



Beneficios y Desventajas de Cookies y Webstorage de Almacenar Información del lado del Cliente

Autores: Antonio Aluralde Sánchez antonio@publiaciertos.net
David Ahmad Zonneveld Michel davidzonneveld@gmail.com

Universidad: Universidad Mayor de San Simón

Carrera: Ingeniería Informática

Materia: Programación Web

1. Introducción

Una de las principales características de todo sitio web moderno es su dinamismo. Esto implica muchas cosas. Una de ellas es el saber responder de manera personalizada a los que la usan. Otra es proveer una experiencia de navegación continua: Que si el usuario salta de una página a otra del mismo fabricante se recuerde lo que el usuario haya hecho en la primera página. Algunos sitios web usan para esto sesiones: El usuario tiene algún tipo de login que lo identifica con la página. Sin embargo, imagine que tuviera que hacer login a toda página que visite! Obviamente esta no es una situación ideal, se tiene que pensar en una alternativa.

La alternativa está en el uso de “tokens de sesión”. Usar a la máquina **del cliente** para almacenar información de quién es y enviársela al sitio web que necesite dicha identificación. Este mecanismo simula de una manera razonable el uso de sesiones. Es por esto que se crearon las cookies, para proveer estos tokens de sesión.

Muchos desarrolladores web, sin embargo, fueron descubriendo que las cookies pueden ser usadas para muchas cosas más que para almacenar tokens de sesión. Se podían usar para:

- Almacenar un historial de páginas visitadas
- Guardar datos de perfil.
- Almacenamiento de datos varios en la máquina del cliente

Sin embargo, siendo que este no fue el uso original pensado de las cookies, trajeron consigo muchos problemas de seguridad y eficiencia y no llegaron a satisfacer todas las necesidades que estos desarrolladores tenían. Es por esto que para HTML5 se decidió dar a cada uno de estos malos usos que se les dieron a las cookies su propia API. Fue así que se crearon las APIs de WebStorage, IndexedDatabase y Application Cache. En este artículo discutiremos la bifurcación cookies/WebStorage. Discutiremos las ventajas y desventajas que ofrece cada uno de ellos en el lo que es el almacenamiento de información en el lado del cliente (razón por la que se creó el WebStorage en primer lugar). Dejaremos la bifurcación de Cookies/ApplicationCache y Cookies/IndexedDatabase para artículos posteriores.

2. Cookies

a. Propiedades

Los usuarios pueden aceptar, borrar o bloquear las cookies según deseen. El tiempo de vigencia de cada cookie es un atributo configurable; pudiendo ser estas de sesión o persistentes durante un tiempo definido. Pueden asignarse a un dominio y ruta específicos, o hacerlas tan generales como se quiera. Pueden ser encriptadas con el HTTP y ser enviadas a través de conexiones seguras. Pueden restringir su acceso a solicitudes HTTP/HTTPS solamente.

b. Inconvenientes

Las cookies no identifican a una persona, sino a una combinación de ordenador, navegador, cuenta de usuario. Por lo tanto una persona podría tener varios conjuntos de cookies. De la misma forma las cookies podrían no diferenciar a varias personas que usen el mismo ordenador, navegador.

Las cookies también son vulnerables a ser hurtadas por distintos medios que explotan versiones antiguas de software y configuraciones inseguras. Dependiendo del contenido de la cookie podrían obtenerse, sesiones o información sensible.

Un atacante puede modificar el contenido de la cookie antes de enviarla al servidor.

No tiene control de concurrencia. Pueden surgir inconsistencias si se trabaja con múltiples pestañas del mismo servidor web que usen las mismas cookies.

3. Webstorage

Existen muchos casos en los que el usuario quisiera guardar grandes cantidades de datos en su máquina sin que se envíen en cada consulta al servidor. Un ejemplo obvio es la bandeja de entradas de e-mails. Ese es el propósito del WebStorage, proveer dicho mecanismo. WebStorage es la nueva API provista por HTML5 que nos permite almacenar grandes cantidades de datos en el lado del cliente.

a. Motivación

Las Cookies están entre las cosas más abusadas de la web. Entre los malos usos que se les da está el usarlas para almacenar grandes cantidades de datos en ellas. ¡Nunca fueron diseñadas para ello! A falta de alternativas, sin embargo, algunos desarrolladores web las usaron precisamente para eso, para almacenar datos en el lado del cliente. Esto trae consigo problemas en la seguridad y eficiencia (por razones descritas en la anterior sección). Además, trae a los desarrolladores muchas limitaciones, pues las cookies tienen tamaño muy reducido (4KB) y están asociadas con páginas específicas. ¿Qué pasa si ya no quiero asociar una página con un archivo físico en mi máquina (Véase, por ejemplo, el patrón de diseño MVC)? Estos problemas y limitaciones llevaron a la W3C (el gran cuerpo de especificaciones web) a considerar una API dedicada para el almacenamiento de grandes cantidades de datos en la máquina del cliente. Este API es el WebStorage.

b. Especificaciones

La W3C halló conveniente dividir el almacenamiento en la máquina del cliente en dos grandes categorías, según si se necesita que la información del cliente persiste a través de

sesiones o no. Es por esto que dividió la especificación del WebStorage en dos especificaciones separadas: SessionStorage y LocalStorage.

i. Session Storage

Session Storage es el mecanismo elegido para almacenar los datos de manera persistente para una sesión determinada. Ni bien se cierre el navegador los datos almacenados en la máquina del cliente a través de Session Storage serán borrados. Además, no es posible usar Session Storage a través de múltiples pestañas. La cantidad máxima de almacenamiento para cada ítem del SessionStorage es de 5MB

ii. Local Storage

Local Storage es el mecanismo elegido para almacenar datos de manera persistente a través de múltiples sesiones. Los datos, al igual que al usar cookies, son almacenados de manera persistente al contrario de SessionStorage. Sin embargo, a diferencia de los Cookies Local Storage sí fue creados específicamente para almacenar datos. Los datos no son enviados constantemente al servidor. La cantidad de datos que se nos permite almacenar crece dinámicamente según sea el requerimiento. LocalStorage es ideal para guardar grandes cantidades de datos en el lado del cliente.

4. comparación

Característica	Cookies	WebStorage
Tamaño máximo de almacenamiento:	4 KB	5 MB - Ilimitado
Cantidad máxima de instancias por sitio:	20	Ilimitado
Cantidad máxima de instancias por navegador:	300	Ilimitado
Se envía en cada solicitud HTTP	Si	No
Accesado por:	Cliente - Servidor	Cliente
Tipos especificaciones (alcance y vida)	Único, configurable	Local y Session

5. Conclusiones

A pesar de que Cookies pueden ser usadas para almacenar datos en el lado del cliente, este no fue su uso original provisto y por tanto las especificaciones no las hacen ideales para esto. La tabla anterior muestra claramente las ventajas de usar WebStorage cuando lo que se quiere es almacenamiento en el lado del cliente. Es de esperarse que en un futuro no muy lejano las especificaciones de WebStorage, ApplicationCache e IndexedDatabases reemplacen por

completo a Cookies cuando hablemos de almacenamiento en el lado del cliente. El uso de la web va en continuo crecimiento; las APIs también tienen que hacerlo.