跨來源資源共用（CORS）心得

跨來源資源共用（CORS）可以繞過同源策略(當網域名稱不同時資料存取會被限制)，這種情況特別會發生前端上，slack 上的文章後面也分別介紹簡單和預檢請求的差別，當初課堂上 ajax 存取不到資料時我有點疑惑，因此回去查相關資料，有的寫說是因為 ajax 只是個“字元型”的請求當下載的檔案是以是二進位制形式儲存因此檔案只能讀取而無法執行，但格錯誤其實跟 CORS 沒什麼關聯，後來我又去查 ajax 與跨域問題，會有這個的原因其實是因為安全疑慮，例如當你的網站僅限公司內部員工存取，而外部人員想使用 ajax 存取資料時會產生資安問題，換句話說 CORS 就是為此產生，有 CORS 擋住的網站我們無法存取該網站資料，CORS 限制的其實是「拿不到 response」，而不是「發不出 request」，request 其實已經發出去了，只是我們沒有權限讀取，而簡單和預檢的主要差別在是否會檢查使用者跨站請求資料，簡單預檢不會檢查使用者可以直接存取，預檢則相反。

如果我們是開發者可以透過兩種方法來讓使用者存取資料，如果是簡單請求則在標頭檔加入 res.header(“Access-Control-Allow-Orrigin”,”*”)，*代表任意網域，預檢請求則加入 Access-Control-Allow-Methods 及 Access-Control-Allow-Headers header。另一種則是寫 callback function。

我對於 CORS 的看法保持正面的態度，近年來資安議題越來越熱門，CORS 可以抵擋駭客攻擊(前提是設定正確)，但 CORS 並非萬能的，伺服器上的安全策略與身分驗證同時也不可或缺，其他像是設定時盡量不使用萬用字元*，不要使用 null 作為白名單使用 Access-Control-Allow-Origin: null，這些都十分重要，其中有個案例是可以使用 CORS 攻擊比特幣 API，原因在於 CORS 設定常被開發人員所忽略，造成開發人員的網站需要讓多網域共享資源時，須動態產生白名單網域來源清單，因開發人員動態產生白名單時信任，攻擊者可控的 HTTP Request Origin 的參數，造成攻擊者可取得伺服器上的機敏資料，顯示出後端工程師的重要，CORS 本身概念是所有工程師必須要會的，前端需與後端工程師需相互配合像是前端回報問題給後端加入 CORS header，如果使用像是 proxy server 會有安全性的疑慮，多一層網站可以提取我們的資料，目前我找到解決的方法有使用測試工具 Burp suite 與 CORSCANNER，我們做為一個工程師須確保網頁的安全性才不容易被駭客入侵。

資料來源

https://blog.huli.tw/2021/02/19/cors-guide-1/

https://blog.huli.tw/2017/08/27/ajax-and-cors/

https://developer.mozilla.org/zh-TW/docs/Web/HTTP/CORS#%E5%AD%98%E5%8F%96%E6%8E%A7%E5%88%B6%E6%83%85%E5%A2%83%E7%AF%84%E4%BE%8B

https://www.youtube.com/watch?v=FF6zra7b7gM&ab_channel=FreeCoder

https://www.informationsecurity.com.tw/article/article_detail.aspx?aid=8703