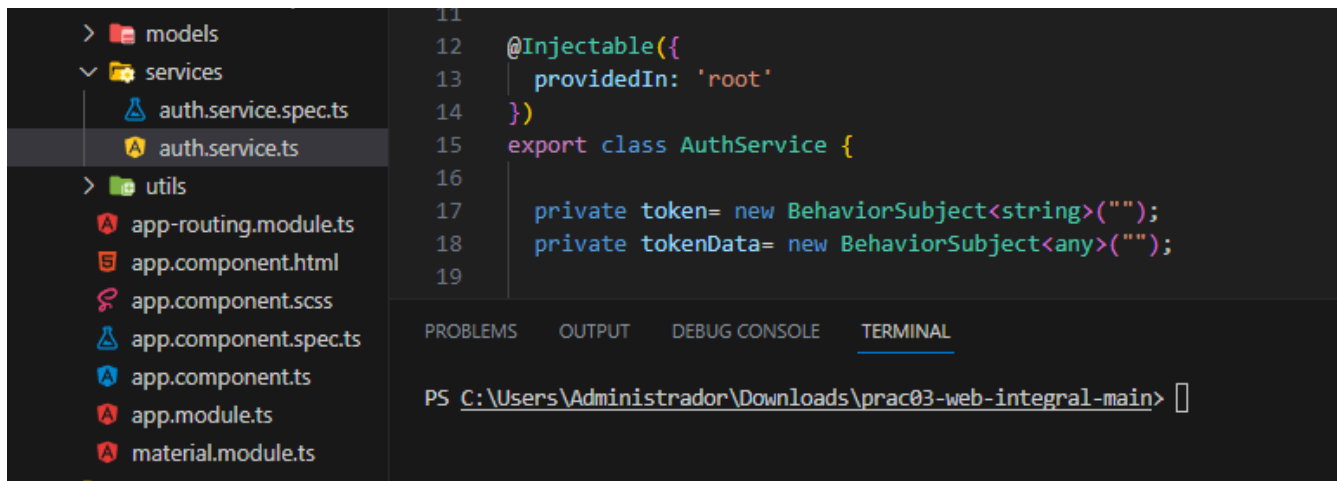


## CONTINUACION FRONT-END

### Crear servicio de Autenticación

Primeramente ,agregar módulo [MatSnackBarModule](#) al archivo `material.module.ts` para usar el servicio de mensajes; modificar el archivo `auth.service.ts`

Abrir terminal y ejecutar el siguiente comando para crear el servicio de autenticación: `ng g s shared/services/auth`



```
11
12 @Injectable({
13   providedIn: 'root'
14 })
15 export class AuthService {
16
17   private token= new BehaviorSubject<string>("");
18   private tokenData= new BehaviorSubject<any>("");
19
```

```
import { Router } from '@angular/router';
import { BehaviorSubject, Observable, catchError, map, throwError } from 'rxjs';
```

```
@Injectable({
  providedIn: 'root'
})
export class AuthService {

  private token= new BehaviorSubject<string>("");
  private tokenData= new BehaviorSubject<any>("");

  constructor(private http:HttpClient,private router:Router, private snackBar:MatSnackBar) { }
```

Agregar las siguientes instrucciones al archivo `auth.service.ts` con el objetivo de definir las propiedades `token` y `tokenData` como Observables y permitir suscripciones y saber de sus valores

```
  constructor(private http:HttpClient,private router:Router, private snackBar:MatSnackBar) { }

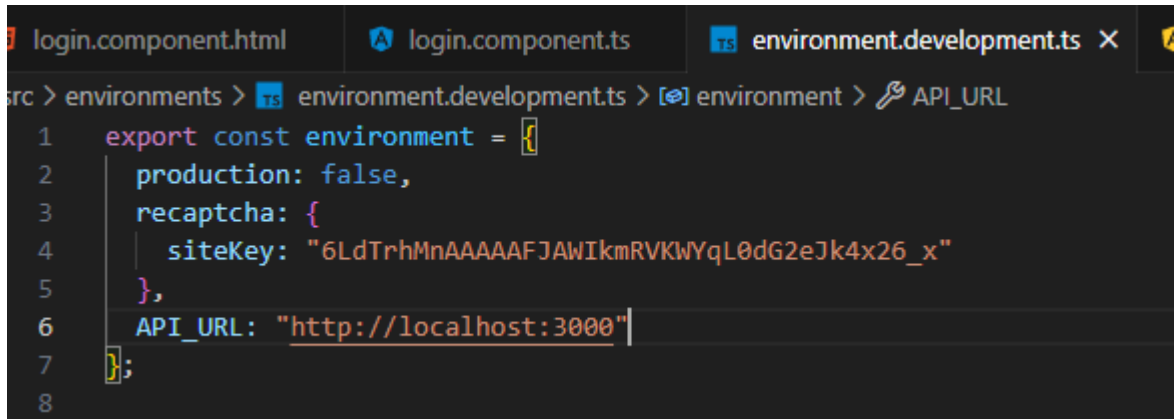
  get token$(): Observable<string> {
    return this.token.asObservable();
  }

  get tokenData$(): Observable<any> {
    return this.tokenData.asObservable();
  }
}
```

el utilizar environments en tu aplicación es con el objetivo de utilizar API KEY para entornos de pruebas y de producción durante el desarrollo de la aplicación.

