Introducción a las Cookies en Express con JavaScript

Las cookies son pequeños fragmentos de datos almacenados en el navegador del cliente que permiten mantener información entre solicitudes HTTP. En aplicaciones web desarrolladas con **Express.js**, las cookies son útiles para manejar autenticación, preferencias del usuario y otras configuraciones persistentes.

En este documento, exploraremos cómo manejar cookies en un servidor Express utilizando JavaScript.

¿Qué son las Cookies?

Las cookies son enviadas y recibidas mediante cabeceras HTTP y se almacenan en el navegador del usuario. Cada cookie tiene un **nombre** y un **valor**, junto con atributos opcionales que definen su comportamiento, como:

- `expires` / `maxAge`: Tiempo de expiración de la cookie.
- `httpOnly`: Accesible solo a través de HTTP (no por JavaScript en el cliente).
- `secure`: Se envía solo a través de HTTPS.
- `sameSite`: Controla el envío de cookies en solicitudes entre sitios.
- `signed`: Permite firmar cookies para validar su integridad y prevenir manipulación.

Las cookies permiten mejorar la experiencia del usuario al proporcionar información persistente entre visitas o sesiones.

Instalación de Dependencias

Para trabajar con cookies en Express, primero instalamos las dependencias necesarias:

npm install express cookie-parser

Luego, en nuestro código, utilizamos `import` para importar los módulos.

Configuración y Uso de Cookies en Express

A continuación, configuramos un servidor Express y manejamos cookies:

1. Importar Express y Cookie-Parser

```
import express from 'express';
import cookieParser from 'cookie-parser';

const app = express();
const PORT = 3000;

// Establecer carpeta de archivos estáticos
app.use(express.static('public'));

// Middleware para manejar cookies firmadas
app.use(cookieParser('miClaveSecreta'));

app.listen(PORT, () => {
  console.log(`Server listening on port ${PORT}`);
});
```

2. Configurar Rutas para Gestionar Cookies

```
// Ruta para establecer una cookie con httpOnly
app.get('/set-httponly-cookie', (req, res) => {
    res.cookie('sessionID', 'abcdef123456', { maxAge: 3600000, httpOnly: true, secure: true,
sameSite: 'Strict' });
    res.send('Cookie httpOnly establecida');
});

// Ruta para leer una cookie httpOnly
app.get('/get-httponly-cookie', (req, res) => {
    const sessionID = req.cookies.sessionID;
    res.send(sessionID ? `ID de sesión: ${sessionID}` : 'No hay cookies de sesión
establecidas');
});

// Ruta para eliminar una cookie httpOnly
app.get('/delete-httponly-cookie', (req, res) => {
    res.clearCookie('sessionID');
    res.send('Cookie httpOnly eliminada');
});
```

3. Seguridad en el Uso de Cookies

Las cookies pueden ser vulnerables a ataques si no se configuran correctamente. Algunas medidas de seguridad incluyen:

- **`httpOnly`**: Evita que las cookies sean accedidas por JavaScript en el navegador, lo que protege contra ataques XSS (Cross-Site Scripting).
- **`secure`**: Permite que la cookie solo se envíe a través de conexiones HTTPS, asegurando que no sea interceptada en conexiones no seguras.
- **`sameSite`**: Reduce el riesgo de ataques CSRF (Cross-Site Request Forgery) asegurando que la cookie solo se envíe en solicitudes de la misma página de origen.
- **Uso de Cookies Firmadas**: Para prevenir la manipulación de cookies en el lado del cliente.

4. Ataques XSS y CSRF

- **XSS (Cross-Site Scripting)**: Es un ataque en el que un atacante inyecta código malicioso en una página web para ejecutar scripts en el navegador de un usuario. Si una cookie no está protegida con `httpOnly`, un atacante podría robarla y suplantar la sesión del usuario.
- **CSRF (Cross-Site Request Forgery)**: Es un ataque en el que un usuario autenticado es engañado para realizar acciones no autorizadas en un sitio web sin su consentimiento. Las cookies con `sameSite: 'Strict'` ayudan a mitigar este tipo de ataques.

Ejemplo de cómo establecer una cookie segura y firmada:

```
app.get('/set-secure-signed-cookie', (req, res) => {
    res.cookie('secureToken', 'randomSecureToken123', {
        maxAge: 60000,
        httpOnly: true,
        secure: true,
        sameSite: 'Strict',
        signed: true
    });
    res.send('Cookie segura y firmada establecida');
});

app.get('/get-secure-signed-cookie', (req, res) => {
    const secureToken = req.signedCookies.secureToken;
    res.send(secureToken ? `Token validado: ${secureToken}` : 'No hay token válido');
});
```

Ventajas y Desventajas del Uso de Cookies

Ventajas:

- **Persistencia de Datos**: Permiten mantener información entre diferentes solicitudes sin necesidad de almacenar datos en el servidor.
- **Facilidad de Implementación**: Se pueden manejar fácilmente con Express y librerías como `cookie-parser`.
- **Mejora la Experiencia del Usuario**: Permiten recordar preferencias, sesiones y configuraciones personalizadas.

