Working with canActivate and canDeactivate

CanActivate – Interface that a class can implement to be a guard deciding if a route can be activated.

Structure of canActivate

interface [CanActivate](https://angular.io/api/router/CanActivate) {

[canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot): Observable<boolean>|Promise<boolean>|boolean](https://angular.io/api/router/CanActivate#canActivate)

}

Takes two params 1) route and 2) state

We need to create a service which extends canActivate and override the canActivate Method

Ex:

**Service**

**import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot } from "@angular/router";**

**import { Observable } from "rxjs/Observable";**

**export class AuthGuardService implements CanActivate {**

**canActivate(route : ActivatedRouteSnapshot , state : RouterStateSnapshot) : boolean | Observable<boolean>**

**{**

**let uname = prompt('Enter UserName : ');**

**if(uname == 'admin' )**

**return true ;**

**else**

**return false**

**}**

**}**

**At AppModule.ts**

providers: [

AuthGuardService,

]

const myRoutes : Routes = [

{

path:'viewbooks',

component:ViewBooksComponent,

canActivate:[AuthGuardService]

},

{

path:'addbook',

component:AddBookComponent,

canDeactivate :[DeactivateGuardService]

}

]

So Guards will be applied for the specified Routes