

CORS(Cross-Origin Resource Sharing)的翻譯是跨來源資源共用。當我們在不同網域、通訊協定或通訊埠請求資料的時候就會遇到 CORS 的問題，舉例來說，使用 Google search 頁面的開發者模式下 fetch <https://www.google.com/>網域的資源時不會產生任何問題，但是如果是 <https://www.yahoo.com/>網域就會產生錯誤，這是基於安全性考量，所以除了使用 CORS 標頭 API 才能拿到跨域的資源。

而 CORS 請求又分為兩種，分別為簡單請求和非簡單請求，若是為簡單請求，被請求端會檢查該來源(Origin)是否是許可範圍，如果是的話就會返 Origin 的值或是*，因此若是被返回 Access-Control-Allow-Origin 的錯誤訊息，可以藉由修改響應頭 Access-Control-Allow-Origin 為*或 Origin 的值以取得資源；如果來源有特殊要求即非簡單請求，瀏覽器(要求端)就會發起預檢請求，被請求端確認了來源(Origin)和特殊請求(Access-Control-Request-Method 和 Access-Control-Request-Headers)後，瀏覽器才會正式發出請求並取得資源。

除了修改響應頭以達到取得資源的權限外，也可以利用 JSONP 和 Iframe 解決跨域資料問題。JSONP 是透過 HTML 中沒有跨域限制的標籤，像是 img、script 等，再藉由指定 callback 函式將回應內容傳回 JavaScript 中，但是只能處理 Get request；利用 Iframe 和 form 也可以解決跨域問題，因為 form 提交時不會產生跨域問題，因此可以利用 form 跨域，首先建立 iframe 標籤以用來接受資料，再將 iframe 建至 Dom tree 中，再從被請求端提交 form 至 iframe 裡就完成跨域資料傳輸。

若是連接外部 API 則可以利用代理伺服器的方式，由於跨域保護的限制是瀏覽器的規範，因此只要不透過瀏覽器發送請求就不會有限制，例如上課練習台北市即時雨量利用 CROS-ANYWHERE 就是一個簡單、免費的掛域代理伺服器。