Open Opened 5 days ago by **Rubén Montero**

# Hacer login... ¡con cookies!

## Resumen

- Explicaremos qué son las *cookies*
- Hablaremos de un flujo de autenticación a través de *cookies*
- Crearemos una nueva página que permite enviar una petición de *login* en la que recibimos una cookie
- Verificaremos que está ahí inspeccionando `document.cookie` en la consola

## Descripción

Hemos implementado un *login* que nos ha permitido entender cómo funciona la autenticación.

Nuestro flujo es:

1. Enviamos el usuario y la contraseña al servidor
2. En respuesta, recibimos un *token*
3. Lo guardamos (con `sessionStorage`)
4. Lo enviamos en futuras peticiones que requieren autenticación

Pero esto... ¡no es una buena práctica!

`sessionStorage` es una mala herramienta para guardar datos sensibles, ya que es vulnerable a ataques como XSS.

Una solución más profesional consiste en usar *Cookies*.

### ¿Qué es una *cookie*?

Una *cookie* es un par clave-valor que el servidor nos envía y nuestro navegador remite de vuelta en las peticiones.

Cuando el servidor nos manda una *cookie*, usa la cabecera HTTP estándar `Set-Cookie`:

```
Set-Cookie: SESSION_TOKEN=7392b38de92a93bc973b0affe
```

Cuando nuestro navegador ha recibido una *cookie*, entonces la manda de vuelta al servidor dentro de la cabecera HTTP estándar `Cookie`:

```
Cookie: SESSION_TOKEN=7392b38de92a93bc973b0affe
```

Las *cookies* tienen muchos usos. Uno de los principales es implementar *autenticación mediante sesiones*.

En lugar de usar nosotros `sessionStorage` *a mano*, podemos confiar en que el navegador almacena las *cookies* de forma segura.

### ¿Y qué vamos a hacer?

Crearemos una nueva página de *login* que atacará el siguiente *endpoint* de nuestro servidor (enviaremos *username* y *password* en el cuerpo de la petición):

```
POST http://raspi:8082/api/v3/sessions
```

Como ves, a diferencia del *endpoint* que atacamos actualmente, éste tiene `/v3/` en vez de `/v2/`.

Esta nueva versión del *endpoint* está diseñada para responder con algo como esto (por ejemplo):

```
201 Created
Set-Cookie: SESSION_TOKEN=7392b38de92a93bc973b0affe

{ "session_id": 14 }
```

¡Recibiremos el *token de sesión* en una `cookie`!

## La tarea

**Crea** una nueva pantalla `src/screens/user/better_login/BetterUserLogin.js`.

`BetterUserLogin.js` **será idéntico** a `UserLogin.js` (cópialo y pégalo).

## Atacar el *login* `/v3/` en vez de `/v2/`

**Modifica** tu función `onSubmit` en `BetterUserLogin.js` y **elimina** las líneas dentro de la función que gestiona `response` **así**:

```
-    axios.post('http://raspi:8082/api/v2/sessions', { username: username, password: password }).then(response => {
+    axios.post('http://raspi:8082/api/v3/sessions', { username: username, password: password }).then(response => {
-     console.log(response.data.session_token);
-     console.log(response.data.session_id);
-     sessionStorage.setItem('SESSION_TOKEN', response.data.session_token);
-     sessionStorage.setItem('SESSION_ID', response.data.session_id);
-     navigate('/');
-     setLoggedIn(true);
    });
```

Luego, simplemente, **añade** un par de `console.log` :

```
    axios.post('http://raspi:8082/api/v3/sessions', { username: username, password: password }).then(response => {
      console.log("Las cookies actuales son las siguientes:");
      console.log(document.cookie);
    });
```

## Conectándolo en `App.js`

**Añade** en `App.js` :

```
      <Route path='/better_login' element={<BetterUserLogin/>}/>
```

Ahora ya estará visible la página http://localhost:3000/better_login

¿Qué tal se ve?

Si tecleas tu usuario y contraseña y te *logueas* no deberías ver nada diferente.

Sin embargo, **haz** Click derecho > Inspeccionar y **verifica** lo que aparece por consola. ¿Ves que en `document.cookie` ahora podemos ver la *cookie* que nos ha mandado el servidor?

## ¿Y esa *cookie* sirve para algo?

De momento, no.

Para no alargar demasiado el *sprint*, no existe una versión `/v3/` del resto de *endpoints* del API REST que funcionen a través de la *cookie*.

Sin embargo, si viajas a tu página http://localhost:3000 y haces click en Inspeccionar > Red... ¿Cómo son las peticiones que tu navegador manda a http://raspi:8082/api/v2/dashboards? ¿Se está mandando la *cookie*? ¿Crees que el servidor podría confiar en tu identidad gracias a dicha *cookie*?

## 🏆 Por último

Sube tus cambios al repositorio en un nuevo *commit*.

---

Rubén Montero @ruben.montero changed milestone to %Sprint 5 5 days ago