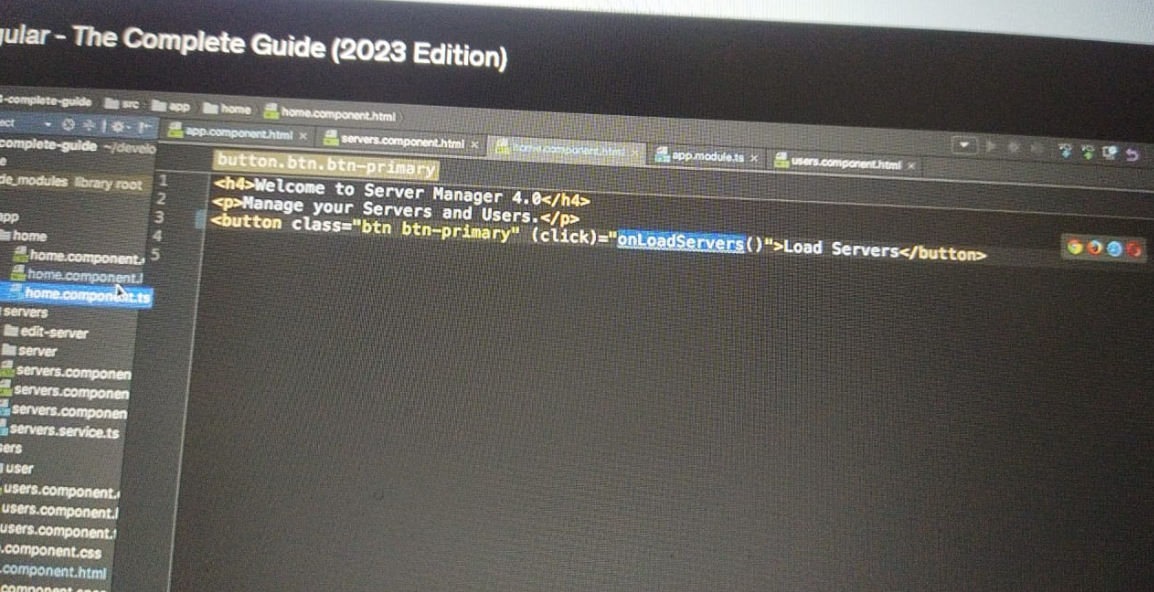
**130. Navigating Programmatically**

* -: Whoa! We already covered quite some things.
* We learned how to add routes and how to load them either by typing it manually into our navigation bar here or by clicking some links.
* We also learned how to stall these links.
* But, what now, if we wanted to load a route programmatically? So we don't have a link, the user can click, but we finished some operation, or the user clicked some button, and then we want to trigger the navigation from our types good code.
* We can do this.



* Let's say, in our home component here, I add a new button to it.
* And, on this button, I simply want to load the server, the service page.
* So here, now we could try adding router link, but let's say I want to have a click listener and execute some method because we do something else than just navigating there.
* I will also add some CSS classes to make this button look nice.
* So, on this Click Listener, I want to execute onLoadServers.
* And here this method now gets implemented in the home component and onLoadServers.
* I now still, again, I simply want to navigate to the service component.
* So we could, for this example, of course, always the user router link, but let's say here we have some complex calculation, or we reach out to our back, and we store something on the server.
* And, once we are done, now we want to navigate away.
* **Injecting the router:**
* To do so, we somehow need to get access to our router, this angular router, because we need to tell it, "Hey, please navigate somewhere else.
* " The nice thing is we can check this router.
* So let's bind it to a private property.
* Maybe we name it router.
* And this will be of type router.

Text

Description automatically generated

* Router needs to be imported from @angular/router.
* With this injected, we can use this router here.
* So this router, and then we get a couple of methods there.
* One of the most important ones being Navigate.
* Now Navigate takes an argument which allows us to, guess what, navigate to a new route.
* And here a route is to find as an array of the single or the different elements of this new path.
* So, just like here, an app component, where I always explained this alternative way of binding, routing, binding it to an array where you could have users for the user's part of your path, and then another element for maybe something for the something part of your path.
* Now, you always have to pass such an array when using the Navigate method.
* But, in here still, the first element is simply the first element or the first segment of your path.
* So if, let's say, we want to go to slash service here, we could add slash service here.
* And, as before, this is now an absolute path.
* You could have a relative one, but here you have to control to what this should be relative to.
* And I will come back to this.
* So for now let's make it an absolute path here.
* And, once we did this, let's check our page here, click this button, and we go to servers.
* So this is now programmatically routing to a different page.
* Still, it doesn't reload our page.
* It does the same as if we clicked a router link.
* But, with this router navigate method, we are able to trigger this programmatically, so trigger this in our code.