JavaScript 透過 fetch API 或 XMLHttpRequest 等方式發起 request，必須遵守同源政策，同源政策（Same-origin policy）會讓網站不能隨意存取來自其他網站的資源，想要存取跨來源資源必須在某些特定情況下才被允許，為網站安全的基礎，來源(origin)包含通訊協定（protocol）、網域（domain）、通訊埠（port）三個部分，所以可以藉由這三者是否相同，來判斷是否為同源。在同源政策下，非同源的請求則會因為安全性的考量受到限制，會產生一個跨來源 http 請求（cross-origin http request），瀏覽器會強制你遵守 CORS（Cross-Origin Resource Sharing，跨域資源存取）的規範，否則瀏覽器會讓請求失敗。

CORS 是針對非同源的請求而定的規範，可以防止惡意的攻擊，能夠增加跨域資料傳輸的安全性，是一種使用額外 HTTP 標頭，令目前瀏覽網站取得存取其他來源（網域）伺服器特定資源權限的機制，透過 JavaScript 存取非同源資源時，伺服器必須明確告知瀏覽器允許何種請求，只有伺服器允許的請求能夠被瀏覽器實際發送，否則會失敗。在 CORS 的規範裡面，跨來源請求有分「簡單」和非「簡單」的兩種跨來源請求，非簡單的跨來源請求，在請求之前都會先發送預檢(preflight)請求，確定伺服器端有設定正確的相關 Http 標頭，當伺服器檢查通過後，才會實際發出請求。

當 server 端收到這個跨來源請求時，它可以依據「請求的來源」，決定是否要允許這個跨來源請求。如果伺服器允許這個跨來源請求，它可以「授權」給這個來源的 JavaScript 存取這個資源。簡單請求（simple requests）僅允許 HTTP GET、POST 或 HEAD，如果是自訂的請求標頭只能是符合「CORS 安全列表請求標頭」的標頭，而且沒有事件監聽器被註冊到任何用來發出請求的 XMLHttpRequestUpload 上，請求中也沒有 ReadableStream (en-US) 物件被用於上傳，不符合上述規定的都是非簡單請求。預檢請求會讓 HTTP 送出請求到另一個網域，確認後續實際（actual）請求是否可以安全送出，由於跨網域請求可能會攜帶使用者資料，所以要先進行預檢請求。

當收到預檢請求時，伺服器必須告訴瀏覽器允許的方法和標頭有哪些，伺服器回傳予取存控制請求之由跨來源資源共用規範所定義的 HTTP 回應標頭。Access-Control-Request-Method 標頭用在發出的預檢請求中，告訴伺服器後續實際（actual）請求所用的 HTTP 方法。Access-Control-Request-Headers 標頭用在發出的預檢請求中，告訴伺服器端後續實際（actual）請求所帶的 HTTP 標頭。

參考資料：

https://shubo.io/what-is-cors
https://developer.mozilla.org/zh-TW/docs/Web/HTTP/CORS