CORS 都不是前端會發生的問題，通常不會是前端工程師要處理的部分，所謂的 CORS 就是跨來源呼叫 API，如果 API 的來源相同就不需要跨來源呼叫(same origin 同源)，例如:https:\\casey.tw 和 https:\\google.com 就是不同源，因為他們的 origin 不一樣。Origin 可以當作是 scheme+host+port 的組合，scheme=http or https，host 就是 casey.tw，port 沒有特定指定通常 https 都是 443，同不同源的判斷方式就是看 scheme 和 host 的內容是不是相同的，此外 domain 和 subdomain 之間也是不同源，例如:api.casey.tw 和 casey.tw 是不同源的關係，基本上網址要長的一模一樣，只可以 path 之後的內容有所不同，那為甚麼要阻擋跨來源呼叫 API 呢?這一切都是為了安全性的問題，減少資料外洩的可能性。當我們想要對一個跨來源資料做 AJAX 時，然後被擋住了，這都是瀏覽器搞的鬼，如果沒有瀏覽器的話，根本不會有 same-origin policy 的問題，所以這就是為什麼 proxy 會存在的原因，因為 proxy 不是透過瀏覽器去抓去 API，而是透過後端自己去拿資料，此外剛剛提到瀏覽器會擋住跨來源 API，說明白一點，瀏覽器擋掉的不是你送出去的 request，而是把送回來的 response 給擋住了，所以其實你的送出去 request 的確有準確的到 serve 端，且 serve 也把 response 給回傳給你。那 CORS 會發生在那些請求呢?當我們使用 XMLHttpRequest 或是 Fetch API 時就會發生上述情況，那要如何解決呢，除了上述使用 proxy，也就是使用別人寫好的或自己寫的後端去抓取跨來源資料，而不是透過瀏覽器，雖然可以關掉瀏覽器安全性設置，這樣就不會有 CORS 的問題，但這個方法治標不治本，還有有些人會把 fetch mode 改成 no-cors，雖然你可以拿到 response，但你會發現 response 的 state 是 0，且 type 會是 opaque，這個方法就類似跟瀏覽器說，我就是要發 request 到一個沒有 CORS header 的 url ，請不要給我出錯，所以你其實就知道本來這樣就會出錯，就算你把 header (Access-Control-Allow-Origin)加上去也拿不到，設置這個 mode 並不會讓你突破限制，相反的這個模式是在跟瀏覽器說，我就是要發 request 給一個沒有 cors header 的資源，我知道我拿不到 response，所以你絕對不要給我 response，那到底要怎麼解決 CORS，很簡單只要加上 header，剛剛提到後端才是有權限決定到底要不要給你的一方，也就是說加上 header，可以告訴瀏覽器：「我(後端)允許這個 origin 跨來源存取我的資源」，例如: response.set('Access-Control-Allow-Origin',' https:\\casey.tw')允許 https:\\casey.tw 跨來源請求，所以當前端用 AJAX 去送 request 的時候，就可以拿到 response，不會出現任何錯誤，此外這個 header 是要加在後端的，因為這個 header 是在 response 裡面的，所以如果你跟想存取的資源有合作關係的話，通常直接請他們設定這個 header 就行了。例如說你在串接公司後端的 API 時碰到 CORS 問題，這時候請去找後端工程師幫你把這個 header 加上去，就可以拿到 response 了