B0929030 周兆騰

CORS(跨源資源共享) 用於讓網頁的受限資源能夠被其他域名的頁面存取的一種機制。通過該機制，頁面能夠自由地使用不同源（英語：cross-origin）的圖片、樣式、指令碼、iframes 以及影片。一些跨域的請求（特別是 Ajax）常常會被同源策略（英語：Same-origin policy）所禁止的。跨源資源共享定義了一種方式，為的是瀏覽器和伺服器之間能互相確認是否足夠安全以至於能使用跨源請求（英語：cross-origin requests）。比起純粹的同源請求，這將更為自由和功能性的（functionality），但比純粹的跨源請求更為安全。跨來源資源共享是一份瀏覽器技術的規範，提供了 Web 服務從不同網域傳來沙盒指令碼的方法，以避開瀏覽器的同源策略。

基於安全性考量，程式碼所發出的跨來源 HTTP 請求會受到限制。例如，XMLHttpRequest 及 Fetch 都遵守同源政策（same-origin policy）。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。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（如 XMLHttpRequest 或 Fetch）中使用 CORS 以降低跨來源 HTTP 請求的風險。那有哪些請求會用到 CORS 呢? 跨來源資源共用標準可用來開啟以下跨站 HTTP 請求：(1) 使用 XMLHttpRequest 或 Fetch API 進行跨站請求，如前所述。(2) 網頁字體（跨網域 CSS 的 @font-face 的字體用途），所以伺服器可以佈署 TrueType 字體，並限制只讓信任的網站跨站載入。(3) WebGL 紋理 (en-US)。(4) 以 drawImage (en-US) 繪製到 Canvas 畫布上的圖形／影片之影格。(5) CSS 樣式表（讓 CSSOM (en-US) 存取）。(6) 指令碼（for unmuted exceptions）。

跨來源資源共用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。

最後來做個 CORS 跟 JSONP 的比較，跨來源資源共享（CORS）是 JSONP 模式的現代版。與 JSONP 不同，CORS 除了 GET 請求方法以外也支援其他的 HTTP 請求。用 CORS 可以讓網頁設計師用一般的 XMLHttpRequest，這種方式的錯誤處理比 JSONP 要來的好。另一方面，JSONP 可以在不支援 CORS 的老舊瀏覽器上運作。現代的瀏覽器都支援 CORS。