

Day 04

使用API 存取網路資料

HTTP 架構與 API 存取



出題教練：張維元



python

本日知識點目標

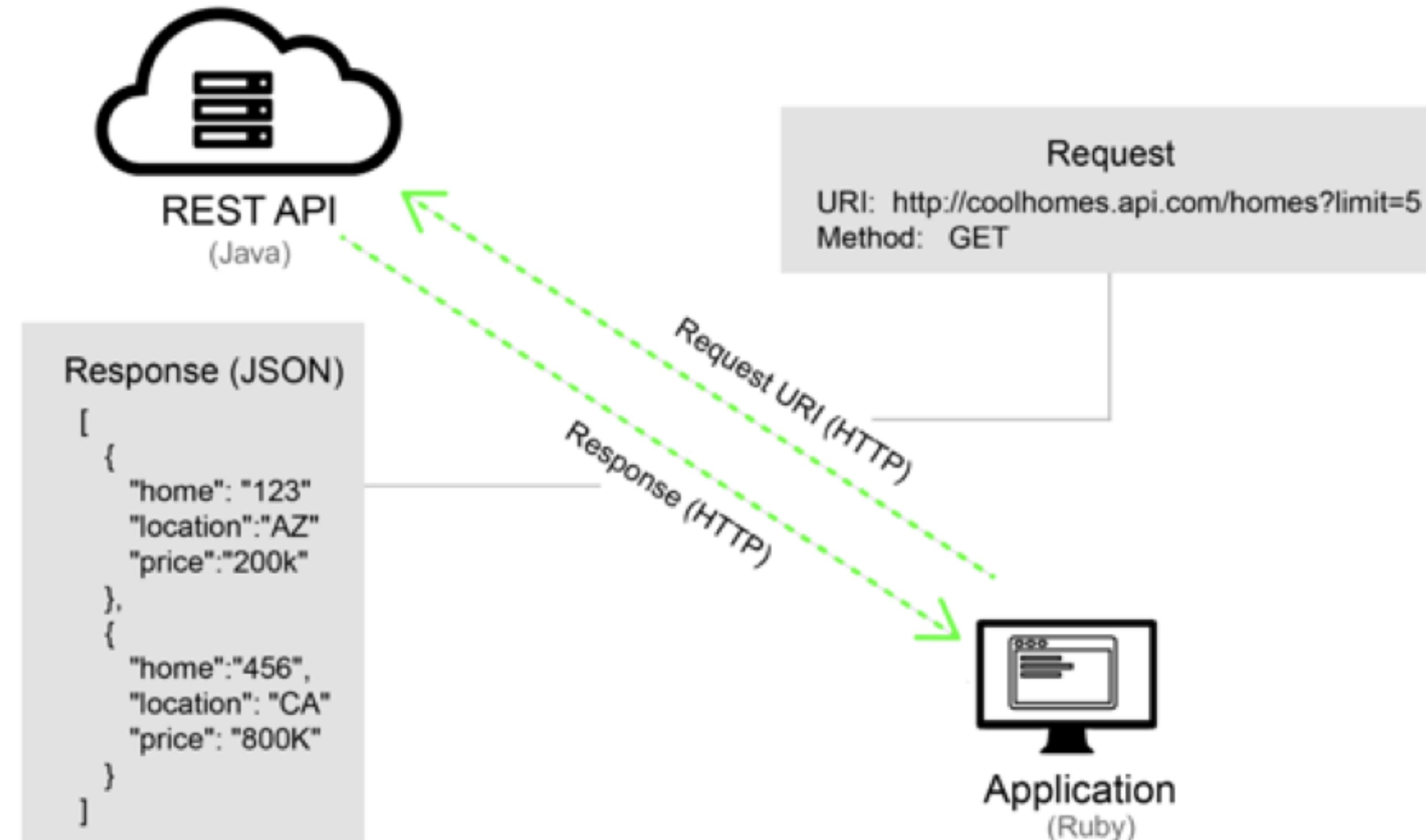
- 了解 Server Client 的架構與溝通方法
- 知道 HTTP Request & Response 的內容
- 什麼是 API？如何用 Python 程式存取 API 資料



