

TokenRefreshInterceptor

August 17, 2026

TokenRefreshInterceptor

Kind: Class

Source: Frontend/src/app/InterceptorTokenRefresh.ts (line 13)

Part of: Frontend

`TokenRefreshInterceptor` handles token refresh behavior within the Angular HTTP request pipeline. Its `intercept()` method receives HTTP events, coordinates refresh handling when authentication state requires it, and returns an `Observable<HttpEvent<any>>` to continue the request flow.

Implements: `HttpInterceptor`

Methods

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

Properties

Property	Type
<code>countRequest</code>	<code>number</code>

Where it refuses work

- `TokenRefreshInterceptor` stops the work with an early return when `event.body instanceof Blob || event.body instanceof ArrayBuffer`, in 3 places.
- `TokenRefreshInterceptor` stops the work with an early return when `!this.networkService.isOnline()`.

- `TokenRefreshInterceptor` stops the work with an early return when `error.status === 401 && !this.isRefreshTokenRequest(request)` .
- `TokenRefreshInterceptor` stops the work with an early return when `body === null || body === undefined` .
- `TokenRefreshInterceptor` stops the work with an early return when `Array.isArray(body)` .
- `TokenRefreshInterceptor` stops the work with an early return when `typeof body === 'string' && this.isoDateRegex.test(body)` .

Diagram

```
graph LR
  A[Application Request] --> B[HttpClient]
  B --> C[TokenRefreshInterceptor]
  C --> D{Token requires refresh?}
  D -->|No| E[Send Request]
  D -->|Yes| F[Refresh Token]
  F --> E
  E --> G[HTTP Response]
  G --> C
  C --> H[Application]
```

Usage

```
import { HTTP_INTERCEPTORS } from '@angular/common/http';
import { TokenRefreshInterceptor } from './InterceptorTokenRefresh';

@NgModule({
  providers: [
    {
      provide: HTTP_INTERCEPTORS,
      useClass: TokenRefreshInterceptor,
      multi: true,
    },
  ],
})
export class AppModule {}
```

AI Coding Instructions

- Keep `intercept()` compatible with the Angular `HttpInterceptor` contract and return an `Observable<HttpEvent<any>>` .

- Do not subscribe inside the interceptor; compose refresh and request handling with RxJS operators.
- Register this class through the `HTTP_INTERCEPTORS` provider with `multi: true`.
- Prevent token refresh requests from being intercepted again, or they may trigger a refresh loop.
- Preserve HTTP errors when refresh handling fails so calling code can handle authentication failures.

Relationships

- `IMPORTS` → `AuthService`
- `IMPORTS` → `SppinerService`
- `IMPORTS` → `AppSettings`
- `IMPORTS` → `NetworkService`

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)

- `TranslateHandler` — `Frontend/src/app/app.module.ts` :1