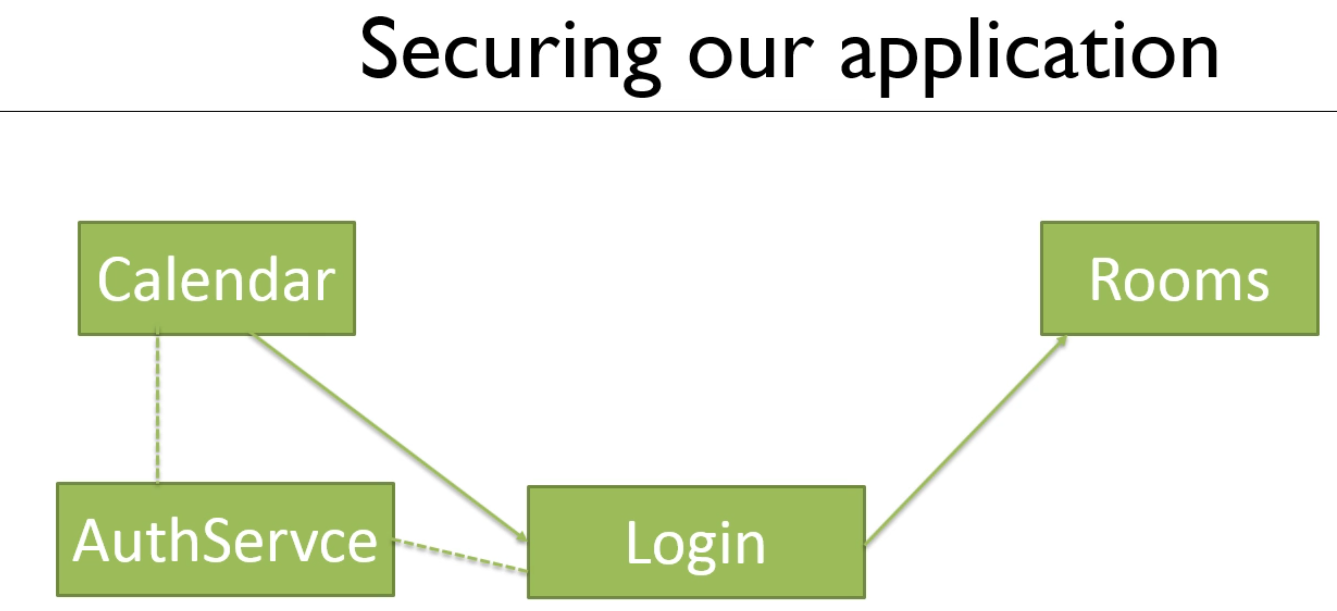
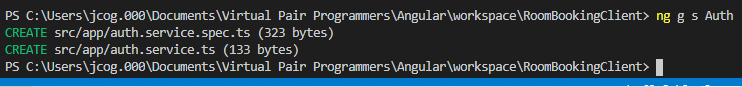
**An overview of the login functionality:**

Create an AuthenticationService and a Login component

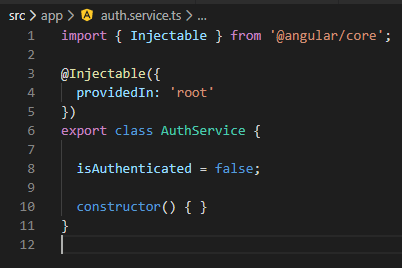


Create the AuthService

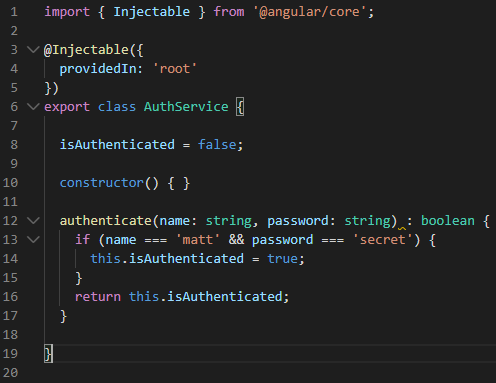
>ng g s Auth



Added property Boolean isAuthenticated

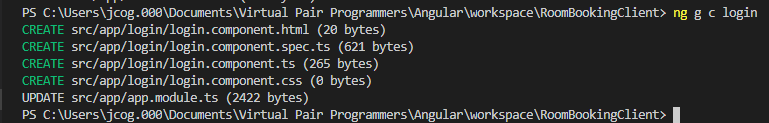


Create method authenticate()

