

Enviando el token a los demás métodos



Enviando el token a los demás métodos

En esta lección vamos a modificar la clase de servicio para enviar el token a los demás métodos y así poder validar las acciones de agregar, modificar o eliminar productos.

Código archivo datos.service.ts:

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Producto } from '../producto/producto.model';
import { Observable } from 'rxjs';
import { LoginService } from '../login.service';

@Injectable({
  providedIn: 'root',
})
export class DatosService {
  url = 'https://tienda-online-7f4e5-default-rtdb.firebaseio.com/';

  constructor(
    private httpClient: HttpClient,
    private loginService: LoginService
  ) {}
}
```

```
cargarProductos(): Observable<{ [llave: string]: Producto }> {  
  const token = this.loginService.getIdToken();  
  return this.httpClient.get<{ [llave: string]: Producto  
}>(`${this.url}datos.json?auth=${token}`);  
}  
  
guardarProducto(producto: Producto): Observable<any> {  
  const token = this.loginService.getIdToken();  
  return this.httpClient.post(`${this.url}datos.json?auth=${token}`, producto);  
}  
  
modificarProducto(producto: Producto, llave: string): Observable<any> {  
  const token = this.loginService.getIdToken();  
  const url_modificar = `${this.url}datos/${llave}.json?auth=${token}`;  
  return this.httpClient.put(url_modificar, producto);  
}  
  
eliminarProducto(llave: string): Observable<any> {  
  const token = this.loginService.getIdToken();  
  const url_eliminar = `${this.url}datos/${llave}.json?auth=${token}`;  
  return this.httpClient.delete(url_eliminar);  
}  
}
```

Con estos cambios ya podremos agregar, modificar o eliminar un producto después de haber hecho login en nuestra aplicación.

Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](http://www.globalmentoring.com.mx)