B0929045 叶子晴 資工二甲

HTTP Cookie（Web Cookie、Browser Cookie），HTTP 協定是無狀態的，伺服器並不會知道你之前做了什麼，Cookie 則是來解決這個問題，為伺服器傳送予使用者瀏覽器的一個小片段資料，而瀏覽器可能儲存之後於下一次請求回傳 Cookie至相同的伺服器，當兩次請求都來自相同的瀏覽器，Cookie通常被用來保持使用者的登入狀態。

Cookies主要用於三個目的，就是Session管理，用於帳號登入、購物車、遊戲分數，或任何其他伺服器應該記住的資訊；個人化，用於使用者設定、佈景主題，以及其他設定；追蹤，用於記錄並分析使用者行為。Cookies曾被當作一般的客戶端儲存方式來使用。在以前Cookie 是將資料儲存在客戶端的唯一合法方法，現在則建議使用現代的 Storage APIs。Cookies 會被每一個請求發送出去，所以可能會影響效能，尤其是行動裝置的資料連線。

Cookie的特性是可以紀錄使用者訊息、儲存在客戶端、連線時會自動帶上，過多的 Cookie 會浪費流量、或帶上無用之 Cookie、大小限制 4kb 左右、能夠設置過期時間，超過後 Cookie 將不再發送、專屬於某網域(路徑)，例如 google.com 的頁面不能存取 facebook.com 的 Cookie。

Session Cookie是指當客戶端關閉時就被刪除，因為它並沒有註明過期 Expires 或可維持的最大時間 Max-Age，網頁瀏覽器可使用 **Session Restoring**，讓 Session Cookies 永久保存，就像瀏覽器從來沒關閉。

Session Cookie其中第一種是叫常駐Cookies，是指不會在客戶關閉後到期，會是在一個特定的日期 (Expires) 或一個標明的時間長度後（Max-Age）到期。

第二種是叫Secure cookie，只有在加密的請求透過 HTTPS 協議時傳送給伺服器，則Cookie 只允許在 HTTPS 下傳輸，即便是 Secure，敏感的資訊絕對不會存在 Cookies 內，因為他們本質上還是不安全的，不能提供真正的保護。而為了避免跨站腳本攻擊，JavaScript 的[Document. Cookie (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/Document/cookie) API無法取得 HttpOnly Cookies，他們只會傳送到伺服器。舉例來說，不需要讓 JavaScript取用仍在伺服器 Sessions 中的 Cookies 時，就應該立 HttpOnly 的旗幟，則新的 Cookies 可經由 JavaScript 的 [Document.cookie (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/Document/cookie) 屬性生成，Cookie 只能被 server 端存取，無法在用戶端存取，沒辦法再透過 JavaScript 去操作它；若沒有立 HttpOnly 旗幟，已存在的 Cookies 可以透過 JavaScript 取得。

第三種是叫SameSite Cookie，讓伺服器要求 Cookie 不應以跨站請求的方式寄送，某種程度上避免了跨站請求偽造的攻擊（[CSRF](https://developer.mozilla.org/zh-TW/docs/Glossary/CSRF)），不過目前仍在實驗階段，尚未被所有的瀏覽器支援。

Domain 及 Path 的指示定義了 Cookie 的作用範圍，則Cookies應該被送到哪些URLs。舉例來說，當設定 Domain=mozilla.org 後，在像 developer.mozilla.org 之類的 Subdomains 中，Cookies 都被包含在內。而Path則指出一個必定存在於請求 URL 中的 URL 路徑，使 Cookie 標頭能被傳出。「/」字元是資料夾分隔符號，子資料夾也同樣會被匹配，當設定 Path=/docs 後，以下的路徑皆會匹配：/docs、/docs/Web/、/docs/Web/HTTP。