* -: In the last lecture I was already mentioning that **get also takes the path to an element.**
* What did I mean with that? It's very simple.

Text

Description automatically generated

* You can specify a path here because you might have a nested form.
* *Let's say username and email should be inside a FormGroup.*
* We used such forms in the template driven approach too.
* So here we could create a *FormGroup named userData.*
* And now this is a new FormGroup again.
* ***FormGroup is not only there to be used on the overall form that just happens to be a FormGroup too.***
* ***But you can't still have FormGroups in FormGroups.***
* So here we could have a FormGroup, which of course like the outer FormGroup for the overall form also takes a JavaScript object.
* And now we can add our FormControls inside of this group.

Text

Description automatically generated

* **So now we have a nested form.**
* We need to reflect this in our HTML template because right now if we have a look at our form, it will break.

Graphical user interface, text, application, email

Description automatically generated

* You see this in the console, we get an error because correctly ***it cannot find a control with the name username because on the overall form there is no such control.***
* *It's nested in the userData FormGroup.*
* So first of all we need to **update our synchronization** and we easily do this ***by wrapping our FormControls*** here in another element like a div for example.
* So the email and username would be the two controls we wrap in there to replicate the structure we have in the TypeScript object.
* And on this diff, we place **the FormGroup name directive**.
* So you kind of could see the schema here FormControl name to tell Angular which property in our TypeScript object representing the form relates to which input and formGroupName to tell it the same for a FormGroup.
* So depending on whether you have new FormGroup or new FormControl here.
* So here the formGroupName is userData with a capital D.

Text

Description automatically generated

* And now with this tiny addition, I don't need to change anything else.
* It's almost working again, but we still get this error here or a different error, but still related to the topic that valid cannot be read or null.

Graphical user interface, text, application, email

Description automatically generated

* Well that makes sense because get username still fails

Text

Description automatically generated

* because what are we trying to do here? We try to get a FormControl with the name username on our signup form.
* We don't have to, here we have a nested userData but that's not the same.
* So ***we need to update, get to point to the path or to contain the path to the username.***
* And that would be userData.
* username.
* And it's as simple as this.
* ***You structure your path separated with dots.***

Text

Description automatically generated

* So if you have a new object, just place a dot in between.
* So let's paste it everywhere where we used one of these nested controls for email as well.
* Therefore, and if we now save this it should update and not display any errors anymore.

Graphical user interface, application, table

Description automatically generated

* Instead, we should see the behavior we saw before with the messages we get if we enter invalid data because now the form is working again taking into accounts, FormGroups.
* So nesting inside of our form now, and as you saw it was really easy to reflect this in our HTML code.
* It's all about keeping that in sync in the end.
* Student feedback 4.
* 6 Rating: 4.
* 6025734 out of 5 Course Rating