

HTTP 是一個「無狀態協議 Stateless Protocol」，每次從客戶端（Client）對伺服器（Server）發出的請求都是獨立的，這一次的請求無法得知上一次請求的內容與資訊。而無狀態協議的優點在於，由於不必保存狀態，自然可減少服務器的CPU 及內存資源的消耗。從另一方面來說，也正是因為HTTP 協議本身非常簡單，所以才會被應用在各種場景里。

Cookie 是伺服器（Server）傳送給瀏覽器（Client）的一小片段資料，並請瀏覽器保存起來，以便往後向相同的伺服器發送請求時，附上這 Cookie 的資料。而Cookie 的常見用途包含「儲存和追蹤使用者行為」、「儲存用戶登入、購物車等伺服器所需的資訊」和「儲存使用者設定和偏好」等。

一般Cookie所具有的屬性，包括：Domain、Path、Expire time/Max-age、secure和httponly。Domain表示當前Cookie所屬於哪個域或子域下，對於服務器返回的Set-Cookie中，如果沒有指定Domain的值，那麼其Domain的值是默認為當前所提交的http的請求所對應的主功能變數名稱的。比如訪問 <http://www.example.com>，返回一個Cookie，沒有指名domain值，那麼其為值為默認的www.example.com。Path表示Cookie的所屬路徑，可用於限制指定Cookie 發送範圍的文件目錄。Expire time/Max-age表示Cookie的有效期，expire的值，是一個時間，過了這個時間，該Cookie就失效了。或者是用max-age指定當前Cookie是在多長時間之後而失效。如果服務器返回的一個Cookie，沒有指定其expire time，那麼表明此Cookie有效期只是當前的Session，即是Session Cookie，當前Session會話結束後，就過期了。對應的，當關閉（瀏覽器中）該頁面的時候，此Cookie就應該被瀏覽器所刪除了。secure表示該Cookie只能用https傳輸，一般用於包含認證信息的Cookie，要求傳輸此Cookie的時候，必須用https傳輸。httponly，表示此Cookie必須用於http或https傳輸，意思是瀏覽器腳本（例如javascript）不允許訪問操作此Cookie的。

Cookie會被附加在每個HTTP請求中，所以無形中增加了流量。且在HTTP請求中的Cookie是明文傳遞的，所以安全性成問題。Cookie的大小限制在4KB左右。對於復雜的存儲需求來說是不夠用的。了解到存放較敏感的資訊在客戶端是有安全上的疑慮，因此我們改使用 Session 將使用者相關的敏感資訊存放在伺服器端 — 可能在記憶體或資料庫中，並創建一個相對應且獨特的 ID(Session ID)，在回傳給客戶端的 Cookie 中一併附上，未來客戶端只要附上含有這個 Session ID 的 Cookie 級伺服器，伺服器就能匹配相對應的 Session，也能找到需要的敏感資料了！

参考:

[Document.Cookie - Web APIs | MDN \(mozilla.org\)](#)

[HTTP Cookies - HTTP | MDN \(mozilla.org\)](#)

[Http协议中Cookie详细介绍 - 李小菜丶 - 博客园 \(cnblogs.com\)](#)

[\[筆記\] HTTP Cookies 和 Session 使用. HTTP 是一個「無狀態協議 Stateless...」 | by Mike Huang | 麥克的半路出家筆記 | Medium](#)

[认识HTTP----Cookie和Session篇 - 知乎 \(zhihu.com\)](#)