HW6-CORS 心得文章

跨來源資源共用（Cross-Origin Resource Sharing（CORS））是一種使用額外 HTTP 標頭令目前瀏覽網站的使用者代理取得存取其他來源（網域）伺服器特定資源權限的機制。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、通訊協定或通訊埠的資源時，會建立一個跨來源 HTTP 請求（cross-origin HTTP request）。透過 JavaScript 存取非同源資源時，server 必須明確告知瀏覽器允許何種請求，只有 server 允許的請求能夠被瀏覽器實際發送，否則會失敗。

CORS 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。另外，針對會造成副作用的 HTTP 請求方法（特別是 GET 以外的 HTTP 方法，或搭配某些 MIME types 的 POST 方法），規範要求瀏覽器必須要請求傳送「預檢（preflight）」請求，以 HTTP 的 OPTIONS（en-US）方法之請求從伺服器取得其支援的方法。當伺服器許可後，再傳送 HTTP 請求方法送出實際的請求。伺服器也可以通知客戶端是否要連同安全性資料（包括 Cookies 和 HTTP 認證（Authentication）資料）一併隨請求送出。

在 CORS 的規範裡面，跨來源請求有分兩種：「簡單」的請求和非「簡單」的請求。所謂的「簡單」請求，必須符合以下兩個條件： 1.只能是 HTTP GET, POST or HEAD 方法 2.自訂的 request header 只能是 Accept、Accept-Language、Content-Language 或 Content-Type（值只能是 application/x-www-form-urlencoded、multipart/form-data 或 text/plain）。只要不符合以上任一條件的請求就是非簡單請求。

一般的 http request 會帶有該網域底下的 cookie；然而，跨來源請求預設是不能帶 cookie 的。因為帶有 cookie 的請求非常強大，如果請求攜帶的 cookie 是 session token，那這個請求可以以你的身份做很多的事情，像是存取你的隱私資料、從你的銀行帳戶轉帳等。所以瀏覽器端針對跨來源請求的 cookie 也做了規範。請求必須要明確地標示「我要存取跨域 cookie」。使用 fetch API 和 XMLHttpRequest 透過 fetch API 發送跨來源請求，需要定 credentials: 'include'：透過 XMLHttpRequest 發送跨來源請求，需要定 withCredentials = true；如此一來跨來源請求就會攜帶 cookie 了。

總之，遇到 CORS 的問題，可以歸納出這樣的 SOP： 1.先認清楚是否為「簡單」的跨來源請求，如果是，在後端 GET/POST/HEAD 方法本身加上 Access-Control-Allow-Origin header。 2. 如果非「簡單」跨來源請求，在後端 OPTIONS 加上 Access-Control-Allow-Methods 及 Access-Control-Allow-Headers header。另外，在後端方法本身加上 Access-Control-Allow-Origin header。 3. 需要使用 cookie 的情況下，前端要加上 credentials: 'include' 或是 withCredentials 參數，後端要加上 Access-Control-Allow-Credentials header，而且 Access-Control-Allow-Origin header 不能用 *。