HTTP 協定

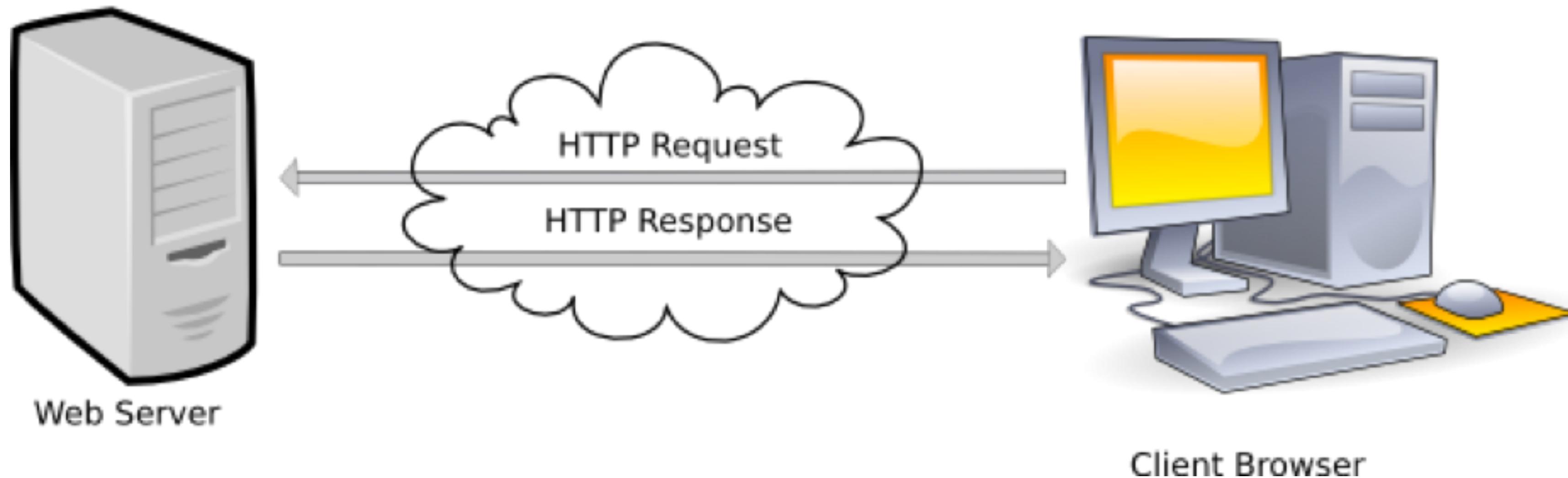
「HyperText Transfer Protocol(HTTP) 是一種用戶端瀏覽器和伺服端伺服器之間溝通的標準協定，規範了「客戶端」與「伺服器端」兩者間如何傳輸資料。」

HTTP 網頁架構 - Server - Client



Request & Response

HTTP 協定簡單來說就是：使用者端發送一個「Request 請求」，伺服器端根據請求回傳一個「Response 回覆」



Request 可以分成幾種方法

HTTP 將請求 (Request) 依據使用情境分成兩種方法：GET & POST。GET 方法將要傳送的資料以 Query String 的方式接在網址後面。Query String 是一種 Key-Value 的組成。

POST 方法則是將傳送的資料寫在傳送的封包當中，從網址上看不到。

- ① GET：將資料帶在網址上，利用 ?Key=Value 接在網址之後
- ② POST：用 form 的方式帶資料

Request 可以分成幾種方法

一般網頁網頁會利用 HTML 表單來發送請求，可以在傳資料的時候設置方法。

利用 GET 方法傳資料的表單

```
<form method="get" action="">  
  <input type="text" name="id" />  
  <input type="submit" />  
</form>
```

利用 POST 方法傳資料的表單

```
<form method="post" action="">  
  <input type="text" name="id" />  
  <input type="submit" />  
</form>
```

