Manejo de headers 17/23

RECURSOS APUNTES

Luego de que el usuario se halla registrado, puedes utilizar el token para realizar peticiones al backend que requieran de autenticación. Para esto, es necesario inyectar dicho token en las solicitudes HTTP.

Autenticación del Usuario

Para esto, puedes obtener el token desde *Local Storage* (o si prefieres guardarlo en Cookies) e inyectarlo directamente en los *Headers* de la petición.

```
// services/auth.service.ts
getProfileUser(): Observable<User> {
  const token = localStorage.getItem('platzi_token');
  return this.http.get<User>(`https://example.com/api/profile`, {
    headers: {
      Authorization: `Bearer ${token}`
    }
  });
}
```

El protocolo de este tipo de token suele emplear la palabra **Bearer** seguido de un espacio por delante del propio token.

Puede ser bastante engorroso tener que inyectar el token, método por método en todos los *endpoints* que necesitas consumir. Puedes simplificar esto con una única función que construya los headers.

```
// services/auth.service.ts
import { HttpHeaders } from '@angular/common/http';

getProfileUser(): Observable<User> {
  return this.http.post<User>(`https://example.com/api/profile`, this.getHttpHeaders());
}

getHttpHeaders() {
  const token = localStorage.getItem('platzi_token');
  return {
    headers: new HttpHeaders({
      Authorization: `Bearer ${token}`
    })
  };
}
```

La clase `HttpHeaders` construye los header en la función `getHttpHeaders()` y llamas a dicha función para que inyecte los headers en cada solicitud.

Es una forma más elegante y limpia de inyectar los headers necesarios en tus solicitudes.
