

AuthInterceptor

August 18, 2026

AuthInterceptor

Kind: Service**Source:** Pams/Web/portal/pams-web/src/app/interceptors/auth.interceptor.ts (line 12)**Part of:** [Pams Web](#)

`AuthInterceptor` intercepts outgoing Angular `HttpClient` requests and adds authentication data before they are sent. Its `tokenGetter()` method retrieves the current token, while `intercept()` forwards the updated request through Angular's HTTP handler chain.

Methods

Method	Signature	Returns
<code>tokenGetter</code>	<code>tokenGetter()</code>	<code>unknown</code>
<code>intercept</code>	<code>intercept(request: HttpRequest<any>, next: <code>HttpHandler</code>)</code>	<code>Observable<HttpEvent<any>></code>

Diagram

```
sequenceDiagram
    participant App as Angular component/service
    participant Client as HttpClient
    participant Interceptor as AuthInterceptor
    participant Handler as HttpHandler
    participant API as API endpoint

    App->>Client: Send HTTP request
    Client->>Interceptor: intercept(request, next)
    Interceptor->>Interceptor: tokenGetter()
    Interceptor->>Handler: Forward request with auth header
    Handler->>API: Send request
    API-->>Handler: Return response
    Handler-->>Interceptor: Return HttpEvent
```

```
Interceptor-->>Client: Return Observable<HttpEvent>  
Client-->>App: Return response
```

Usage

```
import { HTTP_INTERCEPTORS, HttpClientModule } from '@angular/common/http';  
import { NgModule } from '@angular/core';  
import { AuthInterceptor } from './interceptors/auth.interceptor';  
  
@NgModule({  
  imports: [HttpClientModule],  
  providers: [  
    {  
      provide: HTTP_INTERCEPTORS,  
      useClass: AuthInterceptor,  
      multi: true,  
    },  
  ],  
})  
export class AppModule {}
```

AI Coding Instructions

- Register `AuthInterceptor` through Angular's `HTTP_INTERCEPTORS` provider with `multi: true`.
- Keep `intercept()` immutable: clone the request before adding or changing headers.
- Make `tokenGetter()` return the token source expected by the authentication flow, such as browser storage or an injected auth service.
- Handle missing tokens by forwarding the original request instead of adding an invalid authorization header.
- Preserve the `Observable<HttpEvent<any>>` returned by `next.handle()` so callers receive normal HTTP events.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `AppModule` — `Pams/Web/portal/pams-web/src/app/app.module.ts :1`

