

Login / Logout con Firebase



Login / Logout con Firebase

En esta lección, vamos a implementar la funcionalidad de login y logout en la aplicación, integrando Firebase para gestionar la autenticación de los usuarios. El objetivo es mostrar un link de "Login" si el usuario no está autenticado, y un link de "Salir" si el usuario ya ha iniciado sesión.

Modificación del Servicio de Login

Primero, agregamos el método `logout` al servicio de autenticación (`LoginService`). Este servicio manejará tanto el inicio como el cierre de sesión, así como la validación del estado de autenticación.

Código: `login.service.ts`

```
import { Injectable } from '@angular/core';
import { Router } from '@angular/router';
import { FirebaseService } from '../firebase.service';
import { signInWithEmailAndPassword } from 'firebase/auth';

@Injectable({
  providedIn: 'root',
})
```

```
export class LoginService {
  token: string | null = null; // Almacena el token de autenticación del usuario

  constructor(
    private router: Router,
    private firebaseService: FirebaseService
  ) {}

  // Método para iniciar sesión
  login(email: string, password: string) {
    const auth = this.firebaseService.auth;

    signInWithEmailAndPassword(auth, email, password)
      .then(() => {
        auth.currentUser?.getIdToken().then((token) => {
          this.token = token; // Almacena el token obtenido de Firebase
          this.router.navigate(['/']); // Redirige a la página principal tras el login
        });
      })
      .catch((error) => {
        console.error('Error al iniciar sesión:', error); // Manejo de errores de login
      });
  }

  // Método para obtener el token actual
  getIdToken() {
    return this.token;
  }

  // Verifica si el usuario está autenticado
  isAuthenticated() {
    return this.token != null;
  }

  // Método para cerrar sesión
  logout() {
    const auth = this.firebaseService.auth;
    auth
      .signOut()
      .then(() => {
        this.token = null; // Resetea el token al cerrar sesión
        this.router.navigate(['login']); // Redirige al login después del logout
      })
      .catch((error) => console.log('error logout: ' + error)); // Manejo de errores de
logout
  }
}
```

```
}  
}
```

Explicación del Código:

- `login()`: Utiliza Firebase para iniciar sesión con correo y contraseña. Al iniciar sesión correctamente, almacena el token de autenticación y redirige al usuario a la página principal.
- `isAutenticado()`: Verifica si hay un token almacenado, lo que indica si el usuario está autenticado.
- `logout()`: Cierra la sesión del usuario, borra el token y redirige al login.

Modificación del Componente Principal (AppComponent)

El siguiente paso es modificar el componente principal de la aplicación para que muestre un link de "Login" cuando el usuario no está autenticado, y un link de "Salir" cuando el usuario ya inició sesión.

Código: `app.component.ts`

```
import { Component } from '@angular/core';  
import { RouterModule, RouterOutlet } from '@angular/router';  
import { LoginService } from './login.service';  
  
@Component({  
  selector: 'app-root',  
  standalone: true,  
  imports: [RouterOutlet, RouterModule],  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
  titulo = 'Tienda Online';  
  
  constructor(private loginService: LoginService) {}  
  
  // Método que verifica si el usuario está autenticado  
  isAutenticado() {  
    return this.loginService.isAutenticado();  
  }  
  
  // Método para cerrar sesión  
  salir() {  
    this.loginService.logout();  
  }  
}
```

```
}
}
```

Explicación del Código:

- **isAutenticado()**: Invoca el método del servicio de login para determinar si el usuario está autenticado.
- **salir()**: Invoca el método de logout del servicio de login para cerrar la sesión del usuario y redirigir al login.

Modificación de la Vista (`app.component.html`)

El siguiente código HTML agrega los botones de **Login** y **Salir** estilizados con Bootstrap. Dependiendo del estado de autenticación del usuario, se muestra uno u otro botón.

Código: `app.component.html`

```
<div class="container text-center my-3">
  <h1 class="text-info">{{titulo}}</h1>

  <!-- Links de Login y Salir con Bootstrap -->
  <div class="d-flex justify-content-center my-3">
    @if (!isAutenticado()) {
      <a [routerLink]="['/login']" class="btn btn-outline-primary mx-2">Login</a>
    } @else {
      <a style="cursor: pointer" class="btn btn-outline-danger mx-2"
(click)="salir()">Salir</a>
    }
  </div>

  <router-outlet></router-outlet>
</div>
```

Explicación del Código:

- **Condicional @if**: Se utiliza para mostrar los botones de **Login** o **Salir** en función de si el usuario está autenticado.
- **Bootstrap**: Se utilizan las clases `btn btn-outline-primary` para estilizar el botón de **Login**, y `btn btn-outline-danger` para el botón de **Salir**.
- **Enlace routerLink**: El botón de **Login** redirige al usuario a la página de inicio de sesión.

Resultado Final


Después de implementar estos cambios, la interfaz mostrará un botón de **Login** si el usuario no está autenticado, y un botón de **Salir** si ya ha iniciado sesión.




Tienda Online

Login

Iniciar Sesión

Email 

Password 

Iniciar Sesión



Tienda Online

Salir

Listado de Productos

Agregar Producto

Pantalón, \$130	Editar Producto
Camisa, \$80	Editar Producto
Playera, \$50	Editar Producto

Conclusión

Con esta lección hemos logrado implementar la autenticación con Firebase, añadiendo un control sobre el login y logout del usuario, y actualizando la interfaz para mostrar los botones correctos según el estado de autenticación del usuario.

¡Saludos!

Ing. Ubaldo Acosta

Fundador de GlobalMentoring.com.mx