Request 可以分成幾種方法

利用 GET 方法傳資料的表單

```
<form method="get" action="">  
    <input type="text" name="id" />  
    <input type="submit" />  
</form>
```

利用 POST 方法傳資料的表單

```
<form method="post" action="">  
    <input type="text" name="id" />  
    <input type="submit" />  
</form>
```

利用 GET 方法發出的封包

```
GET /?id=010101 HTTP/1.1  
Host: xxx.cupuy.com  
...  
...
```

利用 POST 方法發出的封包

```
POST / HTTP/1.1  
Host: xxx.cupuy.com  
...  
id=010101
```

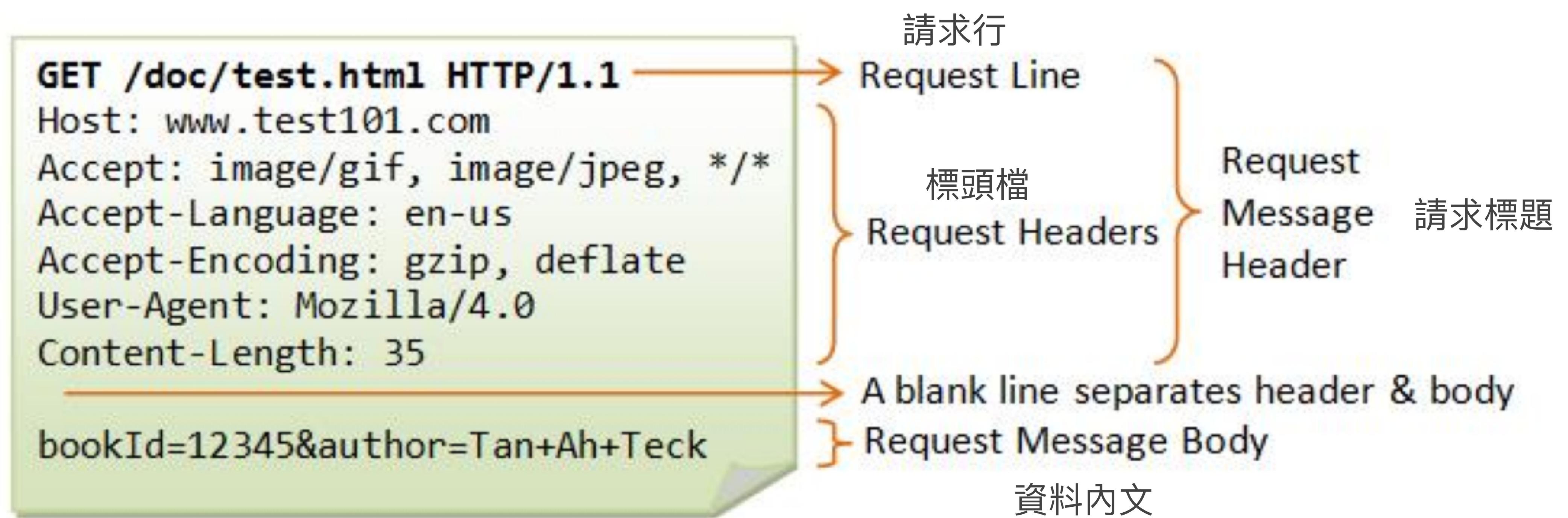
利用 GET 方法的網址

