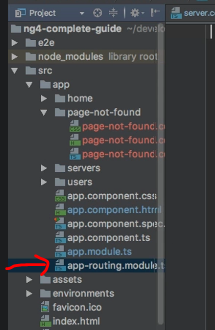
**144. Outsourcing the Route Configuration:**

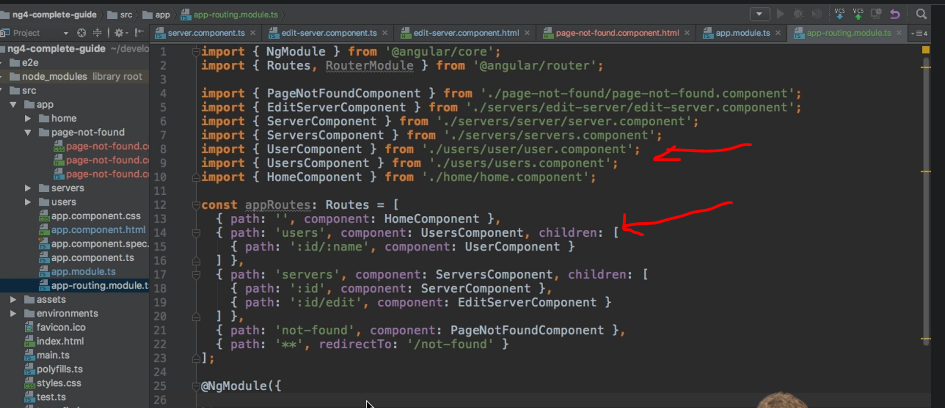
* -: Whoo! So a lot was covered.
* Let's dive even deeper.
* If we have a look at our application, our app module.
* You already see that the routing takes up some significant space in our app module.
* So typically, if you have more than, let's say, two or three routes, you don't add it directly in the app module.
* Instead you add a new file.

**app-routing module:**

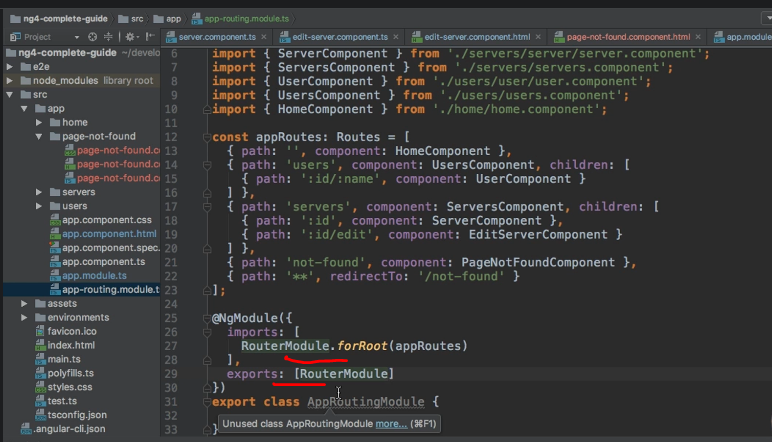
* Which is for the application as a whole, typically is called app-routing.
* module.
* So the app-routing.
* module.
* ts file, now will hold a second module.
* And I will have a whole module course section about Angular modules, later in the course.
* There, we will dive much deeper and learn how you can define and use multiple modules, use them together.
* This is a super easy use case here.
* So let's add it now.
* Here, I will give this a name of, AppRoutingModule, because that is what it is.



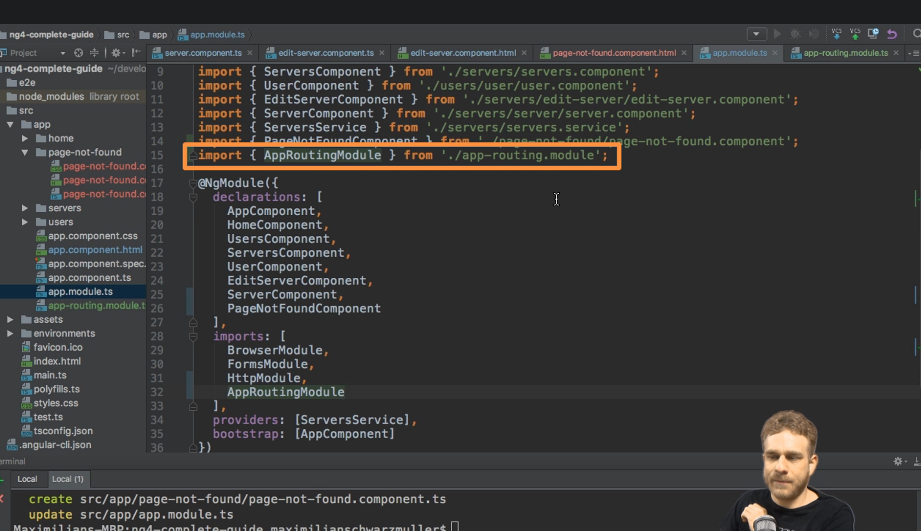
* And it receives this, @NgModulodule, decorator.
* So this is configured with a JavaScript object being passed to it.
* And this will be a super simple module.
* Now, as mentioned, I will dive deeper into this later.
* You can basically build your Angular app from multiple modules.
* This simple module here will handle all our routing related tasks.
* So I will cut all my routes.
* This app routes constant and add it in there.
* Of course, now, I need to make sure that all these imports here are added.
* So make sure to add all the imports to all, all these components.
* And also, of course, here to routes from the add angular router package.
* And you don't need to add declarations here, because these components already are declared in your app module.
* And we will soon add the app routing module to this route module.
* So no need to redeclare them.
* This would actually even give you an error.
* We also have to remove the router module from here, from our route module.
* Therefore, we can remove the imports from the router package.
* Make sure to import router module in your app routing module now.
* So to add this import here.
* And now we need to configure something in NgModule.
* We basically want to add imports, again.



* We had this in the app module too.
* And in imports, we want to use this router module.
* Use for route, as before, and pass our routes array to it.
* *So just like we did before, but that alone is not enough now, because, as I mentioned, we simply want to use this app routing module to outsource our routes.*
* *Therefore, we need to add our, AppRoutingModule, back to our main module.*
* ***Exports Array:***
* And for this, we need to add the exports array here.
* And I just can mention it again, we will dive deeper into modules later in the course.
* Exports simply tells Angular, Hey, from this module, if I were to add this module to the imports of another module, what should be accessible to this module, which imports this module? The one thing we want to make accessible is our RouterModule.
* Now here you don't call, forRoot, because you did this here.
* So in the first step, you configure this RouterModule, added your own routes to it.
* Now we simply export this configured RouterModule.



* And, therefore, in app module, we can now import our own AppRoutingModule.
* So here on the imports array, we can simply add, AppRoutingModule.
* Make sure to add the import path here.



* And with that, we have the same setup as before, but with a bit of a leaner app module and our routing functionality outsourced in this separate module.
* Which is kind of a good practice to do to keep your files organized and easy to read.
* So, same behavior as before.
* If we save this with ngserve still running, we should, therefore, see that everything still works.
* And this looks good to me.
* But again, important to understand, now with a different setup, a bit more readable and a good practice to implement if your application grows more complex.