Angular makes working with forms easier than ever. There are two ways to work with forms in Angular, one being template-driven and the other being reactive forms. To enable either of these form styles you have to import the FormsModule from @angular/forms in the controller. The template-driven forms are used good for simple forms with validation that isn’t to complex using built in features of Angular. Using this form, you add the NgModel directive which enables two-way binding to the form element and then you can handle all validation in the template itself and not inside the controller. Template-driven forms are better for simple form scenarios although unit testing can be a challenge. Reactive forms are used when you have a very large form with complex validation. The reactive forms allow you to implement custom validation easier and you have more control over the form. When you utilize the reactive form approach you are able to implement unit testing easier than you would be able to in template-driven forms. When dealing with form error handling you can utilize the \*ngIf directive to check to see if the form field is valid, if not then display message. Bellow is an example of an input field with error message.

<input required ngModel name=”firstName” #firstName=”ngModel” type=”text”>

<div class=”error” \*ngIf=”!firstName.valid”>First Name is required</div>

The above example shows an input field with the ngModel directive that binds it to Angular. The input also has the “required” statement that angular will use to determine if it is valid. The div bellow the input contains an error message that the first name field is required. There is also the ngIf directive saying if first name is not valid then display the contents of this div.

References

Medium.com/@agoiabeladeyemi/template-driven-forms-in-angular-4a5a6ad960de