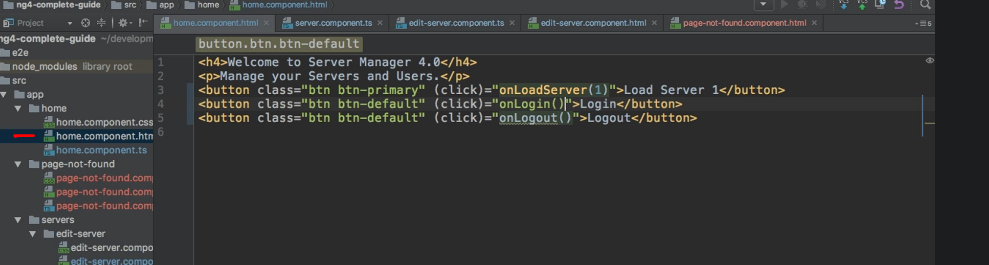
**148. Using a Fake Auth Service:**

* -: Now let's finish this canActivate behavior by allowing the user to log in.
* I'll quickly do this by going to my home component, and there I want to add two new buttons, one log in button and one log out button.
* And I will add some CSS classes, button default maybe, and button default here too.
* And here I want to execute on logout.



* And for the other one, of course, I will execute on log in.
* And I will keep this really simple since it's not about having a nice UI here.
* I will simply inject my off service here, of type off service of course.
* Make sure to add the import at the top and then I'll add my two methods here.
* So on log in, we'll simply reach out to the off service and call the log in method to set this logged in Boolean to true.

Text

Description automatically generated

* And on log out, we'll of course do the same, but we'll call the log out method to set it to false.
* And this simple improvement now makes sure that we can, of course, log in.
* We don't see any visual indication here but we do see it if we try to access a single server.

Graphical user interface, text, application

Description automatically generated

* Now this is working, always after these 800 milliseconds which it takes to evaluate wherever we are allowed to access, but if I switch this back to log out again, you see we're redirected again.

Graphical user interface, text

Description automatically generated

* So now we can control this.
* Now we finish this fake off service and more importantly, the canActivate guards.