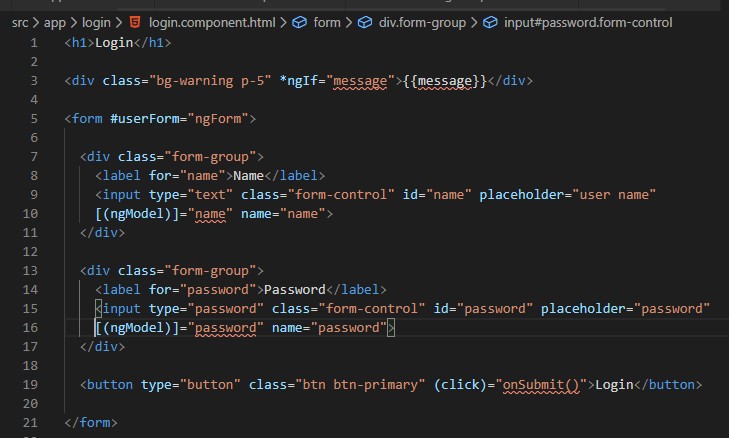


Now, create the login component

>ng g c login



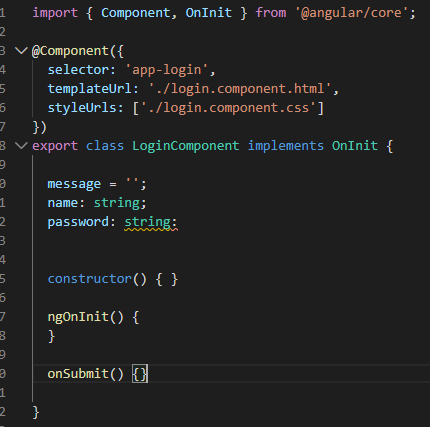
Copy the starting html from the provided component



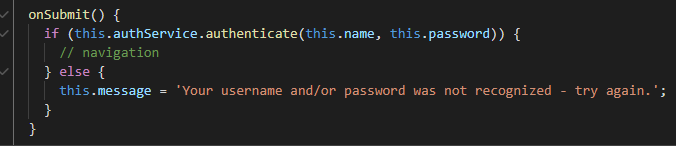
Add the message variable to the login-component.ts class file



