Engineer

cookie

nombre y valor que el navegador devuelve al servidor

Qué es

Una cookie, o cookie HTTP, es un nombre y un valor breves que un servidor web envía al navegador en una cabecera de respuesta y que el navegador devuelve en peticiones posteriores al mismo servidor. Las cookies dan memoria a HTTP, un protocolo que trata cada petición como desconocida: una sesión iniciada que sobrevive al siguiente clic, una cesta que sigue llena al pagar o un idioma elegido una vez y conservado.

El servidor establece cada cookie con su propia cabecera `Set-Cookie` y el navegador devuelve todas las aplicables en una sola cabecera `Cookie` . Las tres primeras líneas son una respuesta y el resto una petición posterior del mismo navegador:

```
HTTP/1.1 200 OK
Set-Cookie: __Host-sid=4f2c9a71e0b8; Path=/; Secure; HttpOnly; SameSite=Lax
Set-Cookie: lang=da; Max-Age=31536000; Path=/

GET /account/ HTTP/1.1
Host: shop.example
Cookie: __Host-sid=4f2c9a71e0b8; lang=da
```

Todo lo que sigue al primer punto y coma es un atributo, una instrucción sobre cuánto tiempo conservar la cookie y adónde enviarla. `Expires` o `Max-Age` fija una duración máxima; sin ninguno, dura hasta que el navegador decide que ha terminado la sesión. `Domain` amplía su alcance a un dominio completo y sus subdominios; sin él, solo se devuelve al host que la estableció. `Path` lo restringe a una parte del sitio. `Secure` impide enviarla por conexiones sin cifrar, `HttpOnly` la oculta a los scripts que leen `document.cookie` y `SameSite` decide si acompaña a las peticiones iniciadas por otro sitio.

En el viaje de vuelta solo se envían nombres y valores, así que el servidor no puede saber por ellos cuándo caduca una cookie, si se estableció con `Secure` o quién la escribió por última vez.

Las reglas están en RFC 6265, publicado en 2011. Su revisión, conocida como rfc6265bis, añade `SameSite`, los prefijos de nombre `__Secure-` y `__Host-` y la exigencia de que los navegadores limiten cualquier duración a un máximo de 400 días. En el momento de escribir este artículo, está aprobada y a la espera de que RFC Editor la publique como sustituta de RFC 6265.

Utilice el concepto en una decisión

Aplique «¿Qué es una cookie?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Un sitio rápido, fácil de encontrar y editable por su propio personal, sin una factura por cada cambio de texto.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. 100 aplicaciones web con HTML/HTML5 — 100 aplicaciones web desarrolladas

100 aplicaciones web con HTML/HTML5 Póngase en contacto →

Por qué importa

Una cookie suele ser lo único que distingue una petición autenticada de otra anónima. Después de iniciar sesión, el identificador de sesión de la cookie representa al usuario para el servidor: quien lo presenta obtiene acceso a la cuenta sin que se le pida una contraseña. La mayoría de los problemas con las cookies consiste en que otra persona lea, copie o utilice una.

El navegador facilita el último caso. Adjunta las cookies a todas las peticiones que les correspondan, sin importar quién las inició. Un formulario de otro sitio que envía datos a una tienda llega con las cookies del visitante, y el servidor ve una petición de un usuario autenticado. RFC 6265 lo llama autoridad ambiental y enumera los ataques que facilita: la falsificación de peticiones entre sitios y el problema del representante confundido.

Las cookies son también la herramienta de seguimiento más antigua de la web. Un script incrustado en miles de sitios puede establecer una cookie de su propio dominio, que el navegador devuelve desde cada uno de esos sitios, permitiendo a una empresa relacionar las visitas de una persona entre todos ellos. La revisión

describe ese uso de cookies de terceros, señala que la mayoría de los navegadores ya las bloquea o particiona y recomienda una política tan restrictiva como permita la compatibilidad.

La ley sigue al almacenamiento, no a la tecnología. En la UE, el artículo 5(3) de la Directiva ePrivacy solo permite que un sitio almacene información en el dispositivo de un visitante o lea la que contiene con un consentimiento otorgado tras recibir información clara y completa. Las excepciones son el almacenamiento destinado únicamente a transmitir una comunicación y el estrictamente necesario para un servicio que el visitante haya solicitado expresamente. Ese artículo es el origen de los avisos de consentimiento y nunca menciona las cookies por su nombre. Las directrices del Comité Europeo de Protección de Datos sobre su ámbito técnico también lo aplican a píxeles y enlaces de seguimiento.

Cómo se aplica en la práctica

Este sitio no establece ninguna. Su página de cookies explica por qué: no hay cuentas, cesta ni nada con lo que seguir a un visitante, así que no existe aviso de consentimiento porque no hay nada que consentir. También enumera otros lugares donde un sitio puede dejar información, desde `localStorage` hasta un service worker, y declara que tampoco se utiliza ninguno.

