1. 查詢所有的 HTTP 請求方法以及狀態回覆碼，並簡述至少 10 個請求方

法或狀態回覆碼之用途。

HTTP狀態碼-常用的

2xx系列(成功):

2XX系列為伺服器成功接收到用戶的請求,以及接受用戶的請求

|  |  |
| --- | --- |
| **狀態碼** | **敘述** |
| 200 | 用戶端要求成功，狀態ok |
| 201 | 已建立 |
| 202 | 以接受 |
| 203 | 非授權資訊 |

4xx系列(用戶端錯誤):

4XX系列為用戶端權限,或程式中發生的錯誤有關

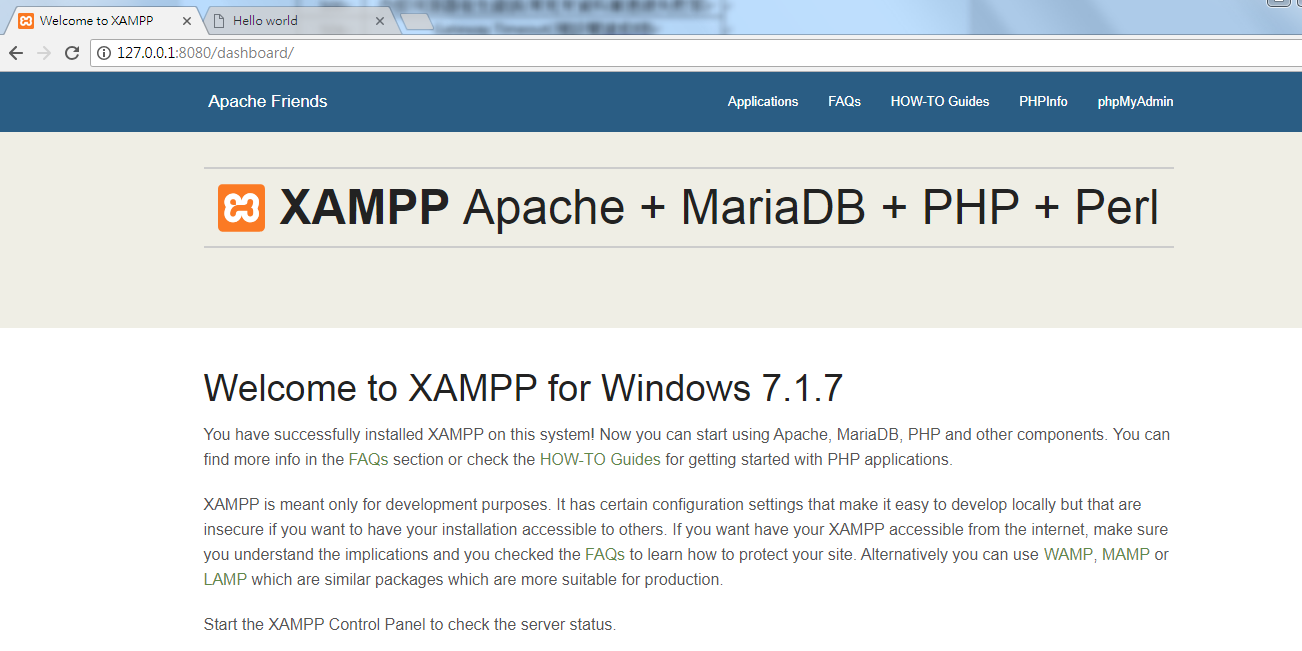
|  |  |
| --- | --- |
| **狀態碼** | **敘述** |
| 400 | 錯誤的要求(Bad Request) |
| 401 | Unauthorized |
| 403 | Forbidden(常見的也是權限問題) |
| 404 | 找不到頁面或資源(Not Found) |

5xx系列(伺服器錯誤):

5xx系列為伺服器端出錯,

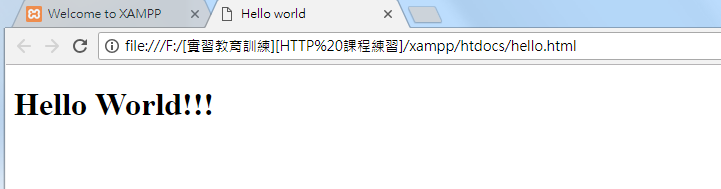
|  |  |
| --- | --- |
| **狀態碼** | **敘述** |
| 500 | 內部伺服器發生錯誤(常見有資料庫連線失敗等) |
| 504 | Gateway Timeout(預設閘道愈時) |

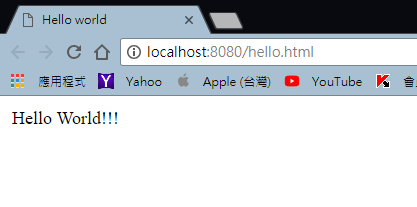
參考網址: <https://ithelp.ithome.com.tw/articles/10194999>

2.使用任一語言/套件（如：Apache）建立 HTTP 伺服器

3. 撰寫一網頁使其印出 Hello World 字串並放置於作業 2 所建立之 HTTP

伺服器所設定之網站目錄下，使其可以透過瀏覽器存取並呈現





4. 閱讀參考文獻 [1]，了解 HTTP2 協定與舊版差異，並嘗試修改作業 2 之 HTTP 伺服器，使其支援 HTTP2

HTTP/2 是如何改良

1.只需要單一網路連線（Single TCP connection），就可以連接網站伺服器，下載所有需要的資源。

2.檔案再多也不怕，不會發生佔用網路連線（TCP connection）的情況。

3.HTTP/2 用了 HPACK 壓縮技術，大大減少每次都要重複傳輸一樣的 Headers。

5. 查詢並簡述何謂 REST 網頁設計方式？使用時機以及優點為何？

Distributed Hypermedia Architect (分散式超媒體架構)有三個方式，有各自的優缺點：

**傳送者將資料** Render 後傳送給接受者，如此的方式對於接收者來說實作最簡單。但是由於傳送者的 Render 功能限制，系統的 Scalability (可擴展性) 相當的差。

**封裝資料**與 Render Engine 然後一起傳送給接收者。如此能夠有效的封裝資料，缺點為有可能造成過高的資料傳送成本。

**傳送原始資料** (RAW Data) 與 Metadata 给接收者，根據使用者可用的 Engine 自行 Render。優點為傳送資料的成本較低，缺點為對於資料的隱藏性較差，而雙方必須對 Data Type 有一定的共識。

REST 混合了這三種實作方法，但 REST 最主要的貢獻在於明確地抽離 Client 與 Server 的耦合性，透過一致性的介面進行溝通，大幅增加 Server 的可擴展性。