http 跨網域連線的情境

發出 http 請求(存取資源)時，瀏覽器會在標頭 Origin 的欄位填上當前網頁的網域，如果當前網頁的網域和發出的請求目的地網域不同，那麼這個請求就是跨網域連線。

那甚麼情況會遇到跨網域連線呢? 當你在網頁端需要串接的資源所在的網域跟你的網頁網域不同的時候就會遇到了(例如: 串接第三方的 api、請求來自第三方網站的圖片)。

問題點

一般來說，存取第三方的資源不是很常見的嗎，但思考一個問題: 如果今天自己建立的伺服器服務，上面的資源(如: 圖片)或 API 被第三方毫無限制的存取，那這時候是不是一種問題? 更嚴重可能會關係到資料安全。

同源政策 (same-origin policy)

接續前面提到的問題，這時候就有了一個基於安全考量用來限制存取**跨來源資源**的概念，稱為同源政策，在這個概念下請求的 header 裡 Origin 的 protocol、host、port 都跟伺服器相同的情況看作同源；如果有任何一個條件不同，就看作是非同源，目前常見的瀏覽器幾乎都會根據同源政策實現對應的實作，對存取非同源資源的動作加上限制，提升安全性。

CORS

有了前面的概念，會發現要存取非同源的資源就會遇到同源政策的問題要面對，而 CORS 便是: 如果我需要存取不同來源的資源，那我該做哪些事情(或是該具備甚麼條件)才能順利存取非同源的資源。

伺服器端的部分，回覆請求的標頭需要含有 Access-Control-Allow-Origin 欄位，收到回覆(response)的瀏覽器端會去檢查這個欄位，如果不符合條件則會因為同源政策的關係而擋下 response，然後 console 跳紅字報錯。

瀏覽器端的部分，會幫你把請求區分成兩種分類，一種是**簡單請求**，另外一個是**非簡單請求**。

當瀏覽器判斷你發出的請求是簡單請求時，他會直接將請求發送出去；如果判斷為非簡單請求，會先把請求擋下，然後根據擋下的請求發送一個預檢**請求**(http method 為 option)，伺服器回覆之後瀏覽器會根據回覆的內容決定要不要將真正的請求發送出去。