



Enviar Token con un interceptor 19/23



RECURSOS

APUNTES

El uso principal para los Interceptores es para la inyección del token en las request.

Inyección de token

Veamos cómo puedes utilizar los interceptores para la inyección de token a las solicitudes HTTP.

1. Crea el interceptor

Para esto, crea un nuevo interceptor que lea el token desde el storage donde decidiste guardarlo y posteriormente inyectarlo en las peticiones.

```
// interceptors/token-interceptor.interceptor.ts
@Injectable()
export class TokenInterceptorService implements HttpInterceptor {

  intercept(request: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<unknown>>
  {
    request = this.addHeaders(request);
    return next.handle(request)
  }

  private addHeaders(request: HttpRequest<any>) {
    let token: string | null = '';
  }
}
```

```

    token = localStorage.getItem('platzi_token');
    if (token) {
      return request.clone({
        setHeaders: {
          Authorization: `Bearer ${token}`
        }
      });
    } else {
      return request;
    }
  }
}

```

2. Duplicando la solicitud

La función `addHeaders()` recibe la request y usando `clone()` crea una copia de si misma para inyectar el token. De no existir este, devuelve la request tal cual fue recibida.

De esta manera, limpias por completo tu servicio que se ocupa de la autenticación de usuarios y centralizas toda la lógica de inyección de headers en el interceptor.

Y no olvides de importar el nuevo interceptor en tu módulo.

```

// app.module.ts
import { ErrorsInterceptor } from './interceptors/errors.interceptor';
import { TokenInterceptorService } from './interceptors/token-interceptor.interceptor';

@NgModule({
  providers: [
    { provide: HTTP_INTERCEPTORS, useClass: ErrorsInterceptor, multi: true },

```

```
    { provide: HTTP_INTERCEPTORS, useClass: TokenInterceptorService, multi: true }  
  ]  
})  
export class AppModule { }
```

Agrégallo a los `providers` y si ya tienes otro interceptor, también agrega la propiedad `multi: true` a cada uno de ellos.

Contribución creada por: Kevin Fiorentino.

Archivos de la clase

 /  



tsconfig.app.json



package.json



angular.json



.eslintrc.json



package-lock.json




.browserslistrc




.gitignore



 obs-promises.js




 tsconfig.spec.json



 proxy.config.json




 tsconfig.json




 README.md



 .editorconfig



 karma.conf.js



 src