

HTTP cookie (web cookie、browser cookie) 為伺服器傳送予使用者瀏覽器的一个小片段資料。瀏覽器可能儲存並於下一次請求回傳 cookie 至相同的伺服器。Cookie 通常被用來保持使用者的登入狀態——如果兩次請求都來自相同的瀏覽器。舉例來說，它記住了無狀態 (stateless) (en-US)HTTP 協議的有狀態資訊。

Cookies 主要用於三個目的，Session 管理: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:使用者設定、佈景主題，以及其他設定、追蹤: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。這在當時 cookie 仍是將資料儲存在客戶端的唯一方法時是合法的，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)。現代客戶端的 storage APIs 為 Web storage API (en-US) (localStorage 和 sessionStorage) 以及 IndexedDB。

建立 cookies，收到一個 HTTP 請求時，伺服器可以傳送一個 Set-Cookie (en-US) 的標頭和回應。Cookie 通常存於瀏覽器中，並隨著請求被放在 Cookie HTTP 標頭內，傳給同個伺服器。可以註明 Cookie 的有效或終止時間，超過後 Cookie 將不再發送。此外，也可以限制 Cookie 不傳送到特定的網域或路徑。HTTP 回應標頭從伺服器傳送 cookies 至用戶代理。一個簡單的 cookie 可以如下例設定：Set-Cookie: <cookie-name>=<cookie-value> 這個來自伺服器的標頭告訴客戶端要儲存一個 cookie。

session cookie：當客戶端關閉時即被刪除，因為它並沒有註明過期(Expires)或可維持的最大時間(Max-Age)。不過網頁瀏覽器可使用 session restoring，讓 session cookies 永久保存，就像瀏覽器從來沒關閉。

常駐 cookies 不會在客戶關閉後到期，而是在一個特定的日期 (Expires) 或一個標明的時間長度後 (Max-Age)。

Secure cookie 只有在以加密的請求透過 HTTPS 協議時，傳送給伺服器。但即使是 Secure，敏感的資訊絕對不該存在 cookies 內，因為他們本質上是不安全的，這個旗標不能提供真正的保護。自 Chrome 52 以及 Firefox 52 開始，不安全的網站 (http:) 就不能以 Secure 的指示設定 cookies。為了避免跨站腳本攻擊 (XSS (en-US))，JavaScript 的 Document.cookie (en-US) API 無法取得 HttpOnly cookies；他們只傳送到伺服器。舉例來說，不需要讓 JavaScript 可以取用仍在伺服器 sessions 中的 cookies 時，就應該立 HttpOnly 的旗幟。

Domain 及 Path 的指示定義了 cookie 的作用範圍：cookies 應該被送到哪些 URLs。Domain 註明了受允許的 hosts 能接收 cookie。若無註明，則預設給當前文件位置的 host (en-US)，不包含 subdomain。若有被註明，則 subdomains 總是被包含。Path 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。%2F (「/」) 字元是資料夾分隔符號，子資料夾也同樣會被匹配。新的 cookies 亦可經由 JavaScript 的 Document.cookie (en-US) 屬性生成，且若沒有立 HttpOnly 旗幟，已存在的 cookies 可以透過 JavaScript 取得。

**Cookies** 會帶有他們所屬的網域名。若此網域和你所在的頁面網域相同，**cookies** 即為第一方 **cookie**，不同則為第三方 **cookie**。第一方 **cookies** 只被送到設定他們的伺服器，但一個網頁可能含有存在其他網域伺服器的圖片或組件，像橫幅廣告。透過這些第三方組件傳送的 **cookies** 便是第三方 **cookies**，經常被用於廣告和網頁上的追蹤。大部分的瀏覽器預設允許第三方 **cookies**，但也有些可以阻擋。