跨來源資源共用（CORS）

B0929060 張孟佳

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（US)取得存取其他來源（網域）伺服器特定資源權限的機制。

為什麼需要 CORS？因為安全性的考量，瀏覽器預設都會限制網頁做跨網域的連線。但如果要提供資料存取的服務給其它人使用，就必須要開放對應的 API 給其它人連線。而 CORS 就是一個瀏覽器做跨網域連線的時要遵守的規範。

什麼是跨來源？當一個網頁送出一個 request 的時候，瀏覽器會在 XMLHttpRequest 的 header 塞入 Origin 這個資料代表這個網頁的來源。Origin 通常就是這個網頁對應網址的網域，也就是網頁發出的 request 必須與原本的網頁要有相同的來源(the same-origin policy)。當一個 request 的 Origin 網域與 request 的目標伺服器不同，就是所謂的跨網域連線。

跨來源請求又可以分成「簡單請求（simple request)」和「預檢請求（preflight request)」。前者基本上使用的一定要是 GET、HEAD、POST 方法，並且不能有客制的 header，僅允許特定的標頭和內容，如此才算是簡單請求，且依然遵循同源政策（指瀏覽器中某些行為，必須與來源頁面的「協定」、「網域」、「連接埠」都相同，才能進行。)；後者通常是在發送會帶有副作用的 HTTP 請求方法前，規範瀏覽器要先發送預檢請求，預檢請求會以 HTTP OPTIONS 的方法送出，以向伺服器確認後續的請求能否傳送，如果預檢請求沒有通過，那麼後續真正要發送的實際請求（例如 POST、PUT、DELETE 等）就不會發送。

如果需要在發送 request 的時候帶上 cookie，那必須滿足三個條件：

1. 後端 Response header 有 Access-Control-Allow-Credentials: true。
2. 後端 Response header 的 Access-Control-Allow-Origin 不能是 *，要明確指定。
3. 前端 fetch 加上 credentials: 'include'。

如果沒有這個限制的話，那代表任何網站（任何 origin）都可以發 request 到這個 API，並且帶上使用者的 cookie，這樣就會有安全性的問題產生。

如何預防 CORS 漏洞？

1.正確配置跨域請求

如果 Web 資源包含敏感資訊，則應在 Access-Control-Allow-Origin 標頭中正確指定來源。

2.只允許信任的網站

使用通配符來表示允許的跨域請求的來源而不進行驗證很容易被利用，應該避免。

3.避免將 null 列入白名單

避免使用標題 Access-Control-Allow-Origin: null。應針對私有和公共伺服器的可信來源正確定義 CORS 頭。

4.避免在內部網路中使用通配符

當內部瀏覽器可以訪問不受信任的外部域時，僅靠信任網路配置來保護內部資源是不夠的。

5.CORS 不能替代伺服器端安全策略

CORS 定義了瀏覽器的行為，絕不能替代伺服器端對敏感數據的保護。攻擊者可以直接從任何可信來源偽造請求。因此，除了正確配置的 CORS 之外，Web 伺服器還應繼續對敏感數據應用保護，例如身份驗證和會話管理。