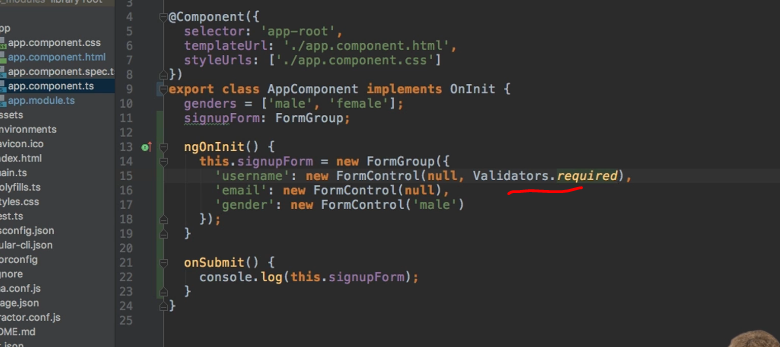
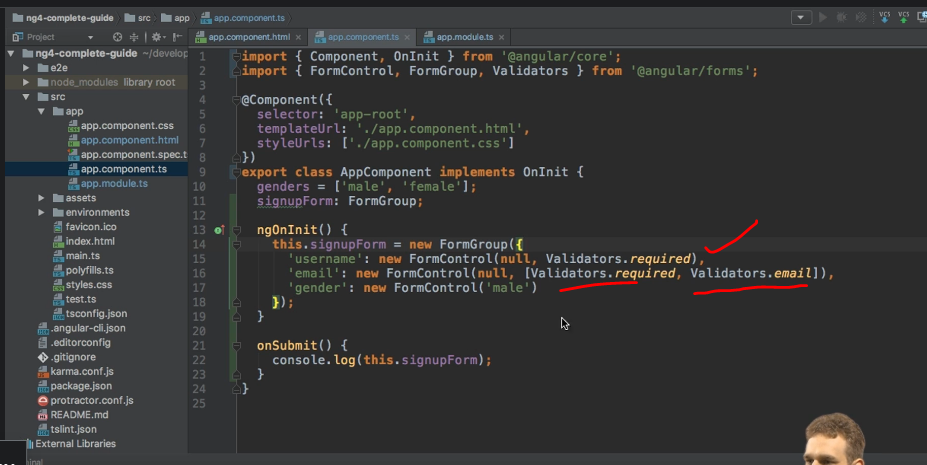
* -: Now that we're able to submit a form let's actually add some validation to it.
* In the template driven approach, we would simply add required here, for example, to make this field required.
* It doesn't work like this in the reactive approach because keep in mind, and that's key.
* **You're not configuring the form in the template.**
* **You're only synchronizing it with the directives form control name and form group.**
* but you're not configuring it here.
* The TypeScript code that is where you configure it.
* *That is why form control takes more than one argument.*
* *This constructor here you cannot only set the default value.*
* ***The second argument allows you to specify some validators.***
* So you can either only pass one validator here.
* Like for username, where I only want to make sure that it is required.
* And you do this by adding the validators object.
* Make sure to import it from at angle of forms.
* And there you get a couple of built in validators you can use.
* For example, required.
* Now make sure to not call it, to not execute it so don't have parenthesis after it because you don't want to execute this method.
* It is a static method made available by validators here.
* Instead, you only want to ***pass the reference to this method.***



* Angular will execute the method whenever it detects that the input of this form control changed.
* So it just needs to have a reference on what it should execute at this point of time.
* So this is the required validator.
* But as I said this is only one option.
* *You can also add multiple validators by simply passing an array of validators.*
* So here to email, I'm going to pass an array of validators because I want to have the required validator just like on the username.
* But I'll add another validator to here.
* I'll add the email validator.
* And this again all doesn't need to get executed.
* We just need to reference.
* So here I now also passed my validator and now we do have validation in place.



* If we now go back to our form and we inspect and let it reload, and we inspect the email field here for example, it's invalid.

Graphical user interface, text, application, email

Description automatically generated

* If I enter something here it's still invalid.
* If I enter email address though, it becomes valid.

Graphical user interface, application, email

Description automatically generated

* when the same of course for username, it's invalid right now.
* As soon as I enter something it becomes well because it's now not empty anymore.
* So this is how you can add validation to this approach.