B0929053 陳致佑

# CORS

CORS (Cross-Origin Resource Sharing),中文為跨來源資源共用。顧名思義，其功能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。而這裡必須先瞭解到什麼是同源資源，所謂的同源，必須滿足以下三個條件 : 1.相同的通訊協定 (protocol)，即 http/https 2.相同的網域 (domain) 3.相同的通訊埠 (port)。若我們的資源不是在同源的情況下時，就會產生一個跨來源 http 請求，像是:

```
try {
  fetch('https://othersite.com/data')
} catch (err) {
  console.error(err);
}
```

而跨來源請求必須遵守 CORS 的規範。當伺服器沒有正確設定時，請求就會因為違反 CORS 失敗，在 Chrome DevTool 就會看到以下的經典錯誤

```
Access to fetch at *** from origin *** has been blocked by
CORS policy: No 'Access-Control-Allow-Origin' header is
present on the requested resource. If an opaque response
serves your needs, set the request's mode to 'no-cors' to
fetch the resource with CORS disabled.
```

因此要如何正確的設定跨來源需求變相當的重要。以下將方法區分成兩種:簡單和一般跨來源請求。

簡單跨來源需求必須符合下面兩個條件:

1. 只能是 HTTP GET, POST or HEAD 方法
2. 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。

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」，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
```

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。最後，瀏覽器會收到 preflight request ，他是一個 http OPTIONS 方法，會帶有兩個 request header：Access-Control-Request-Method 和 Access-Control-Request-Headers。當它收到的 preflight request 是正確的時候，就代表 CORS 的驗證通過，就可以送出跨來源請求了。接下來，瀏覽器實際幫我們送出以下的跨來源請求：

```
POST /data/
Host: othersite.com
Origin: https://shubo.io
Content-Type: application/json
X-MY-CUSTOM-HEADER: 123
```

最後一步，server 還是要回應 Access-Control-Allow-Origin header。瀏覽器會再檢查一次跨來源請求的回應是否帶有正確的 Access-Control-Allow-Origin header，便大功告成！

參考網站: https://shubo.io/what-is-cors/