xxx.cupuy.com/?id=010101

利用 POST 方法的網址

xxx.cupuy.com/

Request 分為 Headers 與 Content



- Header 用來帶「發送方」相關的資訊
- Body 帶需要傳送給伺服器的資料

HTTP Method

HTTP 除了定義 GET 與 POST 之外，還有其他種方法。但是實務上大部分瀏覽器僅支援這兩種而已。

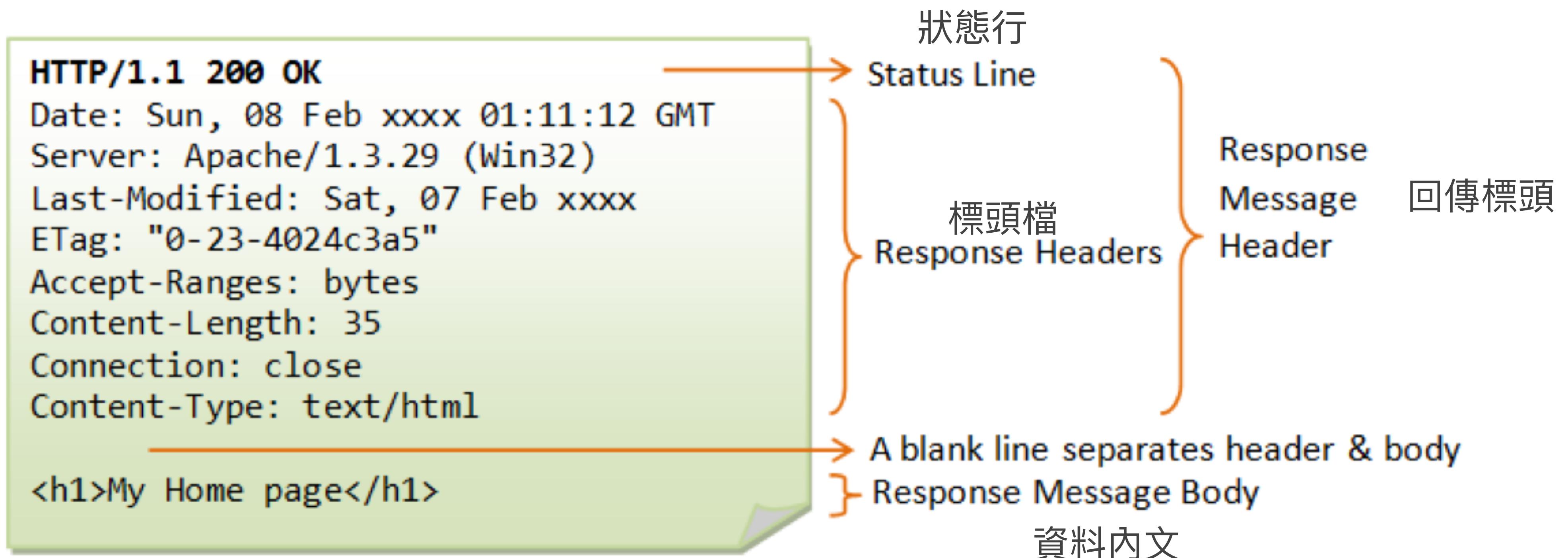
Method	Meaning
GET	Read data
POST	Insert data
PUT or PATCH	Update data, or insert if a new id
DELETE	Delete data

Response 有幾種方式

HTTP 的回應 (Response) 常見有兩種格式，API 與 HTML View。兩者的差別是回傳的 HTTP 內容不同。API 通常會以 JSON 格式而成的字串，是提供給程式存取。HTML View 是包含 JS、CSS、HTML 三種語言所組成的網頁原始碼，會交由瀏覽器進行處理。

