How to restrict access to Login Route if user is already logged in in Angular 4?

0

2

I have successfully implement AuthGuardService which restrict access to a protected route if user is not logged in.

What I am trying to achieve is, if user is already logged in and accessed the Login Route, I want it to redirect to another route like homepage.

angular angular4-router

asked Nov 2 '17 at 14:22

Daniel Chikaka
398 2 7 21

1 I would make another canActivate guard for the login route which redirects to the homepage if the user is already logged in. – LLai Nov 2 '17 at 14:26

Would you mind sharing some sample code? @LLai – Daniel Chikaka Nov 2 '17 at 15:36

It's essentially what Boulboulouboule posted. if they are logged in, navigate to home page (& return false) else return true so they navigate to login page. Then attach this guard to the login route in your route definitions – LLai Nov 2 '17 at 15:41

I tried, it does't work. Blank page is shown instead @LLai – Daniel Chikaka Nov 2 '17 at 16:00

Is the guard redirecting correctly? So you are navigated to `/home` ? (or whatever your home route is) – LLai Nov 2 '17 at 16:02

2 Answers

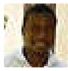
3

You could perform a simple check in the ngOnInit of the login component like so and redirect to another page of your choice if they are authenticated already:

```
ngOnInit() {
```

That worked for me!

answered Nov 3 '17 at 8:06



Daniel Chikaka
398 2 7 21

I did the same but the problem is login component loads for like a second or so. So it shows the login html for the same time then it redirects me to intended route. – prit.patel Aug 20 '18 at 9:24

Make it log the user out when the login component is loaded. Only load the login component from a route guard (implementing CanActivate), with a return url in the query parameters. – Dico Apr 3 at 4:25

2

You can use the **CanActivate guard** on paths that requires user to be logged in:

```
const ROUTER: Routes = [
  {path: 'restricted', component: MyRestrictedCOmponent, canActivate:[LoginActivate]},
  {path: 'home', component: HomeComponent},
];
```

And the guard that redirect unlogged users on the homepage :

```
@Injectable()
export class LoginActivate implements CanActivate {
  constructor(private authService: AuthService, private router: Router) {}
  canActivate(
    route: ActivatedRouteSnapshot,
    state: RouterStateSnapshot
  ): Observable<boolean>|Promise<boolean>|boolean {
    if (!authService.isLoggedIn()) {
      this.router.navigate(['home']);
    }
    return true;
  }
}
```

It doesn't work. All i get is a blank page for "home" route when the user is logged in – Daniel Chikaka Nov 2 '17 at 15:34

Did you have some errors ? Here's the official documentation about guards, take a look to understand how to implement it in your project : angular.io/guide/router#guards – Boulboulouboule Nov 3 '17 at 6:56

1 No errors, it just refuse to work. Never mind, i have found another way to do it will post as answer to help others with similar problem @Boulboulouboule – Daniel Chikaka Nov 3 '17 at 8:02

1 Don't forget to register your guard as a provider in your app configuration, i.e.: @NgModule({ declarations: [ ], imports: [ ], providers: [LoginActivate] }) – Lucaci Andrei Jan 23 '18 at 14:03