

Global function available through all app Angular2

Asked 3 years, 1 month ago Active 2 years, 4 months ago Viewed 9k times



3



I have a PermissionService, which provide user roles. At the server-side data will not be uploaded if the user is not corresponds on role. The back-end is written with asp.net web api, which will use attributes to secure data. On upload page will be static upload user roles, the idea is just to show or hide elements on page which depending from user role. The PermissionsService check available role in its array. There are methods like isSeller(), isManager(). And all what i want is to provide accessibility from each view. For now i have this implementation.



permission.service

```
import { Injectable } from "@angular/core";

export enum Roles {
  Admin,
  Manager,
  Moderator,
  Seller
}

interface IPermissionDictionary {
  [key: string]: boolean;
}

@Injectable()
export class PermissionService {
  private permissions: IPermissionDictionary = {};

  public constructor() {
    this.emitPermissions();
  }

  private emitPermissions(): void {
    let selector = document.querySelectorAll("#roles > span");
    let availableRoles = Array.from(selector).map(element => element.textContent);

    for (let role in Roles) {
      if (!/^\\d+$/ .test(role)) { // for strings types in Roles
        this.permissions[role] = availableRoles.indexOf(role) > -1;
      }
    }
  }
}
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```
public isInRole(role: string): boolean {
    return this.permissions[role];
}

public isAdmin() {
    return this.isInRole(Roles[Roles.Admin]);
}

public isSeller() {
    return this.isInRole(Roles[Roles.Seller]);
}

public isManager() {
    return this.isInRole(Roles[Roles.Manager]);
}

public isModerator() {
    return this.isInRole(Roles[Roles.Moderator]);
}
}
```

app.component

```
import { Component } from "@angular/core";
import { ROUTER_DIRECTIVES } from "@angular/router";

import { PermissionService } from "../share/permission.service";
import { HomeComponent } from "../home/home.component";
import { OrderComponent } from "../order/order.component";

@Component({
    selector: "admin-panel",
    templateUrl: "../app/app.template.html",
    directives: [ROUTER_DIRECTIVES],
    precompile: [HomeComponent, OrderComponent]
})

export class AppComponent {
    constructor(private permissionService: PermissionService) {

    }
}
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```
import { bootstrap } from "@angular/platform-browser-dynamic";
import { AppComponent } from "./app.component";
import { APP_ROUTES_PROVIDER } from "./app.routes";
import { HTTP_PROVIDERS } from '@angular/http';
import { PermissionService } from "../share/permission.service";
bootstrap(AppComponent, [APP_ROUTES_PROVIDER, HTTP_PROVIDERS, PermissionService]);
```

For now to access the method of PermissionService need to inject it in component constructor. And in template is use like

```
<div *ngIf="permissionService.isAdmin()">will show if you are admin</div>
```

But every time to inject my service in each component where i want to use it seems for me strange. And i just want to get access it from every part of my app like:

```
<div *ngIf="isAdmin()">will show if you are admin</div>
```

javascript



angular

typescript

asked Aug 11 '16 at 12:32



unsafePtr

629 1 8 22

does adding and isAdmin() { return permissionService.isAdmin() } to your AppComponent work? – Jackson Aug 11 '16 at 12:49

1 @Jackson yep, but what to do if i want to get it from for example product.component? to inject permissionService again? – unsafePtr Aug 11 '16 at 13:19

could you find a solution to this ? – Arash Nov 14 '17 at 12:04

2 Answers



I think the person who asked this question has another version of Angular2 (perhaps a pre-release?), but in the latest version if you need to export a service for all the app you do it in the following way. First, in your main.ts you must have a class that you bootstrap. like

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



```
platformBrowserDynamic().bootstrapModule(AppModule);
```

In this class "AppModule" (or whatever it is in your case), you should be able to add a global service provider in this way:

```
...
import {GlobalService} from './global-service.service'
@NgModule({
  ...
  providers: [MyGlobalService],
  ...
})
export class AppModule{ ...}
```

In this way `MyGlobalService` is available for all other components.

Hopefully this will be useful to someone :).

answered Dec 8 '16 at 0:05



ACBM

430 8 22



0



Some option could be to create top level super class with the permission methods and then just subclass in view .ts. Not sure if this suits you as you still need to import super class into your components and extend it. It can also violate the "is-a".

answered Jun 9 '17 at 10:49



tomasz

1 1