

Cookie

B0929031 張宜茗

根據我們所理解的 Cors，在 JavaScript 中透過 fetch 或 XMLHttpRequest 存取資源時，瀏覽器在發送請求之前會先發送預檢請求，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，而使用 cookie 的情況下還需額外設定 Access-Control-Allow-Credentials header，從這個概念出發，我們可以知道 cors 和 cookie 有一定程度的相關。然而什麼是 Cookie 的功能是什麼呢，以一句話來概括的話就是將用戶信息存儲在網頁中，是存儲在計算機上的小型文本文件中的數據。

以實際應用來說，當 Web 服務器將網頁發送到瀏覽器時，連接會關閉，服務器會忘記有關用戶的所有信息，可想而知，發明 Cookie 是為了「記住用戶信息」的問題，也就是說當用戶訪問網頁時，姓名可以存儲在 cookie 中，下次用戶訪問該頁面時，cookie 會「記住」用戶的名字。

Cookie 在 javascript 中可以使用屬性創建、讀去和刪除，創建的作法相當直觀，用 document.cookie = "username=名稱";即可，雖然在默認的狀況下，關閉瀏覽器時就會刪除，但仍可藉由路徑使 cookie

屬於當前頁面，同樣地，cookie 也能夠修改，就像程式一樣，舊的 cookie 會被新的所覆蓋、取代；讀取 cookie 只需要 `let 參數 = document.cookie`；如此一來，`document.cookie` 將回傳所有 cookie，就像：`cookie1=value; cookie2=值; cookie3=值`；想要刪除 cookie 比較特別的是不必指定 cookie 值，而是將 `expires` 參數設置為過去的日期，但仍要定義 cookie 路徑才能夠確保刪除正確的 cookie，如果沒指定路徑，則某些瀏覽器就不允許刪除 cookie。

提到 cookie 我們就不能夠漏掉「session」，「session」粗略來說是 server 版的 cookie。雖然 cookie 能夠解決，開發網頁時 HTTP 協定為無狀態，也就是說伺服器並不會記錄使用者先前操作，這樣的難題，但仍衍伸出了一個矛盾的現象，如果程式碼：「`username=名稱;isLogin=true;`」，則名稱填入任何人，責任任何人都能夠登入網站了，因此需要搭配使用 session。讓使用者登入後，後端驗證完，立刻建立 session 以紀錄該用戶已登入，接著回傳寫著此 session ID 的 cookie，從此之後進入網站時，後端收到帶著 session ID 的 cookie，然後查找這個 session 的資訊，讓使用者不用輸入帳密即能登入網站；當然如果覺得這個方法過於麻煩，也可以使用不帶 session ID 的方法，一樣是登入後，經由後端認證，但此時必須設置一個寫著加密字串的 cookie，這樣的作法是為了之後登入時，讓

後端解密這個字串以辨別用戶身分與維持登入狀態，但不管是上述的哪種作法，原理都是把用戶 cookie 帶上的「東西」拿來查詢驗證。