B0929057_游少嫻_CORS 心得

跨來源資源共用（Cross-Origin Resource Sharing）又稱 CORS，是一種用其他 HTTP 開頭，讓目前的瀏覽頁面的使用者代理取得其餘網域伺服器特定資源權限的機制。當使用者代理要求一個不是當前文件來源，諸如相異網域（domain）、通訊協定（protocol）或通訊埠（port）中的資源時所產生一個跨來源的 cross origin HTTP 請求，而 CORS 就是用於讓網頁的受限資源能夠被其他網域頁面存取的一種機制。

其運作方式為新的 HTTP 頭部在瀏覽器具有權限的時候，應該以如何的形式傳送請求到遠端 URLs（統一資源定位符，Uniform Resource Locator 或稱統一資源定位器、定位位址，而 URL 位址就是俗稱網頁位址，即為網址，相當於在網際網路上標準資源的 Address）。雖然伺服器會有一些校驗和認證，但是瀏覽器有責任去支援這些標頭以及增加其他相關的限制。

而為什麼會有 CORS 狀況的產生，回到一開始，在學期初我們學到了 html 和 css 的基本架構，而如果網頁僅有美編使用的技巧，那僅能刻畫出靜態而不具互動效果的網頁。後期配合的 javascript 語言便是為了互動技能而鋪路。互動除了是與客戶端、使用者的行為模式，也有更多是需要與伺服器(server)交換 API(Application Programming Interface，應用程式介面)的資料串接連結。後者的問題則延伸出了拿資料的兩大層面，「跟誰」拿資料和如何「如何」拿資料。API 就像是一個程式間共通的資料讀取串接標準介面，當使用者需要存取其他網域的資料時，便需要找到其單位所提供的 API，與其說明文件來進行串接，文件內部便會提供諸如網址、參數說明。再且便會提到 Ajax(Asynchronous JavaScript and XML，非同步的 JavaScript 和 XML)，非同步的 JavaScript 意味著不用等 Ajax 結果執行完畢，本體程式仍可繼續執行，直到其操作完畢時再呼叫 Callback function(回傳函式)，並把資料帶入主體程式內。

在上述動作執行完畢時會遇到一個問題，「Same-origin policy」又稱同源政策。當現在的網站與呼叫的 API 是不同來源(http/https、網域不同)時瀏覽器仍會發 Request，但是會把 Response 擋下來，不讓 JavaScript 拿到並且傳回錯誤。但在此政策規範下還有另一條文便是 CORS，跨來源資源共用。當如果想開啟跨來源 HTTP 請求時便需要再 Response 的 Header 裡加上"Access-Control-Allow-Origin: * "，(星號代表萬用字源)，或是提到 Access-Control-Allow-Headers 跟 Access-Control-Allow-Methods 來定義資料來源。