Desventajas:

- **Limitación de Espacio**: Tienen un tamaño máximo de aproximadamente 4 KB, lo que puede ser insuficiente para datos complejos.
- **Riesgo de Seguridad**: Si no se configuran adecuadamente (`httpOnly`, `secure`, `sameSite`), pueden ser vulnerables a ataques XSS y CSRF.
- **Almacenamiento en el Cliente**: No son ideales para datos sensibles, ya que los usuarios pueden acceder y modificar las cookies almacenadas en su navegador.

Ejemplos de Uso y Tipo de Información que se Puede Guardar

Las cookies pueden utilizarse para almacenar diversos tipos de información, tales como:

- **Identificadores de sesión**: Para mantener a un usuario autenticado.
- **Preferencias del usuario**: Como el idioma seleccionado o el tema de la interfaz.
- **Historial de navegación**: Para personalizar la experiencia del usuario.
- **Tokens de seguridad**: Para prevenir ataques CSRF o manejar la autenticación.

Conclusión

El uso de cookies en Express es fundamental para la gestión de sesiones y datos persistentes del usuario. Con `cookie-parser`, podemos manejar cookies de manera sencilla y segura.

Las cookies seguras, firmadas y con `httpOnly` permiten mejorar la protección de datos en las aplicaciones web, reduciendo riesgos de exposición y aumentando la privacidad del usuario. Su uso debe estar alineado con buenas prácticas de seguridad para evitar vulnerabilidades.

Regulaciones Legales sobre el Uso de Cookies

El uso de cookies en aplicaciones web está sujeto a diversas normativas y regulaciones en distintas jurisdicciones. Es fundamental que los desarrolladores y administradores de sitios web comprendan estas regulaciones para evitar sanciones y garantizar la privacidad del usuario.

> **Nota:** Esta lista incluye algunas de las principales regulaciones sobre cookies en diferentes países, pero no es exhaustiva. Se recomienda verificar la legislación vigente en cada jurisdicción específica.

---

1. Reglamento General de Protección de Datos (GDPR - Unión Europea)

El **GDPR** regula la recopilación y el uso de datos personales, incluidas las cookies. Sus principales requisitos incluyen:

- **Consentimiento explícito** antes de almacenar cookies no esenciales en el dispositivo del usuario.
- **Opción de aceptar o rechazar cookies**, con una interfaz accesible y clara.
- **Política de cookies detallada**, explicando su propósito, duración y uso.
- **Derecho a revocar el consentimiento** en cualquier momento.

---

2. Ley de Privacidad del Consumidor de California (CCPA - Estados Unidos)

La **CCPA** otorga a los residentes de California ciertos derechos sobre sus datos personales, incluidas las cookies:

- Los usuarios deben ser informados sobre la recopilación de datos mediante cookies.
- Deben tener la opción de **optar por no participar** en la venta de sus datos personales.
- Derecho a **solicitar la eliminación de datos** recopilados por cookies.

---

3. Directiva de ePrivacy (UE - Cookie Law)

Esta directiva complementa el GDPR y establece requisitos específicos para el uso de cookies:

- Las cookies **no esenciales** (publicidad, análisis, etc.) solo pueden usarse con el **consentimiento del usuario**.
- No se requiere consentimiento para cookies esenciales (como las necesarias para el funcionamiento del sitio).

---

4. Otras Regulaciones Globales

- **Argentina (Ley de Protección de Datos Personales 25.326)**: Exige el consentimiento informado del usuario antes de la recopilación de datos personales mediante cookies y otorga derechos de acceso, rectificación y supresión de datos.
- **Brasil (LGPD)**: Exige consentimiento explícito para el uso de cookies y otorga derechos a los usuarios sobre sus datos.
- **Canadá (PIPEDA)**: Requiere transparencia y consentimiento informado para la recopilación de datos mediante cookies.
- **Australia (Privacy Act 1988)**: Regula el uso de datos personales y exige políticas claras sobre la recopilación de cookies.
- **Otros países**: Muchas otras naciones han implementado regulaciones similares a GDPR y CCPA. Se recomienda consultar la legislación local para asegurar el cumplimiento.

---

5. Cómo Cumplir con las Regulaciones

Para garantizar el cumplimiento legal, se recomienda:

- **Implementar banners de consentimiento** para permitir a los usuarios aceptar o rechazar cookies.
- **Clasificar las cookies** según su propósito (necesarias, funcionales, analíticas, marketing).
- **Publicar una política de cookies** clara y accesible en el sitio web.
- **Actualizar la política de privacidad** regularmente según el uso de cookies.
- **Permitir la gestión de preferencias**, para que los usuarios puedan cambiar su consentimiento en cualquier momento.