- ① API => 回傳格式符合特定資料結構的字串
- ② HTML View => 回傳一段 html 網頁原始碼

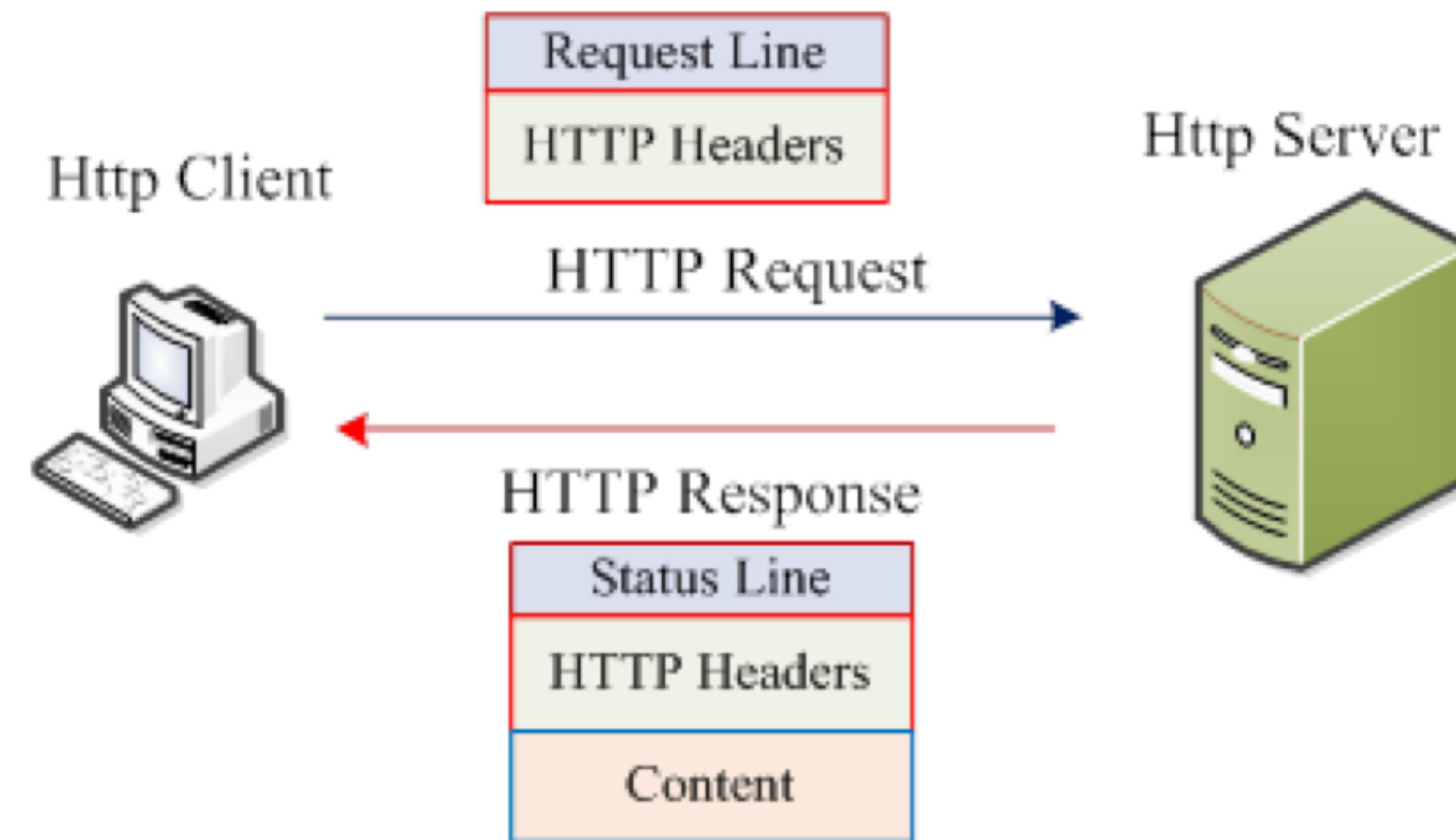
Response 分為 Headers 與 Content



- Header 用來帶「發送方」相關的資訊
- Body 則會帶回傳給瀏覽器的資料

Response 中會回傳請求的結果

在 HTTP Response 當中，除了跟 Request 一樣會有 Headers 跟 Content 之外，還會多帶一個狀態表示這次請求的結果。



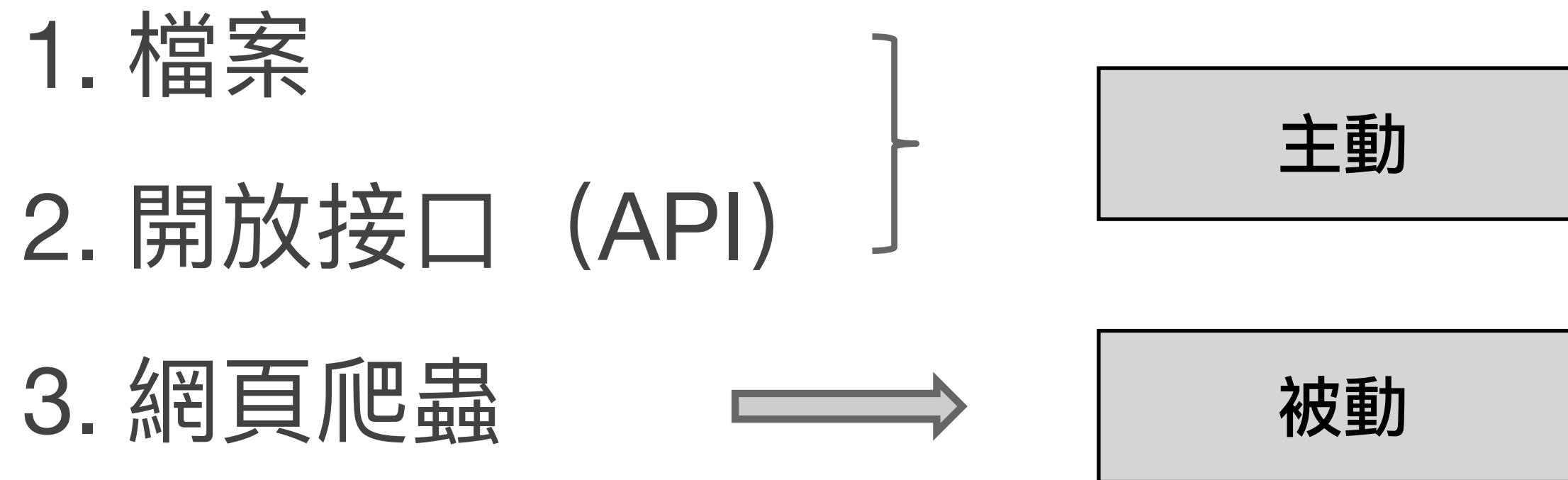
Response 狀態代碼

Response 定義了幾種不同的狀態，用來表示這次的請求結果：

200 OK	代表請求已成功被伺服器接收、理解、並接受
3XX Redirection	用來重新導向，後續的請求位址（重新導向目標）
4XX Error	用戶端看起來可能發生了錯誤，妨礙了伺服器的處理。
5XX Server Error	伺服器在處理請求的過程中有錯誤或者異常狀態發生

回顧：資料釋出的三個來源

資料的來源有三種管道：資料擁有者主動提供的「檔案」、「開放接口（API）」和被動公開的「網頁爬蟲」。



資料的來源方式很多，檔案 & API 是由資料擁有者主動釋出，爬蟲則是資料擁有者被動公開的。所以需要取得資料的時，通常會先考慮前兩者方法，真的無法才使用網頁爬蟲。

