網頁設計HW6-cors

資工系 B0929026 林哲佑

簡單地說，CORS (Cross-Origin Resource Sharing) 是針對不同源的請求而定的規範，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。跨來源資源共用標準的運作方式是藉由加入新的 [HTTP 標頭](https://developer.mozilla.org/zh-TW/docs/Web/HTTP/Headers)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，規範要求瀏覽器必須要請求傳送「預檢」請求，以 HTTP 的 OPTIONS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「預檢」的請求。請求不會觸發 [CORS 預檢](https://developer.mozilla.org/zh-TW/docs/Web/HTTP/CORS#preflighted_requests)。這類請求被稱作「簡單請求（simple requests）」，雖然 [Fetch](https://fetch.spec.whatwg.org/) 規範（其定義了 CORS）中並不使用這個述語。不同於上面討論「[簡單請求](https://developer.mozilla.org/zh-TW/docs/Web/HTTP/CORS#%E7%B0%A1%E5%96%AE%E8%AB%8B%E6%B1%82)」，「預檢（preflighted）」請求會先以 HTTP 的 OPTIONS 方法送出請求到另一個網域，確認後續實際（actual）請求是否可安全送出，由於跨站請求可能會攜帶使用者資料，所以要先進行預檢請求。

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，需要設定 credentials: 'include'：

透過 XMLHttpRequest 發送跨來源請求，需要設定 withCredentials = true;

如此一來跨來源請求就會攜帶 cookie 了。

遇到 CORS 的問題，可以歸納出這樣的 SOP：

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。

如果「預檢」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。

(Optional) 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 \*。