Create variables name, password and onSubmit() method

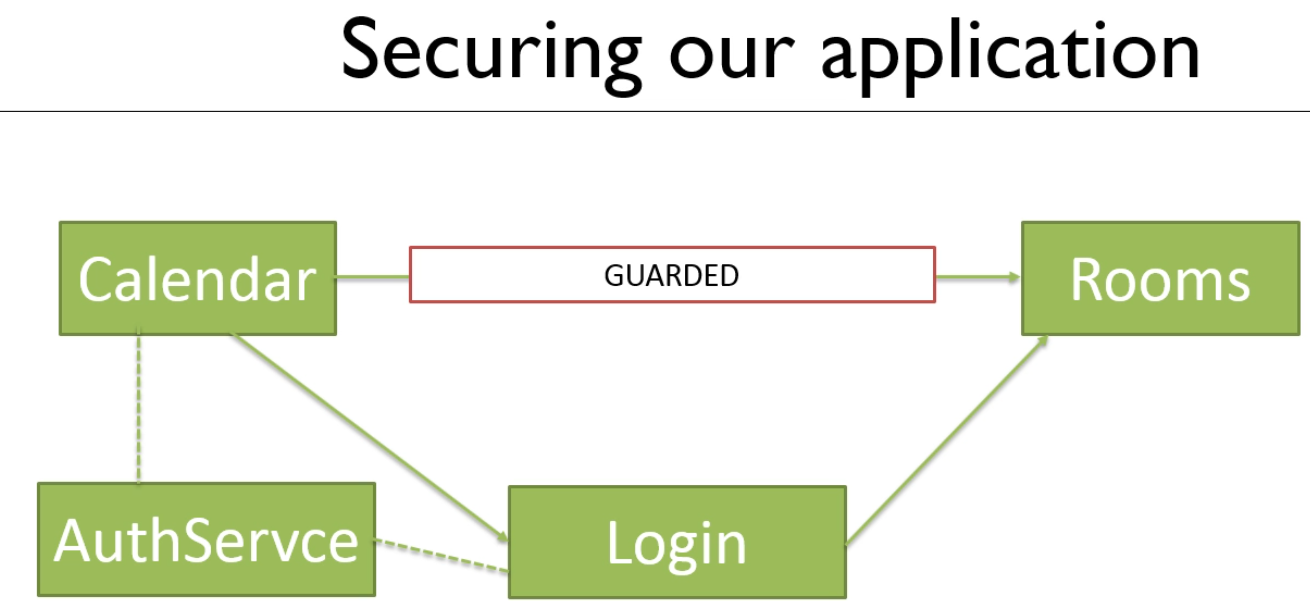


Now, fill out the onSubmit() method

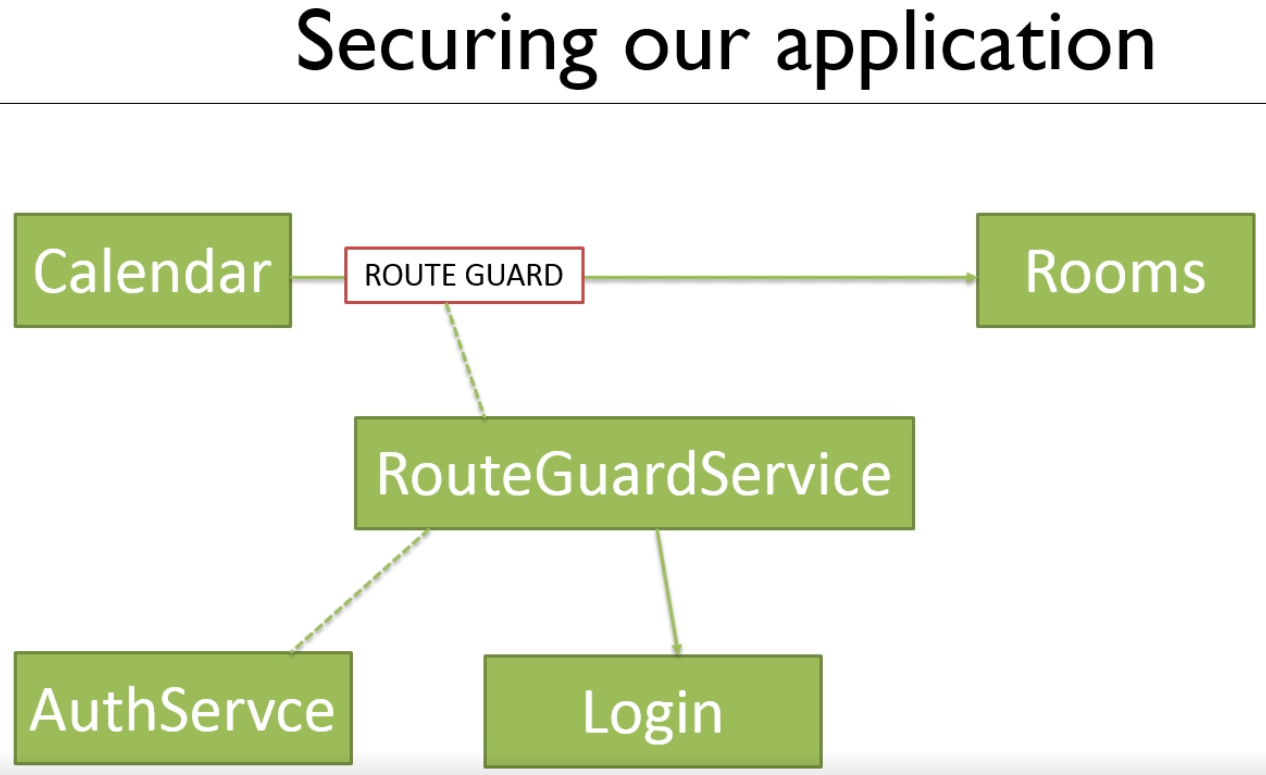


Now, let us create the redirection piece

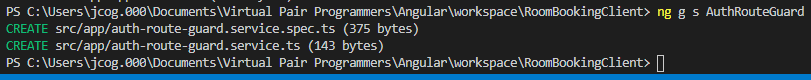
This feature is called a rootGuard, a guarded route is created.



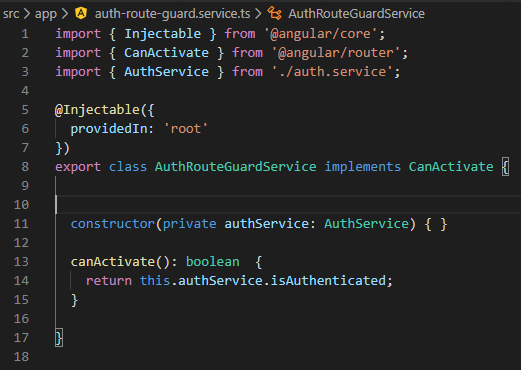
