Cross-Origin Resource Sharing(CORS)

Cross-Origin Resource Sharing(CORS)的中文名稱是 ： 跨來源資源共用。顧名思義，當我們從 **JavaScript** 當中透過 **fetch** 或 **XMLHttpRequest** 來請求存取外部資源時，就屬於一種存取跨來源資源的行為。當這樣的行為發生時，便需要受限於 **CORS**，亦即必須遵守同源政策。在實際的運作過程中，瀏覽器在發送請求前會先發送預檢請求，確認伺服器端設定正確的 **Access-Contorl-Allow-Methods**、**Access-Contorl-Allow-Headers** 及 **Access-Contorl-Allow-Origin** 等表頭時，才會發送請求。另外，在有使用 **cookie** 的情況之下，還需額外設定 **Access-Contorl-Allow-Credentials header**。

如果要確切了解甚麼是跨源就必須先知道同源的界定是甚麼。同源必須滿足以下三個來源都相同 **：1.**相同的通訊協定**(HTTP** 或 **HTTPs) 2.**相同的網域**(**網域例子 **: XXX.com) 3.**相同的通訊埠**(port** 號**)**。

那甚麼到底甚麼是跨來源資源共用**(CORS)**呢**?** 簡單地說，**CORS (Cross-Origin Resource Sharing)** 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。在 **CORS** 的規範當中，有分為簡單即非簡單的兩種不同請求。所謂的「簡單」請求，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**。只要不符合上面其中任一項，就是非簡單請求。

當一個支援跨來源資源共用**(CORS)**的瀏覽器在網頁送出一個 **REQUEST** 時，主要流程大致如下 ： 第一步、瀏覽器根據送出 **request** 的 **HTTP verb** 與 **header**，判斷這個 **request** 是一個簡單請求**(simple request)**或是非簡單請求**(**判斷的細節可參考 **MDN - HTTP access control (CORS) - Simple requests)**。如果是一個簡單請求，則直接送出 **request**。第二步、如果是一個非簡單請求的 **request**，則進行 **CORS preflight**。先對伺服器送出一個 **verb** 為 **OPTION** 的 **preflight request**，它會帶有特定的 **header** 告訴伺服器接下來的 **request** 需要哪些跨網域連線的權限。第三步、當伺服器收到 **preflight** 後，就會回傳帶有特定 **header** 的 **response** 給瀏覽器，告訴它有哪些權限是允許的。第四步、瀏覽器取得伺服器的 **response** 後，如果符合連線權限，就會送出真正的 **request**。如果發現權限不符，就會出現錯誤訊息而中斷送出 **request** 的步驟。

與跨來源資源共用相反的便是同源政策 **(Same Origin Policy)** 。同源政策是網站安全的基礎。以下為同源政策的一個簡單例子:假設今天有一個網站，其網址為:**https://a.com**。那在同源政策的規範下，這個網站只能存取自己網站裡面的資源**(**這裡指的資源包含:圖片、影片、程式碼等等**)**。不允許它去存去別的網站**(**例如 ：**https://b.com)**的資源，也不允許別的網站來存取它的資源。