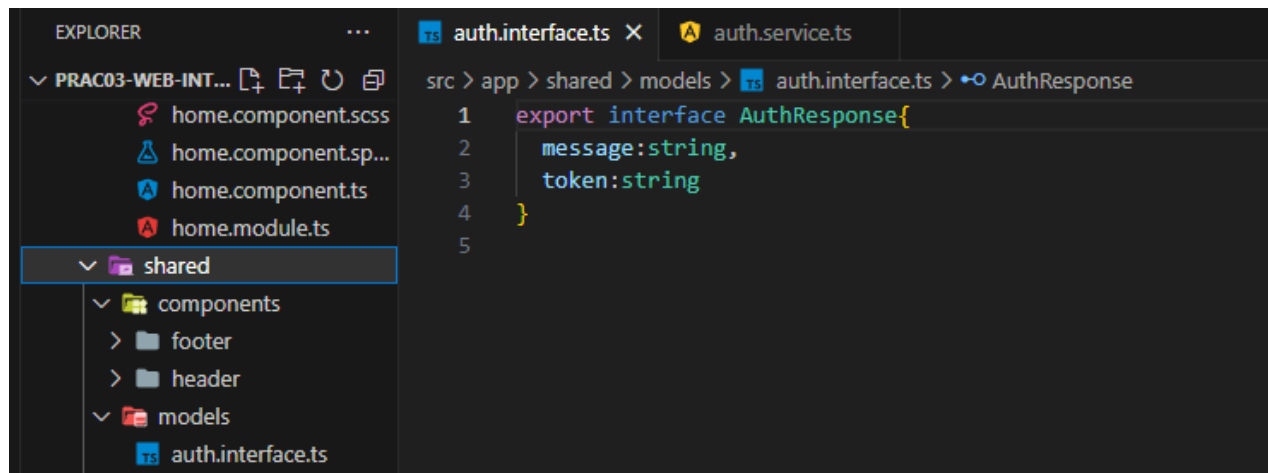
En la terminal ejecutar el siguiente comando: **ng g environments**

Modificar los archivos environments de la siguiente forma



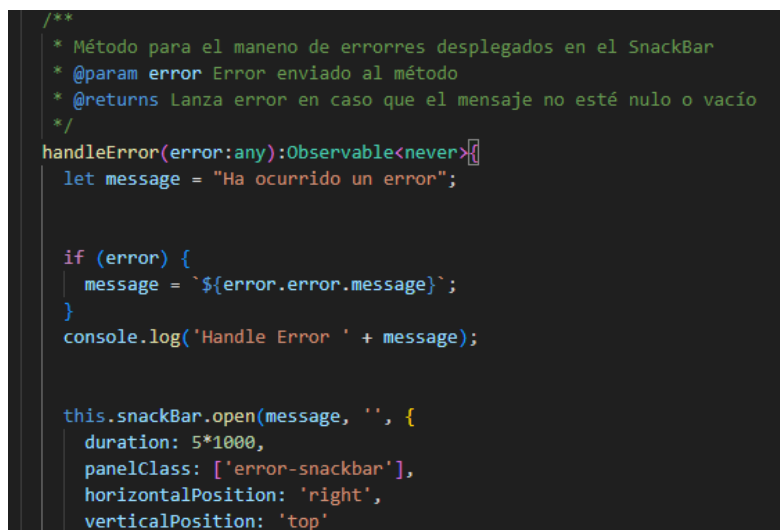
```
src > environments > environment.development.ts > environment > API_URL
1  export const environment = {
2    production: false,
3    recaptcha: {
4      siteKey: "6LdTrhMnAAAAAFJAWIkRVKwYqL0dG2eJk4x26_x"
5    },
6    API_URL: "http://localhost:3000"
7  };
8
```