A rootGuard is a service that implements a specific interface



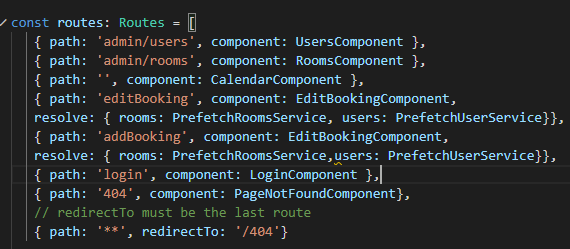
Let us code up the RouteGuardService



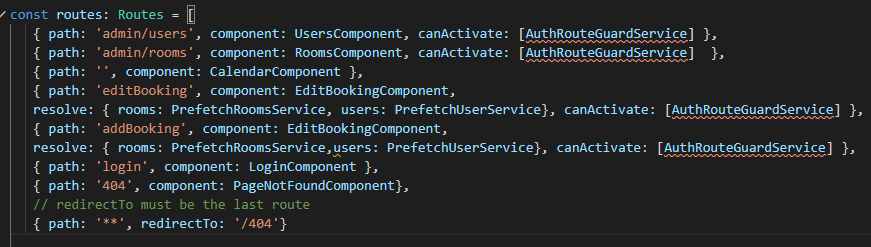
Implement the service



Update the app-module.ts file with a new route for the login component



Add the protected urls with routeGuard



If you try to access these four urls they will not work, the user is not authenticated

