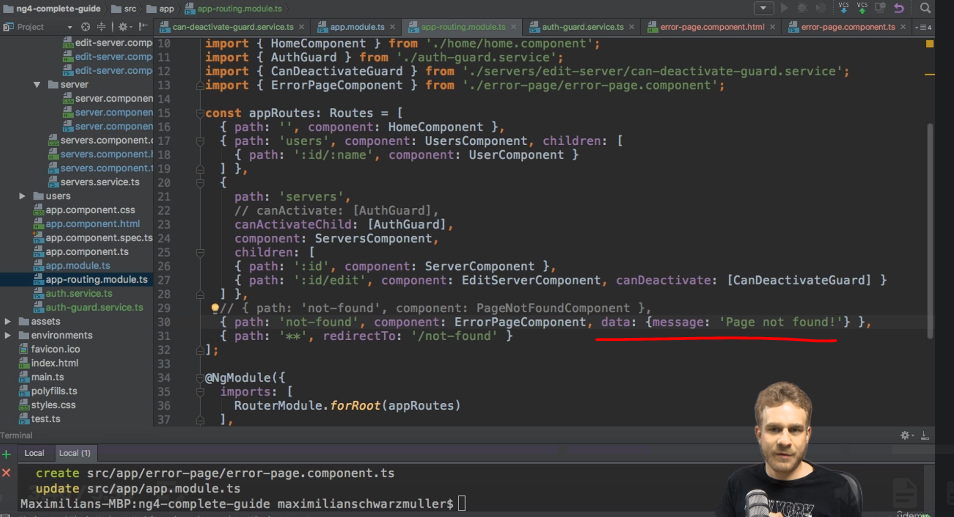
**150. Passing Static Data to a Route**

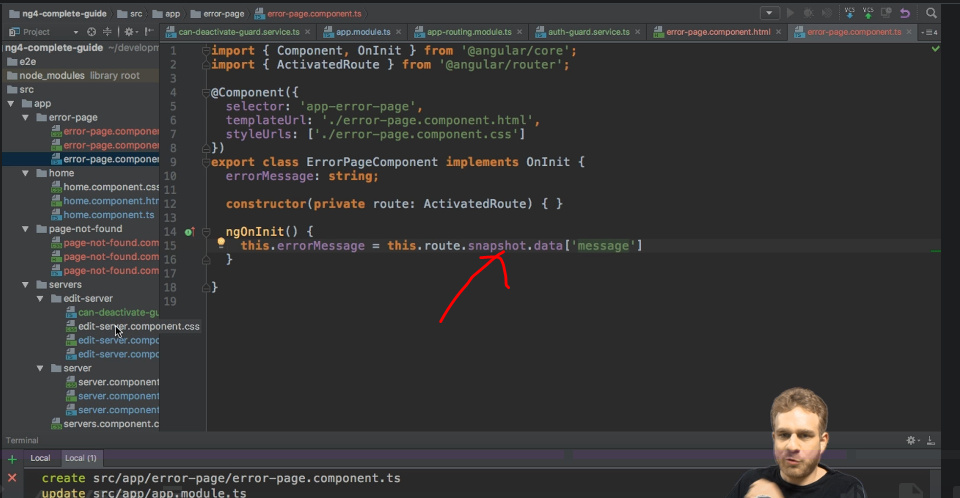
* -: Thus far we had a look at two route guards at the can actuate guard and its neighbor the can actuate child guard and at the can acuate guard.
* Later in this course in the modules section where you learn more about the angular module, we will have a look at our routing related questions and also at guards again, but I'll come back to this in this module here.
* There are still two other things I want to have a look at and ***the first one is how to get some data either static data or dynamic data once a route is loaded.***
* Now you might correctly say, Hey we learned how to get data how to get a parameter out of the URL and then use that.
* But that's not what I mean.
* I mean that some of your routes depend on data they receive either statically each time they are loaded or they will resolve dynamically.
* Maybe I show some examples to make this clearer.
* Let's start with static data.
* We get our page not found component here.
* Now let's say we don't want to use that but instead I will create a new component, the error page component.



