Lab - Building a Reactive Form in Angular

**Objective:** In this lab exercise, you will learn how to create a reactive form in Angular. By the end of this exercise, you will be able to build a form that handles user input and validation using Angular's reactive form approach.

**Prerequisites:**

Basic knowledge of Angular and TypeScript.

Angular CLI installed globally.

**Step 1: Set up the Angular project**

1. Open a terminal or command prompt.
2. Create a new Angular project by running the following command:



1. Change into the project directory:



**Step 2: Create a component for the reactive form**

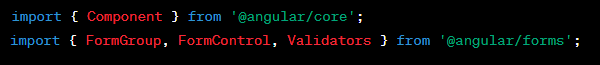
1. Generate a new component by running the following command:



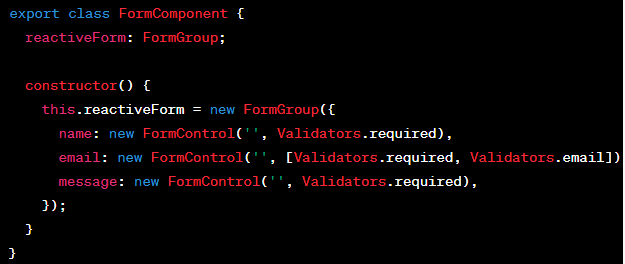
1. Open the newly created component file (form.component.ts) in your preferred code editor.

**Step 3: Import the necessary modules and define the form**

1. Import the ReactiveFormsModule from @angular/forms in the component file:

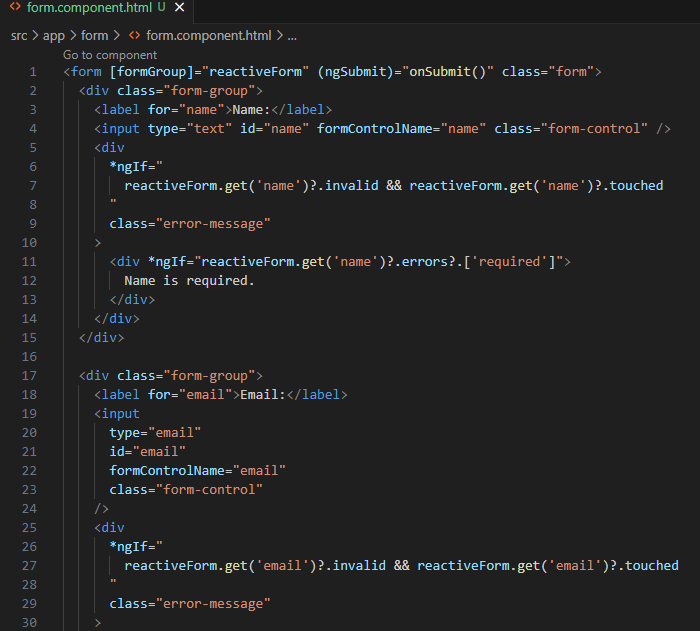


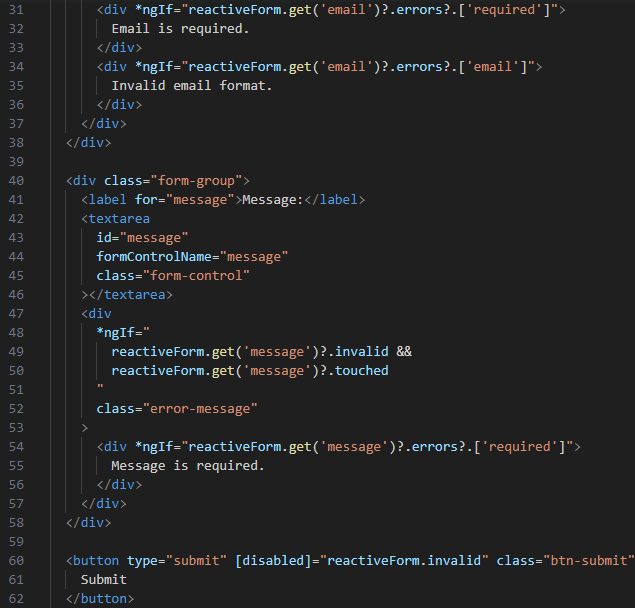
1. Add the following code within the component class:



