

Exercise (Instructions): Angular Reactive Forms Part 3

Objectives and Outcomes

In this exercise you will learn further about adding form validation to your reactive forms in Angular. You will do much of the form validation and generate the error messages within the code and map them into the form. You will take the help of an Angular Forms observable for this exercise. At the end of this exercise you will be able to:

- Add in form validation support to your reactive forms from within the code using observables
- Show validation error messages to the users in your view

Add Form Validation

- Open `contact.component.ts` and update the Form Model as shown below:

```
1  this.feedbackForm = this.fb.group({
2    firstname: ['', [Validators.required, Validators.minLength(2), Validators
      .maxLength(25)] ],
3    lastname: ['', [Validators.required, Validators.minLength(2), Validators
      .maxLength(25)] ],
4    telnum: ['', [Validators.required, Validators.pattern] ],
5    email: ['', [Validators.required, Validators.email] ],
6    agree: false,
7    contacttype: 'None',
8    message: ''
9  });
```

- Continuing with `contact.component.ts`, you will now subscribe to the Angular Form observable named `valueChanges` and initiate form validation as follows:

```

1  . . .
2
3  formErrors = {
4    'firstname': '',
5    'lastname': '',
6    'telnum': '',
7    'email': ''
8  };
9
10 validationMessages = {
11   'firstname': {
12     'required':      'First Name is required.',
13     'minlength':     'First Name must be at least 2 characters long.',
14     'maxlength':     'FirstName cannot be more than 25 characters long.'
15   },
16   'lastname': {
17     'required':      'Last Name is required.',
18     'minlength':     'Last Name must be at least 2 characters long.',
19     'maxlength':     'Last Name cannot be more than 25 characters long.'
20   },
21   'telnum': {
22     'required':      'Tel. number is required.',
23     'pattern':       'Tel. number must contain only numbers.'
24   },
25   'email': {
26     'required':      'Email is required.',
27     'email':         'Email not in valid format.'
28   },
29 };
30
31 . . .
32
33 createForm(): void {
34
35   . . .
36
37   this.feedbackForm.valueChanges
38     .subscribe(data => this.onValueChanged(data));
39
40   this.onValueChanged(); // (re)set validation messages now
41
42 }
43
44 . . .
45
46 onValueChanged(data?: any) {
47   if (!this.feedbackForm) { return; }
48   const form = this.feedbackForm;
49   for (const field in this.formErrors) {
50     // clear previous error message (if any)
51     this.formErrors[field] = '';
52     const control = form.get(field);
53     if (control && control.dirty && !control.valid) {
54       const messages = this.validationMessages[field];
55       for (const key in control.errors) {
56         this.formErrors[field] += messages[key] + ' ';
57       }
58     }
59   }
60 }
61 }
62

```

- Next update the form in contact.component.html as follows:

```

1      <form novalidate [formGroup]="feedbackForm" (ngSubmit)="onSubmit()">
2          <p>
3              <md-input-container class="half-width"
4                  dividerColor="{{(formErrors.firstname) ? 'warn' : 'primary'}}">
5                  <input mdInput formControlName="firstname" placeholder="First
6                      Name" type="text" required>
7                  <md-hint>
8                      <span [hidden]="!(formErrors.firstname)">
9                          {{formErrors.firstname}}</span>
10                 </md-hint>
11             </md-input-container>
12             <md-input-container class="half-width"
13                 dividerColor="{{(formErrors.lastname) ? 'warn' : 'primary'}}">
14                 <input mdInput formControlName="lastname" placeholder="Last Name"
15                     type="text" required>
16                 <md-hint>
17                     <span [hidden]="!(formErrors.lastname)">
18                         {{formErrors.lastname}}</span>
19                     </md-hint>
20                 </md-input-container>
21             </p>
22             <p>
23                 <md-input-container class="half-width"
24                     dividerColor="{{(formErrors.telnum) ? 'warn' : 'primary'}}">
25                     <input mdInput formControlName="telnum" placeholder="Tel. Number"
26                         type="tel" pattern="[0-9]*" required>
27                     <md-hint>
28                         <span [hidden]="!(formErrors.telnum)">
29                             {{formErrors.telnum}}</span>
30                         </md-hint>
31                     </md-input-container>
32                 <md-input-container class="half-width"
33                     dividerColor="{{(formErrors.email) ? 'warn' : 'primary'}}">
34                     <input mdInput formControlName="email" placeholder="Email" type
35                         ="email" email required>
36                     <md-hint>
37                         <span [hidden]="!(formErrors.email)">
38                             {{formErrors.email}}</span>
39                         </md-hint>
40                     </md-input-container>
41                 </p>
42                 <table class="form-size">
43                     <tr>
44                         <td>
45                             <md-slide-toggle formControlName="agree">May we contact you?</md
46                                 -slide-toggle>
47                         </td>
48                         <td>
49                             <md-select placeholder="How?" formControlName="contacttype">
50                                 <md-option *ngFor="let ctype of contactType" [value]="ctype">
51                                     {{ ctype }}
52                                 </md-option>
53                             </md-select>
54                         </td>
55                     </tr>
56                 </table>
57             <p>
58                 <md-input-container class="full-width">
59                     <textarea mdInput formControlName="message" placeholder="Your
60                         Feedback" rows=12></textarea>
61                 </md-input-container>
62             </p>
63             <button type="submit" md-button class="background-primary text-floral
64                 -white" [disabled]="feedbackForm.invalid">Submit</button>
65         </form>

```

- Save all the changes and do a Git commit with the message "Reactive Forms Part 3".

Conclusions

In this exercise you used an Angular forms observable and subscribed to it to initiate form validation in code.

Mark as completed

