07 10:32:46	<button>完成</button>

伺服器: MySQL:3306 » 資料庫: todolist 資料表: todo									
瀏覽 結構 SQL 搜尋 新增 匯出 匯入 權限 操作 腳本									
#	名稱	型態	編碼與排序	屬性	空值	預設值	備註	額外資訊	
<input type="checkbox"/> 1	TODO_NO	int(20)			否	無		AUTO_INCREMENT	
<input type="checkbox"/> 2	TODO_LIST	varchar(500)	utf8_unicode_ci		否	無			
<input type="checkbox"/> 3	TODO_STATE	char(1)	utf8_unicode_ci		否	無			
<input type="checkbox"/> 4	TODO_TIME	timestamp			否	CURRENT_TIMESTAMP			