CORS

CORS 是跨來源資源共用的簡稱，他是一種使用額外 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。而跨來源指令是現今網路上許多頁面所載入的資源，如 CSS 樣式表、圖片影像、以及指令碼（script）都來自與所在位置分離的網域。但程式碼所發出的跨來源 HTTP 請求也會受到限制。主要是基於安全性考量中的同源政策而受到的限制。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。同源政策就是由 fetch API 或 XMLHttpRequest 等方式，讓我們透過 JavaScript 取得資源。常見的應用是向後端 API 拿取資料再呈現在前端，而存取資源時，如果是同源的情況下，存取不會受到限制。而什麼是同源呢，所謂的同源要滿足三個條件，相同的通訊協定（protocol），即 http/https、相同的網域（domain)及相同的通訊埠（port）。

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，必須符合下面兩個條件：只能是 HTTP GET, POST or HEAD 方法及自訂的 request header 只能是 Accept、Accept-Language、Content-Language 或 Content-Type。如果是簡單的跨來源請求，在後端 GET/POST/HEAD 方法本身加上 `Access-Control-Allow-Origin` header。而非「簡單」的跨來源請求，例如：HTTP PUT/DELETE 方法，或是 Content-Type: application/json 等，瀏覽器在發送請求之前會先發送一個 「preflight request（預檢請求)」，其作用在於先問伺服器：你是否允許這樣的請求，真的允許的話，我才會把請求完整地送過去。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。跨來源資源共用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）中使用 CORS 以降低跨來源 HTTP 請求的風險。