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 (en-US)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（domain）、通訊協定（protocol）或通訊埠（port）的資源時，會建立一個跨來源 HTTP 請求（cross-origin HTTP request）。

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（Cross-Origin Resource Sharing，簡稱 CORS）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。

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。

部分請求不會觸發 CORS 預檢。這類請求在本文中被稱作「簡單請求（simple requests）」，雖然 Fetch 規範（其定義了 CORS）中並不使用這個述語。一個

不觸發 CORS 預檢的請求——所謂的「簡單請求（simple requests）」——其滿足以下所有條件：

僅允許下列 HTTP 方法：

- GET
- HEAD (en-US)
- POST

不同於上面討論「簡單請求」的例子，「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。

XMLHttpRequest 或 Fetch 在 CORS 中最有趣的功能為傳送基於 HTTP cookies 和 HTTP 認證（Authentication）資訊的「身分驗證（credentialed）」請求。預設情況下，在跨站 XMLHttpRequest 或 Fetch 呼叫時，瀏覽器**不會**送出身分驗證。必須要於 XMLHttpRequest 物件中或是在呼叫 Request (en-US) 建構式時設置一個特定的旗標。

在回應一個身分驗證請求時，伺服器**必須**於 Access-Control-Allow-Origin 標頭值中指定一個來源，而不是使用「*」萬用字元（wildcard）。並且在 CORS 回應中設定的 cookies 受制於一般的第三方 cookie 政策。因此如果使用者將其瀏覽器設定為拒絕所有第三方 cookies，則 cookies 不會被保存。