**Step 4: Create the form template**

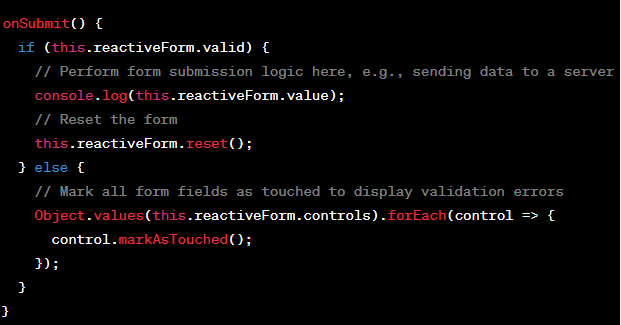
1. Open the component template file (form.component.html) in your code editor.
2. Replace the default content with the following code:





**Step 5: Implement form submission**

1. Go back to the component file (form.component.ts).
2. Add the following method within the component class:



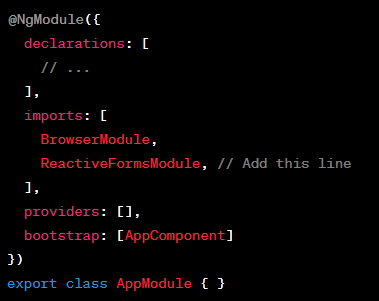
**Step 6: Update app.module.ts**

To complete the setup, you need to update the app.module.ts file to import and include the ReactiveFormsModule module. Here are the steps:

1. Open the app.module.ts file located in the src/app directory.
2. Import the ReactiveFormsModule module from @angular/forms by adding the following import statement at the top of the file:



1. In the @NgModule decorator's imports array, include the ReactiveFormsModule module:

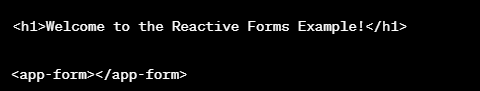


1. Save the file.

By adding the ReactiveFormsModule module to the imports array in app.module.ts, you make the necessary form-related features available throughout your Angular application.

**Step 7: Embed the form component selector in app.component.html**

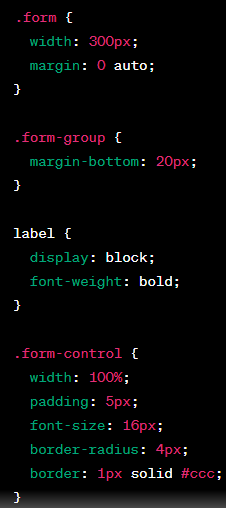
We need to embed the form component selector in the app.component.html file to display the form. Here's an updated version of the app.component.html file:

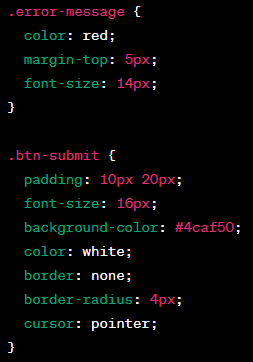


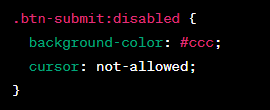
By adding the <app-form></app-form> selector, we include the FormComponent in the AppComponent template and display the form.

**Step 7: Apply some CSS styling to the form**

To apply the improved spacing, you can add the following CSS styles to your **styles.css** or **styles.scss** file:





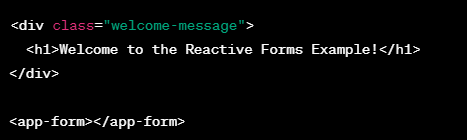


The CSS styles provided include spacing for the form, margin-bottom for each form group, label styles, input field styles, error message styles, and button styles.

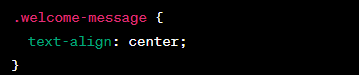
Feel free to adjust the styles as needed to achieve the desired spacing and appearance.

**Step 8: Center the form heading**

To center the text "Welcome to the Reactive Forms Example!", you can apply CSS styles to the app.component.html template. Here's an updated version:



To center the text, add the following CSS styles to your styles.css or styles.scss file:



The .welcome-message class applies the text-align: center; property to center-align the text within the <h1> tag.

With these changes, the text "Welcome to the Reactive Forms Example!" should be centered on the page.

**Step 9: Test the reactive form**

1. Open a terminal or command prompt.
2. Update package.json as follows:



1. Run the Angular development server by executing the following command:



1. Open a web browser and navigate to [http://localhost:4200](http://localhost:4200%20).
2. The reactive form should be displayed. Try entering different values and observe the validation errors.
3. Fill in all the required fields with valid data and click the Submit button. Check the console for the logged form data.
4. Verify that the form gets reset after submission.

Congratulations! You have successfully built a reactive form in Angular. This form allows users to input data, performs validation, and handles form submissions. Feel free to experiment further and enhance the form.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*