當我們在 JavaScript 中透過 fetch 或 XMLHttpRequest 存取資源時，需要遵守 CORS (Cross-Origin Resource Sharing，跨來源資源共用)。瀏覽器在發送請求之前會先發送 preflight request（預檢請求)，確認伺服器端設定正確的 Access-Control-Allow-Methods、Access-Control-Allow-Headers 及 Access-Control-Allow-Origin 等 header，才會實際發送請求。使用 cookie 的情況下還需額外設定 Access-Control-Allow-Credentials header。和 cors 相反的是同源政策，先理解瀏覽器的「同源政策」。大家應該都有用過瀏覽器提供的 fetch API 或 XMLHttpRequest ，讓我們透過 JavaScript 取得資源。常見的應用是向後端 API 拿取資料再呈現在前端。用 JavaScript 透過 fetch API 或 XMLHttpRequest 等方式發起 request，必須遵守同源政策 (same-origin policy)。簡單地說，同源政策就是用 JavaScript 存取資源時，如果是同源的情況下，存取不會受到限制；然而，在同源政策下，非同源的 request 則會因為安全性的考量受到限制。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 request 失敗。同源政策至少需要滿足三項包括相同的通訊協定 (protocol)，即 http/https 和相同的網域 (domain)以及相同的通訊埠 (port)。

而 cors 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。在 CORS 的規範裡面，跨來源請求有分兩種：「簡單」的請求和非「簡單」的請求。所謂的「簡單」請求，必須符合下面兩個條件：只能是 HTTP GET, POST or HEAD 方法自訂的 request header 只能是 Accept、Accept-Language、Content-Language 或 Content-Type（值只能是 application/x-www-form-urlencoded、multipart/form-data 或 text/plain）。細節可以看 fetch spec。不符合以上任一條件的請求就是非簡單請求。違反簡單請求的地方有三個，分別是：(1) 他是 http DELETE 方法；(2) 他的 Content-Type 是 application/json；(3) 他帶了不合規範的 X-CUSTOM-HEADER。Origin（來源）

首先，瀏覽器發送跨來源請求時，會帶一個 Origin header，表示這個請求的來源。

Origin 包含通訊協定、網域和通訊埠三個部分。

所以從 https://shubo.io 發出的往 https://othersite.com/data 的請求會像這樣：

```
GET /data/
Host: othersite.com
Origin: https://shubo.io
...
```

Access-Control-Allow-Origin

當 server 端收到這個跨來源請求時，它可以依據「請求的來源」，亦即 Origin 的值，決定是否要允許這個跨來源請求。如果 server 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。

授權的方法是在 response 裡加上 Access-Control-Allow-Origin header：

```
Access-Control-Allow-Origin: https://shubo.io
```

如果 server 允許任何來源的跨來源請求，那可以直接回 *：

```
Access-Control-Allow-Origin: *
```

當瀏覽器收到回應時，會檢查請求中的 Origin header 是否符合回應的 Access-Control-Allow-Origin header，相符的情況下瀏覽器就會讓這個請求成功，我們也可以順利地用 JavaScript 讀取到回應；反之，則瀏覽器會將這個 request 視為是不安全的而讓他失敗，即便 server 確實收到請求也成功地回應了，但基於安全性的理由 JavaScript 中沒有辦法讀到回應。

非「簡單」的跨來源請求，例如：HTTP PUT/DELETE 方法，或是 Content-Type: application/json 等，瀏覽器在發送請求之前會先發送一個「preflight request(預檢請求)」，其作用在於先問伺服器：你是否允許這樣的請求？真的允許的話，我才會把請求完整地送過去。一般的 http request 會帶有該網域底下的 cookie；然而，跨來源請求預設是不能帶 cookie 的。