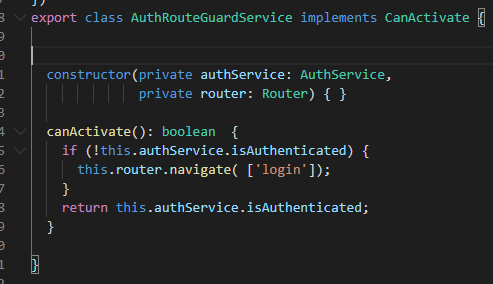
Let redirect to the login component

Code this up

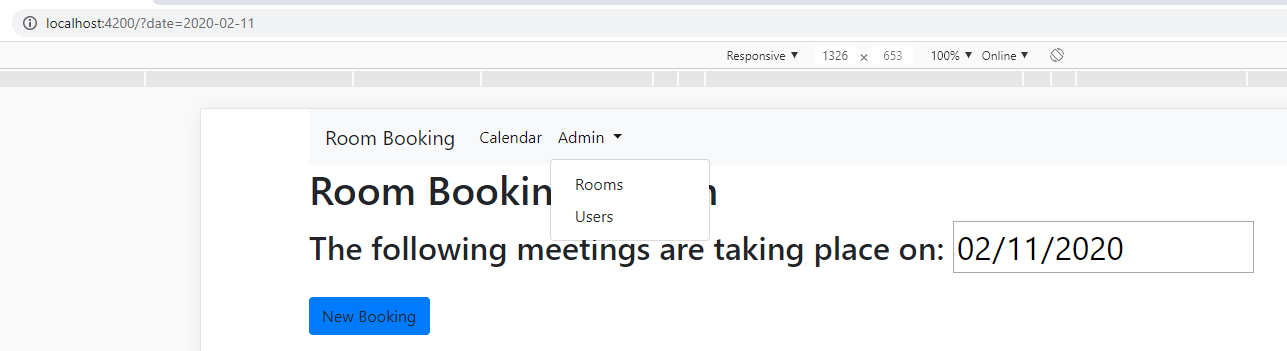
Add the imports first



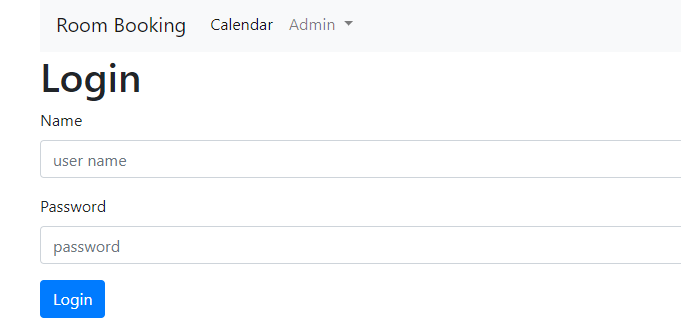
Now code the AuthRouteGuardService



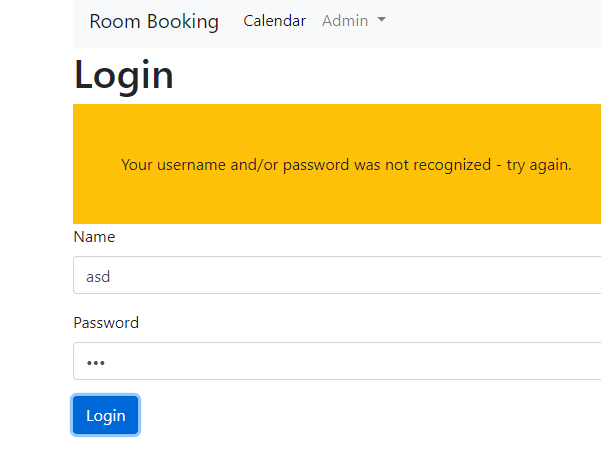
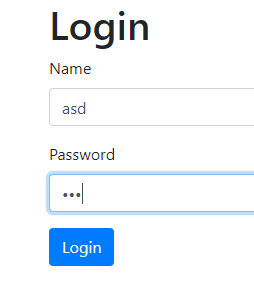
Now we can test this

