

# 標頭內容-

## Request Header

1. Host：瀏覽器存取的 Host 名稱，因一個 IP 可綁定多個不同 domain，此時需知道 Host 才知道要對應哪個伺服器。
2. User-Agent：瀏覽器名稱和版本 & 作業系統名稱和版本。
3. Accept：告訴伺服器瀏覽器接受的類型，但伺服器有最終決定權選擇傳送類型。
4. Accept-Language：瀏覽器接受的語言，q 值表示語言的優先權重。  
ex.zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7(第一順位中文、第二順位美語、第三順位英文)。
5. Accept-Encoding：瀏覽器支援的壓縮格式，大部分瀏覽器都支援 gzip 壓縮。伺服器會壓縮內容後傳給瀏覽器，節省下載時間和頻寬。
6. Date：傳送瀏覽器時間給伺服器，讓伺服器知道瀏覽器目前時間。
7. Expect (使用少)：要求伺服器回應特定結果，但最終還是由伺服器決定。
8. If-Modified-Since：瀏覽器 cache 檔案，下一次 request 則會發送此 header，伺服器使用這個時間來判斷是否修改過檔案，沒有則回傳 304。一般式：If-Modified-Since(hash code/時間)
9. Cookie：記錄目前瀏覽器記錄了哪些資訊。每一個 request 都會包含 cookie。
10. Referer：瀏覽器前一瀏覽頁面的網址，可用此 header 判斷 request 的來源，但不可靠。因為可能會遭到串改。
11. Connection：設定 Keep-Alive 保持連線時間，減少 TCP 三項交握的次數，提高傳輸效率。  
(HTTP 1.0 沒有這個 header)
12. Content-Type：傳送內容的格式 (multipart/form-data 檔案傳輸、application/x-www-form-urlencoded 表單資料傳輸)
13. Authorization：HTTP 連線的身分驗證

## Response Header

1. Etag：產生檔案的 Hash Code 給瀏覽器，下一次瀏覽器可傳送 if-None-Match 給伺服器判斷是否有修改過檔案，沒有則回傳 304 並且不包含 Content。
2. Last-Modified：回傳檔案最後修改的時間，下一次瀏覽器可傳送 if-Modified-Since 給伺服器判斷是否有修改過檔案，沒有則回傳 304 並不包含 Content。
3. Cache-Control：檔案的快取策略，max-age 則表示該檔案有效的 cache 秒數，若還有效就優先使用瀏覽器的快取 (HTTP 1.1 快取機制)，no-cache 則表示不使用快取。

4. Content-Type：回傳 content 的 mime 類型，當 Content-Type 設為 text/html 和 text/plain 可同時設定 charset 文字編碼格式，若無 Content-Type 瀏覽器自動判斷。
5. Content-Disposition：讓瀏覽器可以打開下載視窗並且指定檔案名稱。ex.  
Content-Disposition:attachment;filename="download.zip"
6. Content-length：取得下載傳送內容的大小，讓瀏覽器可判斷下載的速度。
7. Date：傳送伺服器時間給瀏覽器，讓瀏覽器知道伺服器目前的時間。**時差可透過 Request Header 和 Response Header 看出來。**
8. Location：當 status code 301/302 時，瀏覽器則會重新導向 Location 這個位置。
9. Set-Cookie：設定瀏覽器的 Cookie，當瀏覽器收到此 Header 則會建立 Cookie 在瀏覽器中。
10. Content-Encoding：對應 Request Header 的 Accept-Encoding，回傳內容的壓縮格式，讓瀏覽器知道該如何壓縮內容。
11. Access-Control-Allow-Origin：准許跨網域存取的 Domain，全部網域使用\*表示，script 和 link 不在此限。若沒有 Access-Control-Allow-Origin 無法跨網域存取對方 JSON(API)會出現無法跨網域存取的問題(CORS error)
12. X-Frame-Options：是否能夠在<frame>，<iframe>，<object>載入，網站可利用 X-Frame-Options 來確保本身內容不會遭惡意嵌入到其他網站，避免 Clickjacking 攻擊。
13. Upgrade：確認升級到新的協定，此時 Connection 也會同時回應 Upgrade。ex.Connection：Upgrade;Upgrade：websocket;

Response：伺服器 → 瀏覽器

Request：瀏覽器 → 伺服器

Header 一定在內容前面，傳送除了內容的資訊。內容不一定是 HTML，但 HTML 一定是內容之一，內容可能會有 JSON、圖檔、JS 等等。

## Status code

1. 1XX 訊息類（收到請求，請求者繼續執行操作）
2. 2XX 成功類（操作被成功接受並處理）
3. 3XX 重定向類（需進一步操作才能完成）
4. 4XX 客戶端錯誤類（請求語法錯誤或無法完成請求）
5. 5XX 伺服器錯誤類（後端的問題）

- 100 Continue：Server 期待收到更多資訊，以進行完整處理。ex.多半 Request 的 header 都有帶 Expect 要求回應 100
- 101 Switching Protocols：用在 websocket 初始化，然後切換到 TCP 連線進行雙向資料傳輸
- 200 OK：請求正常
- 301 Moved Permanently：永久移動到新的網址，同時需要有 Location 的 header，SEO 會被更新，權重也會更新
- 302 Found：找到網址，同時需要有 Location 的 header，SEO 不會被更新
- 304 Not Modified：內容未修改，回應不包含 Content。ex.網頁資料有快取過的話，第二次再發 Request，就會得到 304。
- 307 Temporary Redirect：暫時重新導向，與 302 類似但 302 如果從 POST 進來會改從 GET 導向網址，307 則不改變請求方式。
- 308 Permanent Redirect：永久重新導向，類似 301，同 307 不改變請求方式。
- 401 Unauthorized：未驗證拒絕存取。
- 403 Forbidden：伺服器接受請求，但拒絕處理。
- 404 Not Found：伺服器找不到要求的內容。
- 413 Request Entity Too Large：通常是使用 GET 傳送 URL 超過上限大小（2K），伺服器無法處理。
- 500 Internal Server Error：伺服器發生錯誤，通常是後端程式有問題。
- 503 Service Unavailable：伺服器維修／過載，伺服器當前無法處理請求。