

Angular

Lab 4: Reactive Forms Exercises

1	LAB SETUP	1
2	GENERATING A NEW ANGULAR PROJECT	1
3	ADDING A BASIC FORM CONTROL USING FORMCONTROL	1
4	USING FORMGROUP TO GROUP RELATED FORM CONTROLS	1
5	SUBMITTING THE FORM AND ACCESSING FORM AND FORM CONTROL VALUES	2
6	UPDATING ALL OR A PORTION OF A FORMGROUP	2
7	CONTROL STATUS FOR A FORMCONTROL AND FORMGROUP	2
8	VALIDATING FORM CONTROLS	2
9	DISPLAY VALIDATION ERROR MESSAGES	2

1 Lab setup

2 Generating a new Angular project

3 Adding a basic form control using FormControl

Q1. Create a [radio button](#) with a random value and bind it to a FormControl instance property in the component

Q2: Use interpolation to display the value of the radio button in the template itself.

4 Using FormGroup to group related form controls

Q1. In the template, add in a new [numeric input field](#) for age just after the email field in the main FormGroup. Bind this element to a new FormControl instance in the component.

Q2. In the template, add in a new [text input field](#) for zipcode just after the state field in the nested FormGroup. Bind this element to a new FormControl instance in the component.

Q3: Use interpolation to display the values of these two new fields in the template itself.

5 Submitting the form and accessing form and form control values

Q1: Repeat Q1 and Q2 from the previous section. Add in statements in the component method that handles the submitted form data in order to obtain the value for the age and zipcode form elements.

6 Updating all or a portion of a FormGroup

Q1: Repeat Q1 and Q2 from the Section 4. Add in additional logic to the component method that uses setValue in order to provide new values for the zipcode and age controls.

Q2: Add in additional logic to the component method that uses patchValue in order to provide new values for the zipcode and age controls.

7 Control status for a FormControl and FormGroup

No exercises here

8 Validating form controls

Q1: Add an additional validator to the first name field so that it only contains letters (no numbers, spaces or any other characters). Hint: [Angular - Validators](#)

9 Display validation error messages

Q1: Add three validators to the job field:

- Make it required
- Make it have a maximum of 10 characters
- Make it only contain letters (no numbers, spaces or any other characters). Hint: [Angular - Validators](#)

Q2: Add appropriate conditional logic to display validation error messages for the job field.

10 Using FormBuilder as a shortcut for creating FormGroup

No exercises here

11 Using FormArray to build dynamic forms

Q1: Add another FormArray called `addresses` to the profile FormGroup items of the main profiles FormArray in the component. This FormArray `addresses` will contain FormGroup items as elements, where each FormGroup item has the following FormControl:

- Street: string
- City: string

Provide the required methods to

- include the addresses FormArray into the main profile FormGroup
- access the addresses FormArray
- create a new FormGroup item to place into the FormArray
- add this new FormGroup item into the FormArray
- remove an existing FormGroup item from the FormArray

Q2: Add the required statements in the template to support the items created in Q1 that allow the user to add and remove new addresses from a given top-level profile

Sample output from adding 3 new addresses to a single profile

The screenshot displays a web form with the following sections:

- Profiles**: Contains an "Add new profile" button and a "Remove Profile" button.
- Jobs**: Contains an "Add info for new job" button.
- Addresses**: Contains an "Add new address" button and three address entries, each with a "Remove Address" button.

The first address entry is labeled "Address # 1" and contains the following fields:

- Street: 111 Good
- City: New York

The second address entry is labeled "Address # 2" and contains the following fields:

- Street: 222 Cool
- City: London

The third address entry is labeled "Address # 3" and contains the following fields:

- Street: 333 Fast
- City: Bangkok

12 Adding other form control elements (radio buttons, checkboxes, etc) to a reactive form

No exercises