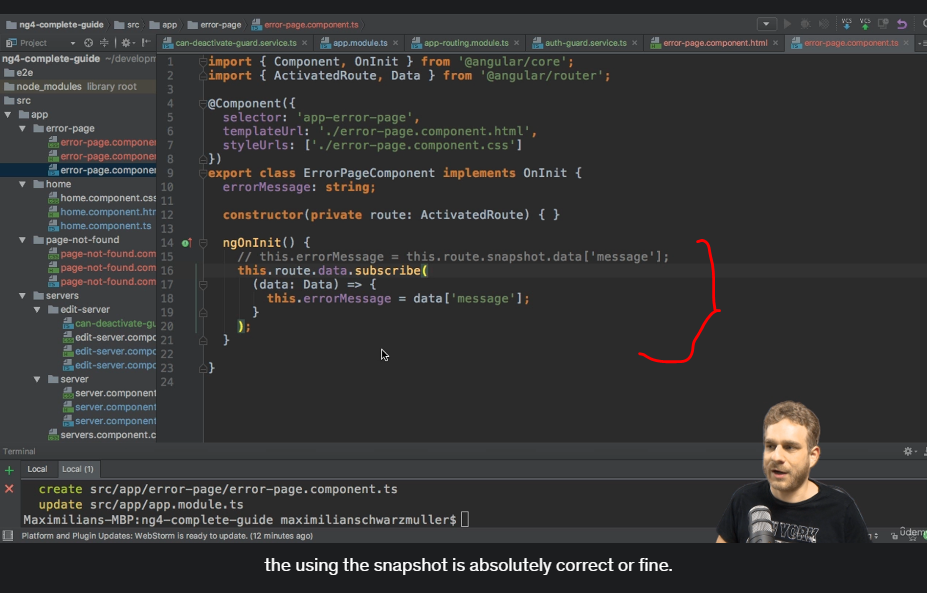
* *So this will be a generic error page where the page not found component is pretty much limited to well, 404 not found errors.*
* We could imagine that we reuse the error page for all kinds of errors.
* So I'll remove this back.
* ts file.
* And here in the template, now I don't want to output page not found or anything like this, instead here, let's say we have some error message which I want to output why string interpolation.
* The issue of course is this property doesn't exist as of now.
* So let's add it here.
* Let's add errorMessage to this component and we know it should be a string but still what would be the message? Because again, we might reuse this page.
* Now for routing, there only is one proper use case you want to target right now and that is that a route is not found.
* So in our route setup here, our redirect round here which redirects to not found, which loads the page not found component here in the end here I will duplicate this and then comment it out so that I have to code stored.
* Here I now want to use the brand new error page component.
* So this error page component which was generated by the CLI and for which you should add an import here.
* Of course you should have also added this to your main app module if DCLI didn't do this automatically for you here and the import two.
* So with this we're now loading this not found page or excuse me, this error page.
* If we target the not found route.
* Still the issue is that here we try to output some message which We haven't gotten here.
* Now we could reach this page through different hooks but or for different errors but for each error there is only one message.
* So in our app routing module, we know that if we have the not found route and therefore a not found error we will always display the same error message and we can pass such static data with the data property here the data property allows us to pass an object here and in this object we can define any key value pairs any properties we want, like for example our message which is page not found.



* Now again, you might say if I define it statically here and we will soon fetch it in the error page.
* Why don't I define it there to begin with? Because you might reuse this error page.
* Maybe we've got another route reaching this where we want to display a different error or we have a different logic in our application which might render this page.
* So again, it might be reusable, but for each use case there is a specific data field, a specific message you want to pass like the page not found message here.
* So with this we now want to retrieve that whenever we load our error page component.
* and for this like parents, like query parents, we need the active route.
* So let's inject or activated route here, make sure to import it.



* And in ngOnInit, I can then again either assign it by using the snapshot of this route and then simply the data object where I can now access message.
* And of course I have to make sure that this property exists and we do make sure because we know we can only reach this component through this route where we will have this message property set.
* So this is the way of using it with its snapshot or just like with parents or query parents if this could possibly change while you still are on this page, you want to use your route and hear the data observable to which you subscribe and which will give you this new data object here of type data, some dummy data type you can import from angular router which will in the end give you access to your data or the properties you defined there.
* So here you could then also set it equal to data message.
* Both would work depending on whether it may change or not.
* So yeah, both should work.
* I will comment out the first use case here, but again, if it doesn't change, the using the snapshot is absolutely correct or fine.



* So with this we should now bind this and therefore, if we save this and go back to our application enter some invalid route here, we correctly see page not found the static error message we passed with the data property and this is a typical use case whenever you have some static data you want to pass to a route.

Graphical user interface, application

Description automatically generated

* Now let's move on to some dynamic data in the next lecture.