

第2次作業題目-作業-QZ2

學號：112111113

姓名：林品承

作業撰寫時間：60 (mins，包含程式撰寫時間)

最後撰寫文件日期：2026/1/1

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- 說明內容
- 個人認為完成作業須具備觀念

1. HTTP Status Code 有哪些？怎麼分類？

Ans:

說明內容：**HTTP 狀態碼總共有分為 5 種類別分別是從 1xx 到 5xx，每一類都有代表著不同的回應情境**

1. 1xx 為資訊回應

※ 代表伺服器已收到請求，正在處理中，請客戶端繼續動作

常見的例如：

- 100 Continue：伺服器已收到請求的開頭部分，客戶端應繼續發送其餘部分。
- 101 Switching Protocols：伺服器同意切換通訊協定。

2. 2xx 為成功回應

※ 代表請求已經成功被伺服器接收、理解並處理

常見的例如：

- 200 OK：是最常見的表示請求成功，伺服器回傳了請求的資源（如網頁 HTML、圖片、JSON 等的資料）。
- 201 Created：請求成功，並且在伺服器上建立了一個新的資源（常用於 POST 請求）。
- 202 Accepted：伺服器已接受請求，但尚未處理完畢（常用於非同步任務）。
- 204 No Content：請求成功，但回應不含任何內容。

3. 3xx 為重新導向

※ 代表資源已經變動，客戶端需要採取進一步的動作（通常是跳轉到另一個 URL）才能完成請求

常見的例如：

- 301 Moved Permanently：永久更改移動。資源已被永久移到新網址。
- 302 Found：暫時移動。資源暫時被移到新網址，但舊網址未來可能還會用。

- 304 Not Modified：未修改（快取用）。資源自上次請求後沒變更過，瀏覽器可以直接使用本機快取，節省流量。

4. 4xx為客(用)戶端錯誤

※ 代表請求有錯誤或無法完成，問題有可能出在瀏覽器或發送請求的一方。可能是語法錯誤、網址打錯或權限的不足

常見的例如：

- 400 Bad Request：請求語法錯誤，伺服器看不懂。
- 401 Unauthorized：未授權，需要身分驗證。
- 403 Forbidden：禁止訪問。伺服器知道你是誰，但你沒有權限存取此資源。
- 404 Not Found：找不到。為最經典的錯誤，伺服器找不到你請求的資源（網址打錯或頁面已刪除）。
- 405 Method Not Allowed：方法不被允許。
- 429 Too Many Requests：請求次數過多，給定時間內發送了過多的請求。

5. 5xx為伺服器錯誤

※ 代表伺服器無法完成有效請求，問題出在「伺服器」。發送的請求是正確的，但伺服器端發生了異常

常見的例如：

- 500 Internal Server Error：內部伺服器錯誤。最通用的錯誤，伺服器程式碼崩潰了，但沒說具體原因。
- 502 Bad Gateway：錯誤閘道。通常發生在伺服器作為網關或代理時，從上游伺服器收到了無效回應。
- 503 Service Unavailable：服務無法使用。伺服器暫時過載或正在維護中，無法處理請求。
- 504 Gateway Timeout：網關逾時。伺服器作為網關，但上游伺服器回應太慢，導致逾時。

完成作業須具備觀念

需要學會認識及識別一些常會在網頁中所看見的 HTTP 狀態碼，當我們在遇到網頁並顯示這些數字代碼時，就能夠快速分辨每一種類開頭的數字各自代表的意涵大概是何種狀態，並加以去了解或尋找錯誤等問題；當我們自己在撰寫網頁程式時，也可以使用 Thunder Client 這個延伸模組，去執行看看輸出的狀態碼會顯示什麼樣的類別，協助我們確認程式的對錯，從而提高效率。

2. 在 Express 中，設計基本上可以分成幾層？請依上述回答實作一個後端與前端的網站（需有程式碼），並且將結果和執行畫面顯示於該份md，並逐步說明。

Ans:

※ Express 開發中，為了讓程式碼易於維護、擴充與除錯，通常會採用 MVC 或 分層式架構

(基礎)常見的 MVC 三層架構分別為：

1. Model (模型層)：處理資料邏輯。定義資料結構、驗證規則，與資料庫互動
2. View (視圖層)：呈現資料。負責前端顯示，可能是伺服器端渲染 (EJS、Pug) 或前後端分離時由前端框架處理。

3. Controller (控制層)：處理請求與回應。接收路由請求，調用 Model 取得資料，最後交給 View 或回傳 JSON。

(進階)三層架構分別為：

1. Controller (控制層)：

- 接收請求，監聽並接收 HTTP 請求
- 參數解析與驗證，檢查格式是否正確
- 呼叫 Service；回應結果，根據 Service 回傳的結果，決定回傳什麼 HTTP 狀態碼以及資料格式

2. Service (服務層 / 商業邏輯層)：

- 商業邏輯運算，執行核心演算法、規則判斷
- 資料組裝，若一個功能需要從多個資料表 (Tables) 取資料，Service 負責呼叫多個 DAO 並將結果組裝起來
- 交易控制，若操作涉及多個步驟，Service 層必須確保這些步驟正確與否

3. Dao/Model (資料層)：

- 與資料庫溝通，專門負責執行 SQL 語法
- 定義資料庫的 Schema，將資料庫取出的 Raw Data 轉換成程式語言中的物件
- 通常一個 DAO 方法只做一件單純的資料庫操作