網頁程式設計 hw6-cors

系級：資工二甲　　姓名：林芊妤　　學號：B0929034

在串接後端或是網路上的 API 時，時常會出現cors的問題，其原因是因為跨來源呼叫 API。所謂跨來源(cross origin)就是想從來源A去拿來源B的東西， 而origin分別是protocol + domain + port 的組合，必須三個都相同才能夠稱為同個來源。舉例來說，protocol通常為http或https，domain則是網域，port在沒有特別指定下http預設是80，https則是443。

之所以要擋住跨來源的AJAX，是因為有安全性的問題，在瀏覽器上，如果想拿到一個網站的完整內容，只能透過 XMLHttpRequest 或是 fetch。若是這些跨來源的 AJAX 沒有限制的話，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。

當發生cors錯誤時，問題是出在response而不是request，會有 same-origin policy 跟 CORS，是因為我們在瀏覽器上寫 JS，所以受到執行環境的限制。如果我們今天寫的是 Node.js，就完全沒有這些問題，想拿什麼就拿什麼，不會有人擋我們。瀏覽器因為安全性的考量所以會把東西給擋住，因此必須要讓瀏覽器知道這其實是安全的，它才會放行。關於跨域請求，解決方案有不少種，例如JSONP，也就是透過 HTML中沒有跨域限制的標籤如 img、script 等，再藉由指定 callback 函式，將回應內容介接回 JavaScript 中；或是透過 iframe，繞過跨域保護取得目標資源等等。

其中，最標準、正確的解決方法是透過W3C規範的跨來源資源共用，透過伺服器在 HTTP Header 的設定，讓瀏覽器能取得不同來源的資源。由於後端才是擁有權限的，若要存取該資料，必須讓此後端在response加上如 Access-Control-Allow-Origin、Access-Control-Request-Method、Access-Control-Request-Headers 等設定。當瀏覽器發送資源請求時，如果是簡單請求，便會直接送出請求；若不符合前述條件，則會透過預檢（Preflighted）請求先敲敲門，確認可以通過伺服器限制，才會發送正式的請求。由此一來，瀏覽器便知道此資料允許某個origin去做存取的動作。

然而，若是沒辦法要求後端加上header的話，就必須透過ajax proxy server。proxy server就像是藝人與經紀人一樣，對外的工作都是經紀人負責接洽，談完以後才告知藝人。而藝人如果想找誰合作，也是讓經紀人去問，問完再跟藝人說，所以經紀人其實就是藝人明星的代理人。如果你想拿 A 網站的資料，但是它沒有提供Access-Control-Allow-Origin這個header，就可以自己寫個server，從後端去拿A網站的資料，再把資料丟回給自己的前端。