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）**。
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。
「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。

CORS 通訊協定最初要求此預檢請求重新導向的行為，但在隨後的修訂中即改為不要求使用。然而，大多數的瀏覽器尚未實作此變動，且仍舊依照原本的行為要求。

因此直到瀏覽器趕上規範之前，你可以使用下列一或兩種方法來解決這個限制：

- 變更伺服器端的行為以避免預檢以及／或是避免重新導向——假如你對被請求的伺服擁有控制權
- 變更請求為簡單請求，讓預檢不會發生
- 建立一個簡單請求來測定（使用 Fetch API 的Response.url (en-US)或XHR.responseURL來測定預檢請求最終真正導向的 URL）。
- 建立另一個請求（「真正的」請求）傳送至第一步自Response.url (en-US)或XHR.responseURL所獲得的 URL。

然而，假如請求是由於存在Authorization標頭而觸發預檢，便無法利用以上的步驟來解除限制。並且直到你對被請求的伺服擁有控制權前，沒有其他方式能夠解決。

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）。

Access-Control-Allow-Credentials (en-US)標頭表示了當請求的credentials旗標為真時，是否要回應該請求。當用在預檢請求的回應中，那就是指示後續的實際請求可否附帶身分驗證。由於簡單的GET請求沒有預檢，所以如果一個簡單請求帶有身分驗證，同時假設此標頭沒有與資源一併回傳，則回應會被瀏覽器所忽略並且不會回傳予呼叫的網站內容。