## MODELO



```
EXPLORER
PRAC03-WEB-INT...
  home.component.scss
  home.component.sp...
  home.component.ts
  home.module.ts
  shared
    components
    footer
    header
    models
      auth.interface.ts

src > app > shared > models > auth.interface.ts > AuthResponse
1  export interface AuthResponse{
2    message:string,
3    token:string
4  }
5
```



```
/**
 * Método para el manejo de errores desplegados en el SnackBar
 * @param error Error enviado al método
 * @returns Lanza error en caso que el mensaje no esté nulo o vacío
 */
handleError(error:any):Observable<never>{
  let message = "Ha ocurrido un error";

  if (error) {
    message = `${error.error.message}`;
  }
  console.log('Handle Error ' + message);

  this.snackBar.open(message, '', {
    duration: 5*1000,
    panelClass: ['error-snackbar'],
    horizontalPosition: 'right',
    verticalPosition: 'top'
  });
}
```

## GUARDADO DE SESION

Instalar el módulo [@auth0/angular-jwt](#)

### Installation

```
# installation with npm
npm install @auth0/angular-jwt

# installation with yarn
yarn add @auth0/angular-jwt
```

```
import { BehaviorSubject, Observable, catchError, map, throwError } from 'rxjs';
import { JwtHelperService } from '@auth0/angular-jwt';
```

```
const helper = new JwtHelperService;
```

## LOCAL STORAGE

```
}

saveLocalStorage(token:string){
  localStorage.setItem('token', token);
}
```

```
logout(){
  localStorage.removeItem('token');
  this.token.next(""); //Notofica a suscirptores que es vacio
  this.tokenData.next(null); //Notifica a los suscriptores.
  this.router.navigate(['/']);
}

/**
```

```

/**
 * Cada vez que se realice una petición verifica que Token
 */
checkToken() {
  let token = localStorage.getItem("token");

  if(token) {
    //Checa expiración de token traído del BackEnd
    const isExpired = helper.isTokenExpired(token);

    if(isExpired) {
      this.logout(); //Cierra sesión
    } else {
      this.token.next(token); //Notifica token a suscriptores

      //Decodifica token JWT Encriptado
      const {iat, exp, ...data} = helper.decodeToken(token);
      this.tokenData.next(data); //Notifica token a suscriptores
    }
  }
}

```

```

* Hace la petición al BackEnd
* @param loginData Usuario y Contraseña de usuario
* @returns
*/
login(loginData:any):Observable<AuthResponse|void> {

  return this.http.post<AuthResponse>(`${environment.API_URL}/`, loginData)
    .pipe(map((data:AuthResponse) => {

      if(data.token) { //No hay errores y contiene información
        this.saveLocalStorage(data.token);
        this.token.next(data.token); //Notifica a suscriptores
        this.router.navigate(['/home']);

        this.checkToken();
      }
      return data;
    })),
    catchError((error) => this.handleError(error)));
}

```

```

service.ts x material.module.ts header.component.scss header.component.html x environment.ts environment.devel
src > app > shared > components > header > header.component.html > mat-toolbar > button.example-icon.favorite-icon
Go to component
1 <mat-toolbar>
2   <span>Desarrollo web Integral</span>
3   <span class="example-spacer"></span>
4
5   <button mat-icon-button class="example-icon favorite-icon" aria-label="Example icon-button with heart icon">
6     <mat-icon>favorite</mat-icon>
7   </button>
8   <button mat-icon-button class="example-icon" aria-label="Example icon-button with share icon">
9     <mat-icon>share</mat-icon>
10  </button>
11  <button mat-button>{{data?.username}}</button>
12  <button mat-icon-button (click)="onLogout()">
13    <mat-icon>login</mat-icon>
14  </button>
15
16 </mat-toolbar>
17

```

phpmyadmin

test

web\_integrat

Nueva

tbl\_usuario

Columnas

Nueva

apellido (varchar)

nombre (varchar)

password (varchar, NULL, nullable)

role (varchar, NULL, nullable)

username (PK, varchar)

Indice

#	Nombre	Tipo	Cotejamiento	Atributos	Null	Predefinido	Comentarios	Estra	Acción
1	username	varchar(30)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
2	password	varchar(255)	utf8mb4_general_ci		Si	NULL			Cambiar Eliminar Más
3	role	varchar(20)	utf8mb4_general_ci		Si	NULL			Cambiar Eliminar Más
4	nombre	varchar(255)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
5	apellido	varchar(255)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

☐ Seleccionar todo
 Para los elementos que están marcados:
 ☐ Examinar
 ☐ Cambiar
 ☐ Eliminar
 ☐ Primaria
 ☐ Único
 ☐ Índice

☐ Agregar a columnas centrales
 ☐ Eliminar de las columnas centrales

```

{
  "username": "PaolaPG",
  "password": "$2a$10$wADE2UUsyedFWlNzZhYuDOJATuTZipUmycN0r.isaQFpxqyQZsF82",
  "role": "admin",
  "nombre": "Paola",
  "apellido": "Patlan G"
},

```

RESULTADO

