# Protecting Rotes

## Create a new service call “auth-gard”

ng g s auth-gard

## Add providers

Added AuthGuradService into providers list on app.module.ts file

providers: [AuthGuardService],

## implement CanActivate class

import { CanActivate } from ‘@angular/router’;

export class AuthGuardService implements CanActivate{}

## Rewrite constructor

constructor(private afAuth:AngularFireAuth, private router: Router) { }

## Rewrite canActivate function

Using map method instead of subscribe method because canActivate has return type of Observable<boolean>

// reutrn true when user is login, otherwise return false

 canActivate(){

  return this.afAuth.authState.pipe(map(

    user=>{

      if(user) return true;

//if user is null, then redirected to login in page

this.router.navigate([‘/login’]);

      return false;

    }));

  }

## Add canActiveate

Add authGuardService to check-out page by adding canActivate parameter

//using canActivate to protect none sigin user access check-out pager

 {path:'check-out', component:CheckOutComponent, canActivate:[AuthGuardService]}

# Redirected after login success

## Import ActivatedRouteSnapshot and RouterStateSnapshot for auth-guard.service.ts

import { ActivatedRouteSnapshot, RouterStateSnapshot } from '@angular/router';

## Rewrite canActivate function with route and state as parameters

canActivate(route:ActivatedRouteSnapshot,state:RouterStateSnapshot)

## Show the return URL

{ queryParams : {returnUrl : state.rul}}

//if user is null, then redirected to login in page and show the state url

      this.router.navigate(

        ['/login'],

        { queryParams : {returnUrl: state.url}}

        );

Result : 

## Store returnUrl into local storage

Get returnUrl from auth-guard service, if returnUrl does not exist, set returnUrl to home page.

let returnUrl = this.route.snapshot.queryParamMap.get('returnUrl') || '/';

Store returnUrl into local storage as returnUrl.

localStorage.setItem('returnUrl',returnUrl);

## Restore returnUrl and call navigate method after success login

this.afAuth.signInWithPopup(new firebase.auth.GoogleAuthProvider())

    //after login, redirected to the returnUrl

    .then(()=>{

      let returnUrl = localStorage.getItem('returnUrl');

      //checked if returnUrl is null

      if(returnUrl){

        this.router.navigate([returnUrl]);

      }

    });