

## Http Interceptors

**Http Interceptor** to klasa (od Angular 15 można także implementować je jako funkcje), która implementuje interfejs **HttpInterceptor**. Z jej poziomu możemy modyfikować request (żądanie do serwera) lub modyfikować / reagować na przychodzące response (odpowiedzi ze serwera). Przykłady wykorzystywania:

- Implementacja globalnego spinnera (ładowanie przy pracy z HTTP)
- Ustawianie nagłówków request'a (np. Authorization)
- Obsługa błędów dla całej aplikacji w jednym miejscu
- Notyfikacje (powiadomienia, że coś zostało załadowane itp.)

I wiele innych.

Aby dany Interceptor działał globalnie (dla całej aplikacji) musimy upewnić się, że importujemy **HttpClientModule** tylko raz w naszej aplikacji – w AppModule (lub CoreModule, który jest importowany przez AppModule).

# Http Interceptors wykorzystanie

```
@Injectable()
export class XInterceptor implements HttpInterceptor {
  no usages
  constructor() {}
  no usages
  intercept(
    request: HttpRequest<unknown>,
    next: HttpHandler
  ): Observable<HttpEvent<unknown>> {
    return next.handle(request).pipe(
      tap( observerOrNext: (event : HttpSentEvent | HttpHeadersResponse | HttpResponse<any> | HttpProgressEvent | HttpUserEvent<any> ) => {
        if (event instanceof HttpResponse) {
          console.log('Mam status:', event.status);
        }
      })
    );
  }
}
```

```
providers: [
  { provide: HTTP_INTERCEPTORS, useClass: XInterceptor, multi: true },
],
exports: [],
})
export class CoreModule {}
```

Dzięki użyciu operatorów RxJs mamy dostęp także do odpowiedzi serwera. Gotowy `HttpInterceptor` musimy dostarczyć w module, w którym mamy zaimportowany `HttpClientModule`.