

AuthService

August 17, 2026

AuthService

Kind: Class

Source: Frontend/src/app/services/security/auth.service.ts (line 131)

Part of: Frontend

`AuthService` manages authentication-related state and navigation checks in the frontend. It exposes methods for login, token access, route detection, branch switching, user information persistence, and activation decisions.

Methods

Method	Signature	Returns
<code>isLoggedInIn</code>	<code>isLoggedInIn()</code>	<code>boolean</code>
<code>getToken</code>	<code>getToken()</code>	<code>string</code>
<code>isLoginPage</code>	<code>isLoginPage()</code>	<code>boolean</code>
<code>isSheetPage</code>	<code>isSheetPage()</code>	<code>boolean</code>
<code>routeToLoginPage</code>	<code>routeToLoginPage()</code>	<code>void</code>
<code>canActivate</code>	<code>canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot)</code>	<code>boolean</code>
<code>switchBranch</code>	<code>switchBranch(branchId: number)</code>	<code>string</code>
<code>login</code>	<code>login(usernameInput: string, passwordInput: string, rememberMe: boolean, cookiesCode: string, locationInfo: LocationInfo)</code>	<code>void</code>
<code>saveUserLoggedInInformation</code>	<code>saveUserLoggedInInformation()</code>	<code>void</code>
<code>verify</code>	<code>verify(verifyRequest: VerifyRequest)</code>	<code>void</code>

Method	Signature	Returns
resendCodeToVerfy	resendCodeToVerfy(TempToken: any)	void
countOfActiveLoginDevicesByUser	countOfActiveLoginDevicesByUser()	void
verifiedOTPLogin	verifiedOTPLogin(username: undefined, password: undefined, id: undefined, grantCode: undefined, grantIP: undefined, isTwoFactorVerified: undefined)	void
trustedMachine	trustedMachine(grantCode: string)	void
useJwtHelper	useJwtHelper(givenToken: any)	void
checkCanActivate	checkCanActivate(url: string)	boolean
getSubscriptionPackages	getSubscriptionPackages()	void
getSubscriptionFeatues	getSubscriptionFeatues()	void
HasPermission	HasPermission(data: string, ID: number, SecurityAccessTypeID: number)	any
GetCompany	GetCompany(domain: string)	void
HasPermission2	HasPermission2(PersonUserID: number, SecurityAccessTypeID: number, PageID: number)	any
userPermissionsForNewList	userPermissionsForNewList()	void
getPermissionsForSheet	getPermissionsForSheet(moduleId: number, objId: number)	any
setListOfNewPermission	setListOfNewPermission(value: ListOfNewResponse)	void
setExportLogs	setExportLogs(moduleId: number)	void
checkErrorResponse	checkErrorResponse(Response: undefined, module: string)	string
getNotificationCount	getNotificationCount()	Observable<any>
clearStoredData	clearStoredData()	void
logOut	logOut(companyId: number, userId: number)	void
getRefreshToken	getRefreshToken()	string

Method	Signature	Returns
storeTokens	storeTokens(accessToken: string, refreshToken: string)	void
isTokenExpired	isTokenExpired()	boolean
refreshAccessToken	refreshAccessToken(refreshToken: string)	Observable<any>
isLoggedInWithValidToken	isLoggedInWithValidToken()	boolean
initializeTokenRefresh	initializeTokenRefresh()	void
getTokenExpirationInfo	getTokenExpirationInfo()	{ expiresAt: Date, timeUntilExpiry: number, shouldRefresh: boolean }
startTokenRefreshTimer	startTokenRefreshTimer()	void
stopTokenRefreshTimer	stopTokenRefreshTimer()	void
shouldRefreshToken	shouldRefreshToken()	boolean
getTimeUntilExpiry	getTimeUntilExpiry()	number

Properties

Property	Type
localStorageRememberMe	string
refreshMinutesTimeBeforeExpiration	number
jwtHelper	any
userPermissions	UserPermissions[]
loggedInUser	Person
_listOfNewPermission	ListOfNewResponse
listOfNewPermission	any
SubscriptionPackages	number[]

Where it refuses work

- AuthService stops the work with an early return when !token , in 6 places.

- `AuthService` stops the work with an early return when `!expirationDate` , in 2 places.
- `AuthService` stops the work with an early return when `item === undefined || item === null` .
- `AuthService` stops the work with an early return when `window.location.href.includes("ogin") || window.location.href.includes("ignin") || window...` .
- `AuthService` stops the work with an early return when `window.location.href.toLocaleLowerCase().includes("edit") || window.location.href.toLocal...` .
- `AuthService` stops the work with an early return when `url === 'home' || url === ''` .

When something fails

- `AuthService` handles failure in 5 places: it turns it into a return value in 4, and logs it and continues in 1.

Diagram

```
graph LR
  App[Frontend Component or Route Guard] --> AuthService[AuthService]
  AuthService --> Login[login]
  AuthService --> Verify[verify]
  AuthService --> Token[getToken]
  AuthService --> Status[isLoggedIn]
  AuthService --> Routes[isLoginPage / isSheetPage]
  AuthService --> Guard[canActivate]
  AuthService --> Navigation[routeToLoginPage]
  AuthService --> Branch[switchBranch]
  AuthService --> UserInfo[saveUserLoggedInInformation]
```

Usage

```
import { Component } from '@angular/core';
import { AuthService } from '../services/security/auth.service';

@Component({
  selector: 'app-account',
  template: `
    <button *ngIf="!isLoggedIn" (click)="login()">Log in</button>
    <button *ngIf="isLoggedIn" (click)="switchBranch()">Switch branch</button>
  `,
})
export class AccountComponent {
```

```

constructor(private readonly authService: AuthService) {}

get isLoggedIn(): boolean {
  return this.authService.isLoggedIn();
}

login(): void {
  this.authService.login();
}

switchBranch(): void {
  const branch = this.authService.switchBranch();
  console.log('Selected branch:', branch);
}
}

```

AI Coding Instructions

- Keep authentication checks in `AuthService` rather than duplicating login or token logic in components.
- Call `canActivate()` from route guard integration points and redirect with `routeToLoginPage()` when access is denied.
- Check `isLoginPage()` and `isSheetPage()` before applying page-specific authentication behavior.
- Read tokens through `getToken()` instead of accessing authentication state directly.
- Keep `saveUserLoggedInInformation()` and `verify()` aligned with the login flow.

Used by

218 references from 218 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (218)

- `AppRoutingModule` — `Frontend/src/app/app-routing.module.ts :1`
- `AppComponent` — `Frontend/src/app/app.component.ts :1`
- `MrqDetailsComponent` — `Frontend/src/app/components/MRQ/mrq-details/mrq-details.component.ts :1`
- `MrqInquiryComponent` — `Frontend/src/app/components/MRQ/mrq-details/mrq-inquiry/mrq-inquiry.component.ts :1`
- `MrqListComponent` — `Frontend/src/app/components/MRQ/mrq-list/mrq-list.component.ts :1`

- `BillOfMaterialsDetailsComponent` — `Frontend/src/app/components/bill-of-materials/bill-of-materials-details/bill-of-materials-details.component.ts :1`
- `BillOfMaterialsListComponent` — `Frontend/src/app/components/bill-of-materials/bill-of-materials-list/bill-of-materials-list.component.ts :1`
- `BlogPopupComponent` — `Frontend/src/app/components/blogs/blog-popup/blog-popup.component.ts :1`

...and 210 more.