Hw6-cors

系級:資工二甲　姓名:謝牧辰 學號:B0929055

CORS 是跨來源資源共用（Cross-Origin Resource Sharing）的縮寫，這個機制可以使用額外的 HTTP 標頭來讓目前瀏覽網站的使用者代理（en-US）取得存取其他網域伺服器特定資源的權限，當使用者代理請求一個不是目前文件來源——例如來自於不同網域（domain）、通訊協定（protocol）或通訊埠（port）的資源時，會建立一個跨來源 HTTP 請求（cross-origin HTTP request）。跨來源資源共用（Cross-Origin Resource Sharing，簡稱 CORS）機制提供了網頁伺服器跨網域的存取控制，增加跨網域資料傳輸的安全性。現代瀏覽器支援在 API 容器（如 XMLHttpRequest 或 Fetch）中使用 CORS 以降低跨來源 HTTP 請求的風險。基於安全性考量，程式碼所發出的跨來源 HTTP 請求會受到限制。例如 Fetch 及 XMLHttpRequest 都遵守同源政策（same-origin policy）。這代表網路應用程式所使用的 API 除非使用 CORS 標頭，否則只能請求與應用程式相同網域的 HTTP 資源。跨來源資源共用標準的運作方式是藉由加入新的 HTTP 標頭讓伺服器能夠描述來源資訊以提供予瀏覽器讀取。在瀏覽器上，如果你想拿到一個網站的完整內容（可以完整讀取），基本上就只能透過 XMLHttpRequest 或是 fetch。若是這些跨來源的 AJAX 沒有限制的話，你就可以透過使用者的瀏覽器，拿到「任意網站」的內容，包含了各種可能有敏感資訊的網站。因此瀏覽器會擋跨來源的 AJAX 是十分合理的一件事，就是為了安全性。這時候有些人可能會有個疑問：「那為什麼圖片、CSS 或是 script 不擋？」因為這些比較像是「網頁資源的一部分」，例如說我想要用別人的圖片，我就用 <img> 來引入，想要用 CSS 就用 <link href="...">，這些標籤可以拿到的資源是有限制的。再者，這些取得回來的資源，**我沒辦法用程式去讀取它**，這很重要。我載入圖片之後它就真的只是張圖片，只有瀏覽器知道圖片的內容，我不會知道，我也沒有辦法用程式去讀取它。既然沒辦法用程式去讀取它，那我也沒辦法把拿到的結果傳到其他地方，就比較不會有資料外洩的問題。有很多人認為：「跨來源請求擋住的是 request」但這個說法其實想一下就知道有問題，你看錯誤訊息就可以知道：

request has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource

瀏覽器說沒有那個 header 存在，就代表什麼？代表它已經幫你把 request 發出去，而且拿到 response 了，才會知道沒有 Access-Control-Allow-Origin 的 header 存在。所以瀏覽器擋住的不是 request，而是 response。你的 request 已經抵達 server 端，server 也回傳 response 了，只是瀏覽器不把結果給你而已。