簡單地說，*CORS 是一種安全機制，允許來自一個域或 Origin* 的網頁訪問具有不同域的資源（*跨域請求*）。CORS 是對現代瀏覽器中實施的同源策略的放寬。如果沒有 CORS 等功能，網站就只能通過所謂的同源策略訪問來自同源的資源。CORS (Cross-Origin Resource Sharing) 是針對不同源的請求而定的規範，透過 JavaScript 存取非同源資源時，server 必須明確告知瀏覽器允許何種請求，只有 server 允許的請求能夠被瀏覽器實際發送，否則會失敗。跨來源資源共用是一種使用額外 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）。CORS 分為簡單請求（Simple requests）和預檢請求（Preflighted requests）兩種，基本上能不能成功進行 CORS 都是看後端的造化，前端只能看著錯誤訊息請後端趕快修正。簡單請求

如果 Request method 是 GET、HEAD、POST 其一，且 Request header 的 Content-Type 是以下其中一種就是簡單請求，後端不需再做額外設定。

- application/x-www-form-urlencoded
- multipart/form-data
- text/plain

預檢請求

只要不符合簡單請求的規則例如使用了 PUT、DELETE 等 Method 或者 Content-Type 是 application/json，在送出該 Request 之前，瀏覽器會先進行一次預檢（Preflight），和簡單請求不同的是如果沒有通過預檢，就不會發送 Request。

假設您 在登錄 https://examplebank.com 時*瀏覽到惡意網站 https://evilunicorns.com*。如果沒有同源策略，即使黑客網站無法直接訪問銀行的 cookie ，黑客網站也可以對 *https://examplebank.com/api* 進行經過身份驗證的惡意 AJAX 調用。POST /withdraw

這是由於瀏覽器行為會自動附加綁定到 *https://examplebank.com* 的任何 cookie 以用於對該域的任何 HTTP 調用，包括從 *https://evilunicorns.com* 到 *https://examplebank.com* 的 AJAX 調用。通過將 HTTP 調用限制為僅對同源（即瀏覽器選項卡的域）的調用，同源策略關閉了一些黑客後門，例如跨站點請求偽造（CSRF）（儘管不是全部。仍然需要像 CSRF 令牌這樣的機制）。