什麼是 CORS?

Cross-Origin Resource Sharing 簡稱 CORS，中文為跨來源資源共享。

web 瀏覽器有同源政策的限制，而 CORS 則是一種安全確認機制，針對不同源的請求而定的規範，讓瀏覽器和伺服器之間能確保安全的進行 cross origin 資源共享，即若伺服器同意，即可達成跨來源資源共享。

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。

1.簡單跨來源請求

簡單請求

. HTTP method 為: 「GET」 或 「HEAD」 或 「POST」

. 自訂的 request header 只能是 Accept、Accept-Language、Content-Language 或 Content-Type（只能是 application/x-www-form-urlencoded、multipart/form-data 或 text/plain）。

. 請求中的任意 XMLHttpRequestUpload 對象均沒有註冊任何事件監聽器；XMLHttpRequestUpload 對象可以使用 XMLHttpRequest.upload 屬性訪問

.請求中沒有使用 ReadableStream 對象

不符合以上任一條件的請求就是非簡單請求。

ＥＸ：

```
const response = await fetch('https://othersite.com/data', {
   method: 'DELETE', //違反 http method 規定
   headers: {
      'Content-Type': 'application/json', //違反 content-type 規定
      'X-CUSTOM-HEADER': '123' //帶了不合規範的 X-CUSTOM-HEADER
   }});
```

Origin、Access-Control-Allow-Origin

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
...
```

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: *
```

瀏覽器在收到 Server response 之後，會先檢查 Server response header 中的 Access-Control-Allow-Origin 裡面是否有包含發起 Request 的 Origin，有的話就會允許通過，讓 js 順利接收到 Response。

2.非簡單跨來源請求

.HTTP method 為: PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH 任一個
. Content-type 標頭值為除了這些以外的值：
「application/x-www-form-urlencoded」或「multipart/form-data」或「text/plain」。
. XMLHttpRequestUpload 對象註冊了任何事件監聽器

非簡單請求會先發送一個預檢請求（preflight），若 server 同意後，瀏覽器才會真正發出要資料的 request。

簡單來說，preflight 就是一個驗證機制，確保後端知道前端要送出的 request 是預期的，瀏覽器才會放行。
針對 request，瀏覽器會幫忙帶上兩個 header：

1.Access-Control-Request-Headers ->帶上不屬於簡單請求的 header
2.Access-Control-Request-Method ->帶上 HTTP Method
針對 preflight request，我們也必須給 Access-Control-Allow-Origin 這個 header 才能通過。
且當你的 CORS request 含有自訂的 header 的時候，preflight response 需要明確用 Access-Control-Allow-Headers 來表明：「我願意接受這個 header」，瀏覽器才會判斷預檢通過。

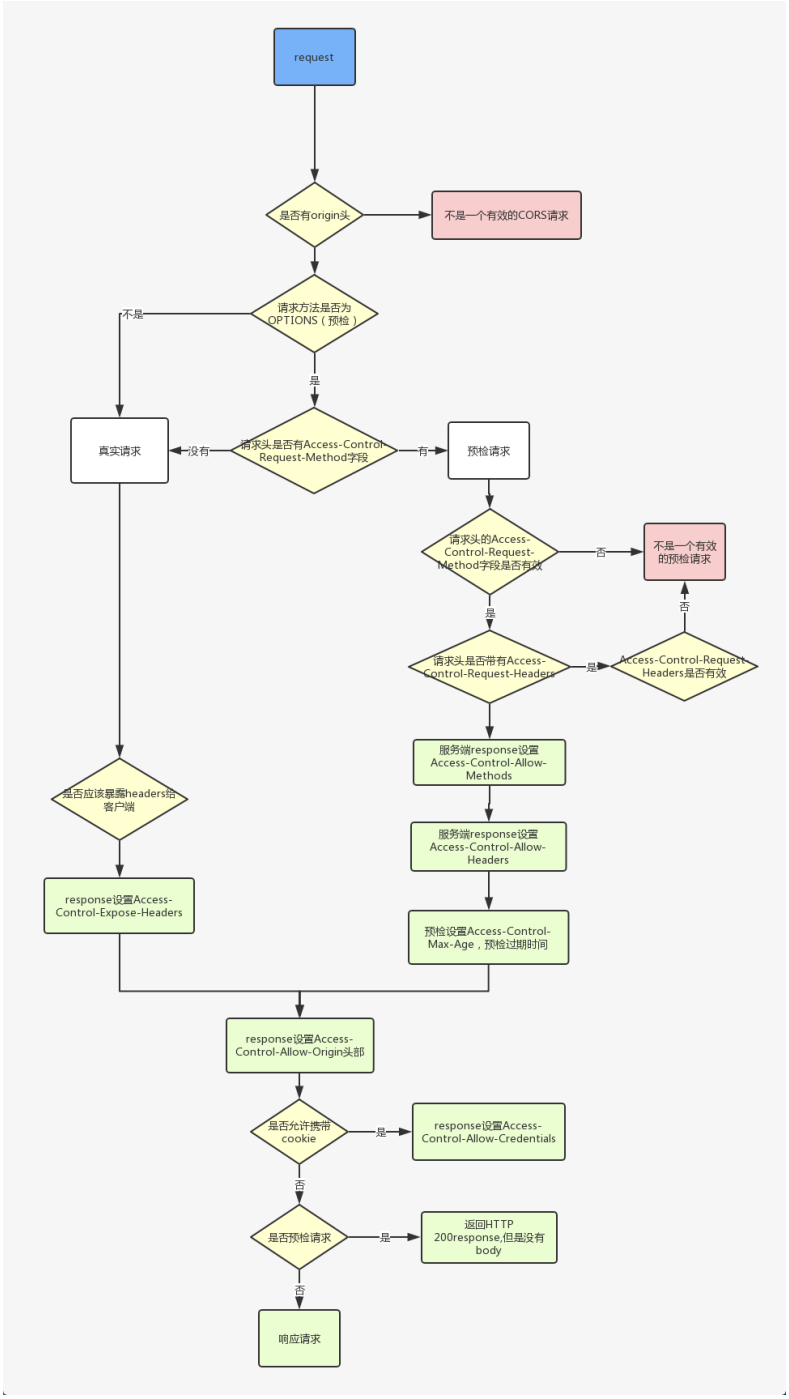
流程會像是這樣：

1. 我們要送出 POST 的 request 到 http://localhost:3000/form
2. 瀏覽器發現是非簡單請求，因此先發出一個 preflight request

3. 檢查 response，preflight 通過
4. 送出 POST 的 request 到 http://localhost:3000/form

所以如果 preflight 沒有過，第一個步驟的 request 是不會被送出的。

CORS 流程圖



https://github.com/amandakelake/blog/issues/62

https://vocus.cc/article/5f7abea0fd897800014ca129

https://shubo.io/what-is-cors/#%E5%90%8C%E6%BA%90%E6%94%BF%E7%AD%96-same-origin-policy