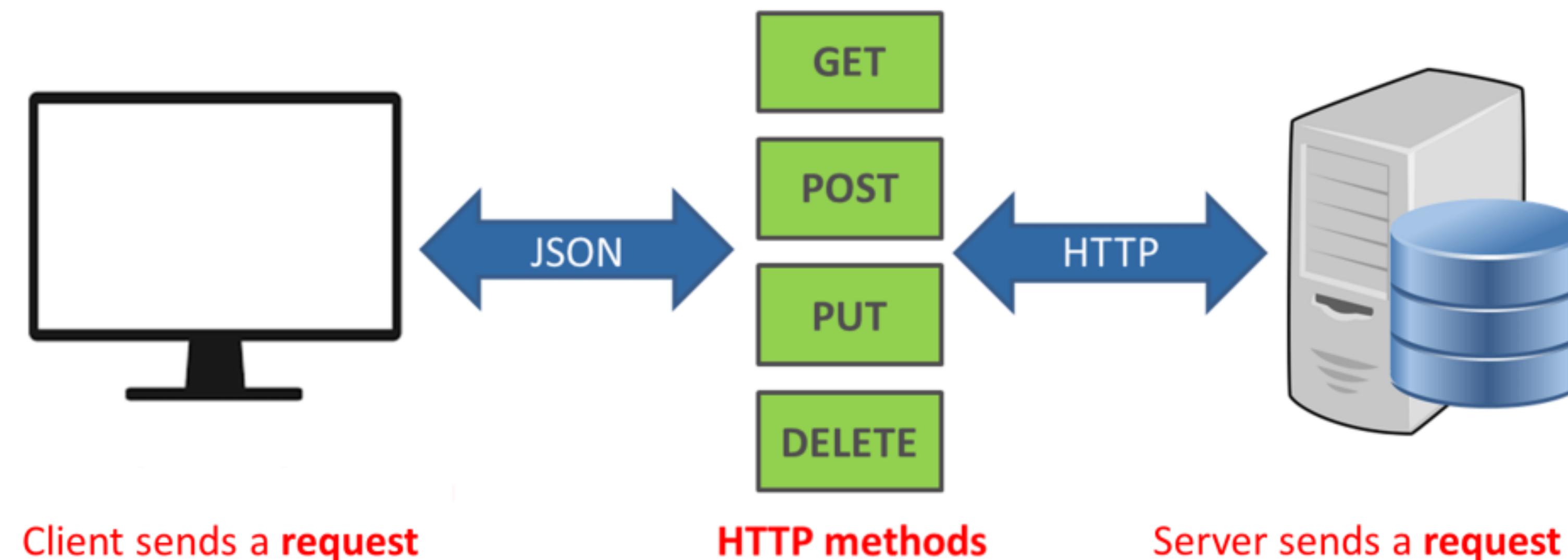


什麼是 API ?

第二種資料的發布方式是 API 的存取接口，API 的全名叫應用程式介面（Application Programming Interface），表示程式與程式間的溝通方式或介面。換句話說，就是資料方寫好一組程式，提供你可以用程式的方式與其串接資料。

RESTful API

Representational State Transfer (REST) , 它是一種網路架構風格，他並不是一種標準。 RESTful API 是基於 REST 的概念，進一步 結合 HTTP Method ，達到以直觀簡潔的方式存取資源。



RESTful API

一般的 API 設計

獲取使用者資料 GET /getAllUsers

獲取使用者資料 GET /getUser/1

新增使用者資料 POST /createUser

更新使用者資料 POST /updateUser/1

刪除使用者資料 POST /deleteUser/1

RESTful API

獲取使用者資料 GET /users

獲取使用者資料 GET /user/1

新增使用者資料 POST /user

更新使用者資料 PUT /user/1

刪除使用者資料 DELETE /user/1

RESTful API 將 API 的設計與 HTTP Method 結合

也就是說，不同的Method就是對同一件事情做不同的操作

思考流程與使用套件



Request Library

Requests 提供了基本的 HTTP Request 與接收 Response 的兩個核心功能。
Requests 的目標是使 HTTP 請求更簡單，更人性化。

```
1 import requests  
2  
3 # 引入函式庫  
4  
5 r = requests.get('https://github.com/timeline.json')  
6  
7 # 想要爬資料的目標網址  
8  
9 response = r.text  
10  
11 # 模擬發送請求的動作
```

重要知識點複習

- 了解 Server Client 的架構與溝通方法
- 知道 HTTP Request & Response 的內容
- 什麼是 API？如何用 Python 程式存取 API 資料



參考資料



- 淺談 HTTP Method：表單中的 GET 與 POST 有什麼差別？

◦ GET 與 POST 有寫過網頁表單的人一定不陌生，但你了解什麼是 GET 與 POST 嗎!? 現今的網頁設計工具相當的發達，甚至不需要接觸 HTML 語法就能完成一個規模不小的網站，漸漸地很多人都忘記了 HTTP 底層的實作原理，造成在發生錯誤的情況下無法正確進行偵錯。



- [不是工程師] 休息(REST)式架構？寧靜式(RESTful)的Web API是現在的潮流？
 - 介紹了 REST 架構及如何設計 RESTful API。
造成在發生錯誤的情況下無法正確進行偵錯。
- Public-apis
 - 收集了許多 API 的資源。

解題時間

LET'S CRACK IT

請跳出 PDF 至官網 Sample Code & 作業

開始解題