If I access Rooms or Users I should be redirected to the login component

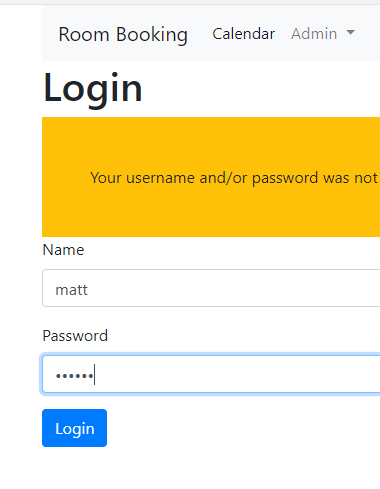
And it does



Put in the wrong username and/or password

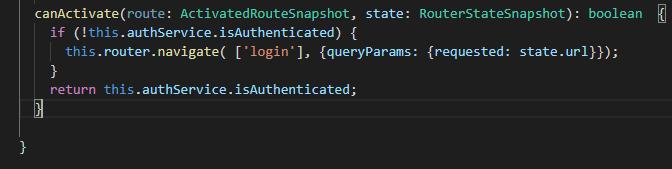


Put in the correct information: matt and secret

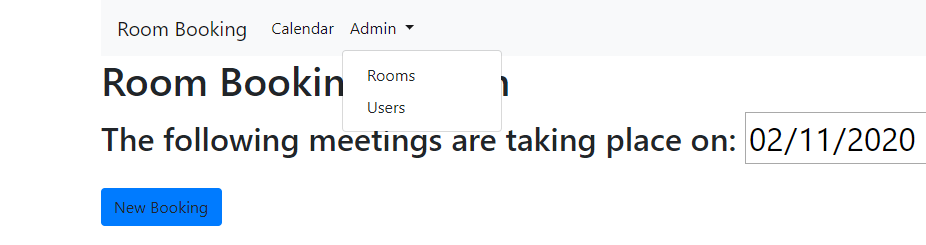


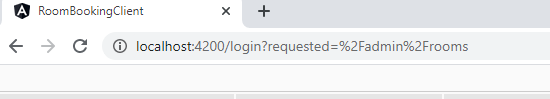
Nothing happens

So, let us capture where the user wanted to go with snapshots

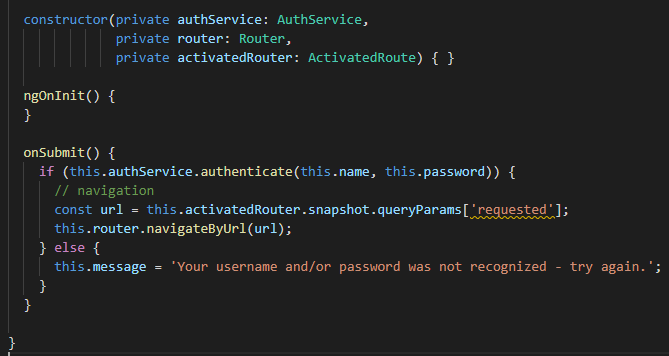


Now let us see what happens to the browser url with the new navigation

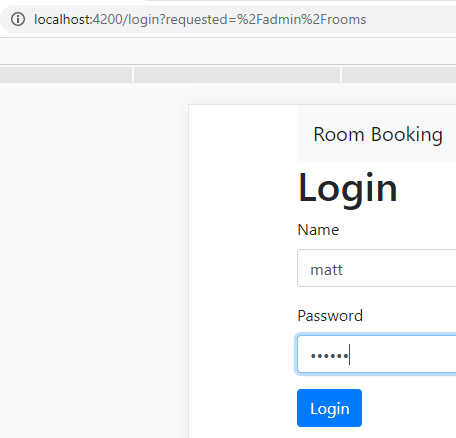




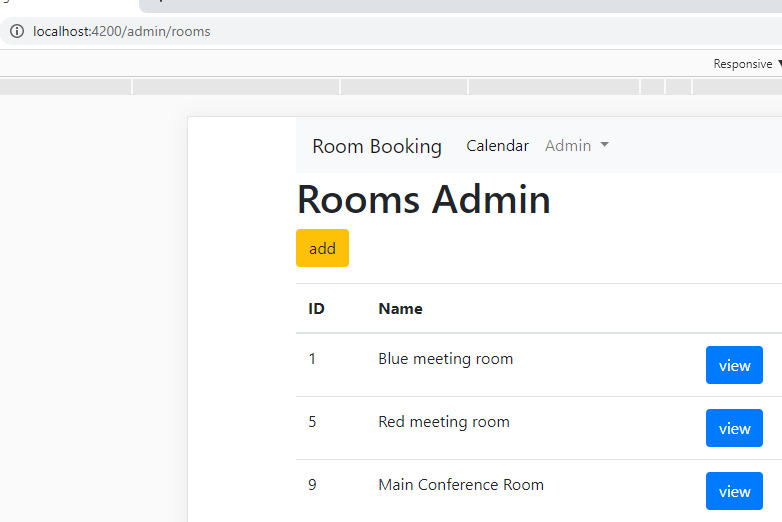
The route path admin/rooms has been captured



Test if the login page can route us to the correct page on successful authentication



Looking at the URL we should be sent to admin/rooms



And that is exactly what happens