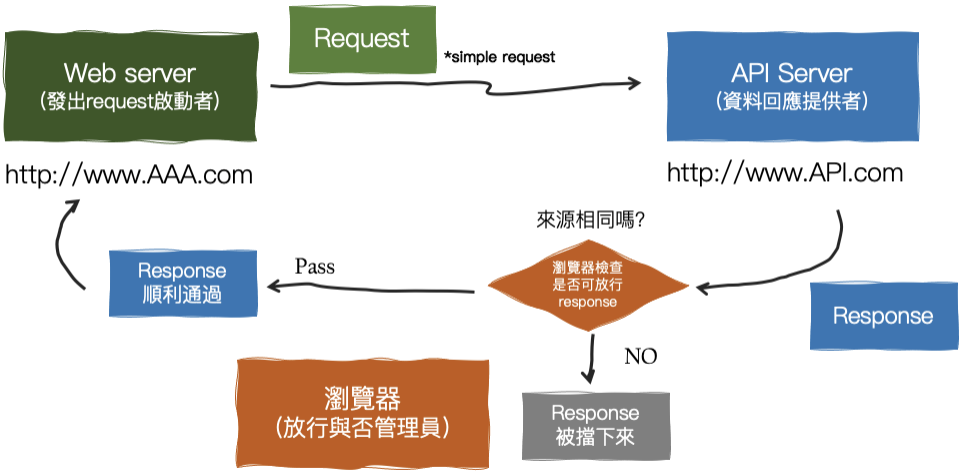
跨來源資源共用（CORS）

首先，我們要知道 CORS 的全名為 Cross-Origin Resource Sharing，是我們在使用 JS 存取資源時需要遵守的，其提供了網頁伺服器跨網域的存取控制，以增加資料傳輸的安全性：瀏覽器在發送請求之前會先發送預測請求，才會發送實際的請求。

講了這麼多，我們還是很難理解 CORS 究竟是什麼東西。在查看各篇文章時看到一個很淺顯易懂的例子，即，將 API 想像成大樓裡的一個個部門辦公室，每個部門皆有各自的門禁卡，且各個部門的人是不能隨意進出其他部門的，當他們必需進入其他部門時，便會需要一個管理員去開啟門禁卡的權限，而 CORS 就是那位管理員！



導入 CORS 的方法有很多種，在網路上最常看到的為傳送 Origin: <domain> HTTP 標頭的請求，伺服器回應時會傳送 Access-Control-Allow-Origin: <domain>，前者的<domain>為提供該網頁的網域，後者的則是特定網域清單用以允許所有網頁的萬用字元。

當然，也不是每次的請求皆會成功，以網路上別人的實作結果為例：「access to XMLHttpRequest at 連結 A from origin 連結 B has been blocked by CORS policy: response to preflight request doesn't pass access control check: No 'Access-control-allow-origin' header is present on the requested resource」這段錯誤訊息代表當你要從連結 B 要 request 到連結 A 時，被 CORS 的擋下了，冒號後面則會告訴你原因——preflight request 未通過存取檢測，需要檢查看看 request 的來源的 header 是否包含 access-control-allow-origin。

在使用 CORS 時，可能會遇到許多其他不同種類的錯誤，以下幾點是我們可能會先簡單常識，但未必是正確的解決方法：

第一，關閉瀏覽器的安全性設置。有許多請求被擋住的原因其實是瀏覽器的限制，因此，只要瀏覽器沒有這個限制，我們就能平平安安快快樂樂拿到 response。但是，這是個治標不治本的方法，為什麼會這樣說呢？因為這只有在你的電腦上沒有問題，其他人的仍會出現錯誤。

第二，將 fetch mode 設成 no-cors。當我們看到錯誤訊息「set the request's mode to 'no-cors'」時，可能會照著訊息上的指示去設定，設定後我們也會發現錯誤訊息消失了，但是！我們並沒有拿到 response，因為 mode: no-cors 的真正意思是在和瀏覽器說：「我就是要發 request 到一個沒有 CORS header 的 url，所以請不要給我錯誤」如此一來，我們是絕對拿不到 response 的。

第三，不要用 ajax 拿資料。既然使用 ajax 會被阻擋跨來源的請求，那如果可以不用 ajax 拿資料不就沒有問題了嗎！做法即用 script 引入他人的 script，直接用變數存取資料的 data，再將 data 印出，我們就會發現資料已經到手了。這個方法是利用 script 標籤不會擋跨來源請求的特性，讓 server 動態產生檔案的內容，並且利用呼叫 JS 函式的方式傳遞 JSON 格式的資料