Prescindir de cookies cambia cómo conoce un sitio a sus lectores. Contar lectores solo a partir del registro de acceso mide las lecturas sin cookies, scripts ni píxeles a partir del registro del propio servidor web y elimina de cada línea, al escribirla, la cabecera de petición `Cookie` junto con la de autorización. El resultado es deliberadamente poco detallado: no permite conocer visitantes únicos ni el recorrido de un visitante por el sitio, y la página presenta esas limitaciones junto a las cifras.

Cuando una aplicación sí necesita una cookie, la pregunta útil es qué se le confía. En una plataforma donde una persona actúa para varias organizaciones, la organización activa se guarda en el navegador y una cookie del lado del servidor la refleja, para que las páginas renderizadas en el servidor muestren el mismo contexto. Ninguna copia decide el acceso. El servidor comprueba la autorización en cada petición, de modo que editar el valor solo cambia lo que aparece en la pantalla de esa persona. Esa es la estructura segura de cualquier cookie: el valor puede elegir qué se renderiza y la decisión sobre lo que se permite hacer sigue en el servidor.

Dónde surgen los problemas

Un script de la página lee la sesión. Sin `HttpOnly`, un script inyectado por un atacante puede leer `document.cookie` y enviar el identificador de sesión a cualquier lugar. `HttpOnly` retira la cookie de esa API. Impide el robo, pero no el uso indebido: el script inyectado aún puede enviar peticiones y el navegador sigue adjuntando la cookie, así que `HttpOnly` acompaña a una Content Security Policy en lugar de sustituirla.

La cookie cruza HTTP sin cifrar. Una cookie sin `Secure` sale en cualquier petición `http://` a su host y puede leerla cualquiera situado en el recorrido de red. En RFC 6265, `Secure` solo protege la confidencialidad, y un atacante en la red todavía puede sobrescribir una cookie segura desde una conexión insegura. La revisión reduce esa brecha al ignorar cookies `Secure` establecidas por HTTP sin cifrar y rechazar una cookie de HTTP sin cifrar que se superpondría a otra segura.

Un subdominio hermano establece la cookie. `Domain=example.com` envía una cookie a todos los hosts bajo ese dominio, y cualquiera de ellos puede establecer una cookie para todo el dominio que el sitio principal acaba leyendo como propia. `Path` tampoco es una frontera de seguridad. El prefijo `__Host-` responde a ese problema: un navegador solo acepta una cookie así si es `Secure`, tiene `Path=/` y no tiene `Domain`, lo que la vincula exclusivamente al host que la estableció.

Otro sitio inicia la petición. `SameSite=Strict` restringe la cookie a peticiones del mismo sitio; `Lax` añade navegaciones de nivel superior con métodos seguros, como seguir un enlace; y `None` la envía a todas partes y se ignora salvo que la cookie también sea `Secure`. La revisión trata una cookie sin `SameSite` de forma muy similar a `Lax`; un servidor que dependa de ello debería declararlo expresamente. A estos efectos, un sitio es el esquema más el dominio registrable, por lo que `SameSite` no protege frente a un subdominio hermano hostil.

El servidor cree el valor. La persona cuyo navegador guarda una cookie puede editar cualquier parte de ella. RFC 6265 recomienda que los servidores cifren y firmen su contenido y advierte que aun así la cookie sigue siendo vulnerable a ataques de repetición. El patrón más sólido es un identificador de sesión aleatorio que apunte a un estado conservado en el servidor y se sustituya al iniciar sesión, para que un identificador colocado antes por un atacante, en el ataque de fijación de sesión del que advierte el RFC, no sirva después.

El navegador la descarta. Una fecha de caducidad es un máximo. Los navegadores expulsan cookies para mantenerse dentro de sus límites, y un atacante puede desalojarlas estableciendo muchas propias. Los servidores no deberían depender de que el navegador las conserve. Eliminarlas también tiene una trampa: un `Set-Cookie` con una caducidad pasada solo retira la cookie si sus valores de `Path` y `Domain` coinciden con los de su creación.

Lecturas adicionales

- RFC 6265: mecanismo de gestión de estado HTTP — el estándar de `Set-Cookie` y `Cookie`, con las reglas de atributos y una sección de seguridad sobre autoridad ambiental, integridad débil e identificadores de sesión.
- Cookies: mecanismo de gestión de estado HTTP (rfc6265bis) — la revisión aprobada, con `SameSite`, los prefijos `__Secure-` y `__Host-`, el máximo de 400 días y una sección sobre cookies de terceros.
- Directiva 2002/58/EC, texto consolidado — la Directiva ePrivacy modificada en 2009, con el artículo 5(3) sobre almacenar y leer información en el dispositivo de un usuario.
- Directrices 2/2023 del EDPB sobre el ámbito técnico del artículo 5(3) — cómo alcanza la norma a píxeles y enlaces de seguimiento, procesamiento local y otras técnicas más allá de las cookies.

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es una cookie?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/cookie/