

AuthService

August 17, 2026

AuthService

Kind: Class

Source: Pams/Web/Pams.WebApp/Client/app/services/security/auth.service.ts (line 35)

Part of: Pams

`AuthService` manages client-side authentication state, JWT helper setup, branch switching, company lookup, and permission checks in the Pams web application. Route-related code can call `canActivate()` and `verifyLogin()` to determine whether a user session is valid.

Implements: `CanActivate`

Methods

Method	Signature	Returns
<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)</code>	<code>void</code>
<code>useJwtHelper</code>	<code>useJwtHelper()</code>	<code>void</code>
<code>verifyLogin</code>	<code>verifyLogin(url: string)</code>	<code>boolean</code>
<code>HasPermission</code>	<code>HasPermission(data: string, ID: number, SecurityAccessTypeID: number)</code>	<code>any</code>
<code>GetCompany</code>	<code>GetCompany(domain: string)</code>	<code>void</code>
<code>HasPermission2</code>	<code>HasPermission2(PersonUserID: number, SecurityAccessTypeID: number, PageID: number)</code>	<code>any</code>
<code>logout</code>	<code>logout()</code>	<code>void</code>

Properties

Property	Type
userPermissions	UserPermissions[]
jwtHelper	JwtHelper

Where it refuses work

- AuthService stops the work with an early return when `item == undefined || item == null`.
- AuthService stops the work with an early return when `branchId == 0`.
- AuthService stops the work with an early return when `url == "" && this.LoggedIn`.

Diagram

```
graph LR
    UI[Application UI] --> AuthService[AuthService]
    AuthService --> Login[login]
    AuthService --> Jwt[useJwtHelper]
    AuthService --> Verify[verifyLogin]
    AuthService --> Guard[canActivate]
    AuthService --> Branch[switchBranch]
    AuthService --> Company[GetCompany]
    AuthService --> Permissions[HasPermission / HasPermission2]
    AuthService --> Logout[logout]
    Guard --> Routes[Protected Routes]
```

Usage

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

@Component({
  selector: "app-account-menu",
  template: `
    <button *ngIf="isLoggedIn()" (click)="changeBranch()">
      Switch branch
    </button>

    <button (click)="signOut()">Log out</button>
  `
})
export class AccountMenuComponent {
```

```

constructor(private readonly authService: AuthService) {}

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

canAccessProtectedRoute(): boolean {
    return this.authService.canActivate();
}

changeBranch(): string {
    return this.authService.switchBranch();
}

signOut(): void {
    this.authService.logout();
}
}

```

AI Coding Instructions

- Keep authentication and authorization checks routed through `AuthService` rather than duplicating session logic in components.
- Preserve the existing method names and casing, including `HasPermission`, `HasPermission2`, and `GetCompany`.
- Call `verifyLogin()` or `canActivate()` before rendering or navigating to authenticated areas.
- Keep JWT-related changes aligned with `useJwtHelper()`, `login()`, and `logout()` so session state is cleared consistently.

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `MaritalStatusService` — `Pams/Web/Pams.WebApp/Client/app/services/common/marital-status.service.ts :1`
- `SharedServicesModule` — `Pams/Web/Pams.WebApp/Client/app/services/shared-service.module.ts :1`