**187. Submitting and Using the Form**

* -: In the last lecture, we configured our form here .
* We added some controls by placing ngModel on the inputs .
* *Now let's make this form submittableso that we can actually see what the user entered .*

Text

Description automatically generated

* For this, I'll go to my app component .
* I already do have a method here, which you will use later .
* Let's ignore it for now .
* And I'll add a new method onSubmit maybe .
* This should be triggered whenever this form is submittedby the user .
* In onSubmit, I want to output whatever the user entered .
* Well first of all, we need to call this method .
* So back in the template, how can we call onSubmit?Now you might thinkthat a good place would be on a click listeneron this button here at the bottom right?Because this is the button we clickwhen we want to submit the form .
* However, this is not the best place to do it .
* Keep in mind that this button here is of type submit .
* So if we click it,as it placed inside of a HTML form element,something else will happen .
* The default behavior of HTML will be triggeredto call it like this .
* If you have a button in a form element,this button will submit the form,will send a request normally .
* But besides that, it will also trigger a JavaScript eventthe submit event, that's built into JavaScriptbuilt into html you could say .

Text

Description automatically generated

* **ngSubmit:**
* Angular takes advantage of this .
* It gives us a directivewe can place on this form element as a whole .
* It is called ngSubmitand it actually only gives us one event we can listen to .
* So let's wrap it in parenthesis .
* This event made available by the ngSubmit directivewill be fired whenever this form is submitted .

Text

Description automatically generated

* So whenever this default behavior is triggeredand here we can call onSubmit, like thisand to show you that this works,I will simply go into onSubmit and log Submitted here,just like that .
* Now, if we save this and let it recompilelet's go back and let's open up the developer tools .
* And if I now hit the submit buttonyou see submitted here on the rightbecause indeed, the form gets submitted .

Graphical user interface, application, Word

Description automatically generated

* Now it would be niceto see the actual values of the form though,to see that form object .
* And to see it,we have to go back to our templatebecause we're in the template-driven approach .
* So as a rule of fund, everything you doyou do it in the template .
* Everything you want to change about this formyou want to add on as functionalityyou do it in a template .
* On this form object,we want to get access to the form created by Angular .
* Now you learn about local referencesyou can place on HTML elements to get access to them .

Text

Description automatically generated

* So we could place #F on the form elementand now we could access this form elementon well the F reference in our template .
* And we could pass F as a argument to the onSubmit methodand print it there .
* So now we know that we get the formand actually this will be of type element ref as we learned .
* So if we import Element Refand make this of type element ref hereand we output the form here, so this element .

Text

Description automatically generated

* If we now go back to this and hit submit,yeah, we indeed see that form object .

Graphical user interface, application

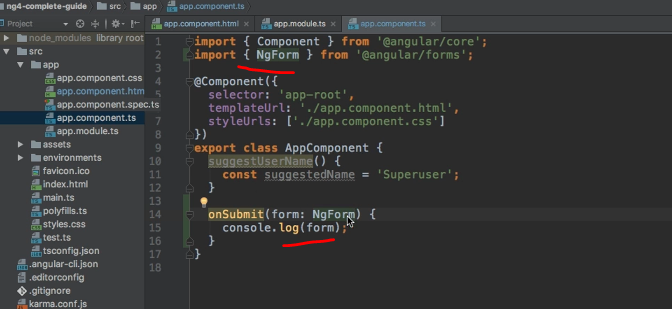
Description automatically generated

* We see some strange classes here, I will come back to thembut that still is not what we want .
* It's not a JavaScript object created by Angular .
* But this object is therewe just need to know how to get to it .
* And there actually is a trick you could call it,but it's no trick, it's the default way to get accessto this automatically created object .
* You set this local reference equal to something .
* Now we haven't done that yet,but we can set it equal to something exposedby this form here .
* Keep in mind the form element is kind of a selectorfor a directive built into Angularwhich will create this JavaScript object automaticallyand then it will expose some data we can fetch hereon this form element .
* We can get access to itby writing ngForm here between the quotation marks .

Text

Description automatically generated

* So this might look super strangebut what this in the end does is,it tells Angular, "Hey please give me accessto this form you created automatically .
* "That's just something you have to keep in mind .
* This is how you get access to the formto this JavaScript object created by Angular, automatically .
* So therefore, here where we get this form,we now still pass it .
* Now this will no longer be a element refSo let's remove this type here,instead this will now be of ngForm .



* So let's import ngForm from @Angular formsand kind of makes sense that it is of type ngFormbecause that is what we're accessing here .
* This automatically created form .
* Now let's print it to the console one more timeand let's actually enter something into these fieldsto see that it worked .

Graphical user interface, text, application

Description automatically generated

* If I now type submit here,you see that now we get ngForm object,object which we certainly didn't create .
* And there we have a lot of properties .
* We'll have a close look at those later .
* We also do have a value property .

Timeline

Description automatically generated

* And if we expand this,we indeed see a couple of key value pairs here,where we have the names of the controls,so the names we set up here on the name attribute,like username and email .
* We find them here,and then the values the user entered .
* And this is how we can submit such a formand how we can also get accessto the form object created by Angular .
* We also see that our setup of controls also worked .