**Registro con servicios de Google**

**Objetivo**: Implementar el registro de usuarios utilizando los servicios de Google en tu aplicación Angular.

**Recursos necesarios:**

* Una cuenta de desarrollador en Google: Debes crear un proyecto en la Consola de Desarrolladores de Google y obtener las credenciales necesarias para utilizar los servicios de Google en tu aplicación.
* Angular CLI: Asegúrate de tener Angular CLI instalado en tu entorno de desarrollo.

**Pasos:**

**1. Configurar el proyecto en la Consola de Desarrolladores de Google:**

* Crea un proyecto en la Consola de Desarrolladores de Google y habilita los servicios necesarios, como Google Sign-In.
* Obtiene las credenciales de cliente para la autenticación con Google. Puedes obtener el ID de cliente y la clave de cliente.

**2. Instalar el paquete angularx-social-login:**

* En la terminal, ejecuta el siguiente comando para instalar el paquete angularx-social-login:

ng add angularx-social-login

**3. Configurar el módulo de autenticación con Google:**

* En el archivo app.module.ts, importa SocialLoginModule de angularx-social-login y configura la inicialización del proveedor de autenticación de Google utilizando las credenciales obtenidas en el paso 1.
* Asegúrate de proporcionar el ID de cliente y la clave de cliente en la configuración del proveedor.

**4. Implementar el botón de registro con Google:**

* En el componente donde desees implementar el registro con Google, importa SocialAuthService de angularx-social-login y configura la lógica necesaria.
* Agrega un botón en tu plantilla HTML con un evento de clic asociado a un método en el componente.
* En el método del componente, utiliza SocialAuthService para iniciar el flujo de autenticación con Google.

Ejemplo de código en el componente:

import { Component } from '@angular/core';

import { SocialAuthService, GoogleLoginProvider } from 'angularx-social-login';

@Component({

selector: 'app-register',

templateUrl: './register.component.html',

styleUrls: ['./register.component.css']

})

export class RegisterComponent {

constructor(private authService: SocialAuthService) {}

signInWithGoogle(): void {

this.authService.signIn(GoogleLoginProvider.PROVIDER\_ID)

.then((response) => {

// La autenticación con Google fue exitosa, puedes acceder a la información del usuario en response

// Realiza las acciones necesarias, como el registro en tu backend, utilizando los datos proporcionados por Google

})

.catch((error) => {

// La autenticación con Google falló, maneja el error adecuadamente

});

}

}

**5. Manejar la respuesta de autenticación:**

* En el método del componente asociado al evento de clic del botón de registro con Google, utiliza SocialAuthService para iniciar el flujo de autenticación con Google.
* Suscríbete al observable de autenticación para recibir la respuesta de autenticación.
* En el callback de éxito, obtén la información del usuario autenticado y realiza las acciones necesarias, como el registro en tu backend.

import { Component } from '@angular/core';

import { SocialAuthService, GoogleLoginProvider, SocialUser } from 'angularx-social-login';

@Component({

selector: 'app-register',

templateUrl: './register.component.html',

styleUrls: ['./register.component.css']

})

export class RegisterComponent {

user: SocialUser | null;

constructor(private authService: SocialAuthService) {}

signInWithGoogle(): void {

this.authService.signIn(GoogleLoginProvider.PROVIDER\_ID)

.then((response) => {

this.user = response;

// La autenticación con Google fue exitosa, puedes acceder a la información del usuario en response

// Realiza las acciones necesarias, como el registro en tu backend, utilizando los datos proporcionados por Google

console.log('Usuario autenticado:', this.user);

// Llamar a tu servicio de registro con los datos del usuario

})

.catch((error) => {

// La autenticación con Google falló, maneja el error adecuadamente

console.error('Error en la autenticación con Google:', error);

});

}

signOut(): void {

this.authService.signOut()

.then(() => {

this.user = null;

// Realizar cualquier acción adicional después de cerrar sesión

})

.catch((error) => {

console.error('Error al cerrar sesión:', error);

});

}

}