2.6 Routing Mechanism



This section will guide you to:

* Add a router to your Angular application
* Change the source code to implement the sign-up page

This guide has four subsections, namely:

2.6.1 Configuring the Angular application

2.6.2 Changing source code to implement the routers

2.6.3 Completing the functionality by implementing validators

2.6.4 Pushing the code to GitHub repositories

**Step 2.6.1:** Configuring the Angular application

* Create a folder using mkdir command and navigate to the created folder
* Type ng new <app\_name> command to create an angular application
* Open Visual Studio Code and import the application by following the steps mentioned below:
* Click on **File** 🡪 **Open Folder** and provide the path to where you have installed your Angular application.
* Click on **Terminal** 🡪 **New Terminal** to open the terminal of Visual Studio Code.
* Use **ng serve** in the terminal to start the local server and to make it run at port 4200.

**Step 2.6.2:** Changing source code to implement the routers

* Open the **app.component.html** available in the **app** folder of the Angular application. Replace all the code available in the file to the source code mentioned below:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Page Title</title>

</head>

<body style="padding:40px ">

<p style="text-align: center;font-size: 30px"> Hello <input type="text" placeholder="Your name" (input)="ontyping($event)"/>, Welcome to the <b>angCare!</b></p>

<p style="text-align: center;font-size: 20px">Hello {{name}}<br>I am Joe, your personal assistant! I will guide you further...</p>

<div style="text-align:left;padding: 20px">

<p> Click on Sign up to create your account with angCare: <button (click)="signup()" class="btn btn-primary" [routerLink]="'/signup'">Sign up with {{title}}</button> : {{status}}</p>

<p> Click on Sign up to create your account with angCare: <button>Sign In </button></p>

</div>

</body>

</html>

<router-outlet></router-outlet>

**Step 2.6.3:** Completing the functionality by implementing validators

* Open the **app.component.ts** file and replace only the source code available in **AppComponent** class with the following source code:

import { Component,OnInit } from '@angular/core';

import {FormBuilder, FormGroup, Validators } from '@angular/forms';

import { Title } from '@angular/platform-browser';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

constructor(private formBuilder: FormBuilder) { }

title = 'angCare';

status = 'You haven\'t signed up yet';

name = '';

ontyping(event:Event) {

this.name = (<HTMLInputElement>event.target).value;

}

signup() {

this.status = 'Oops! We are working on it!';

}

}

* Open the **app.module.ts** file and add the source code mentioned below:

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import {RouterModule, Routes} from '@angular/router'

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { ReactiveFormsModule} from '@angular/forms'

import { SignupComponent } from './signup/signup.component';

import { SigninComponent } from './signin/signin.component';

const routes: Routes = [{

path:'',

component:AppComponent

},

{

path:'signup',

component:SignupComponent

},

{

path:'sigin',

component:SigninComponent

}

]

@NgModule({

declarations: [

AppComponent,

SignupComponent,

SigninComponent

],

imports: [

BrowserModule,

AppRoutingModule,

ReactiveFormsModule,

RouterModule.forRoot(routes)

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

* Open the **signup.component.html** available in the **app/signup** folder of the Angular application. Replace all the code available in the file with the source code mentioned below:

<p>

signup works!

</p>

<div class="jumbotron">

<div class="container">

<div class="row">

<div class="col-md-6 offset-md-3">

<h2>Angular 6 Reactive Form Validation</h2>

<form [formGroup]="registerForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label>First Name</label>

<input type="text" formControlName="firstName" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.firstName.errors }" />

<div \*ngIf="submitted && f.firstName.errors" class="invalid-feedback">

<div \*ngIf="f.firstName.errors.required">First Name is required</div>

</div>

</div>

<div class="form-group">

<label>Last Name</label>

<input type="text" formControlName="lastName" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.lastName.errors }" />

<div \*ngIf="submitted && f.lastName.errors" class="invalid-feedback">

<div \*ngIf="f.lastName.errors.required">Last Name is required</div>

</div>

</div>

<div class="form-group">

<label>Email</label>

<input type="text" formControlName="email" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.email.errors }" />

<div \*ngIf="submitted && f.email.errors" class="invalid-feedback">

<div \*ngIf="f.email.errors.required">Email is required</div>

<div \*ngIf="f.email.errors.email">Email must be a valid email address</div>

</div>

</div>

<div class="form-group">

<label>Password</label>

<input type="password" formControlName="password" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.password.errors }" />

<div \*ngIf="submitted && f.password.errors" class="invalid-feedback">

<div \*ngIf="f.password.errors.required">Password is required</div>

<div \*ngIf="f.password.errors.minlength">Password must be at least 6 characters</div>

</div>

</div>

<div class="form-group">

<button [disabled]="loading" class="btn btn-primary">Register</button>

</div>

</form>

</div>

</div>

</div>

</div>

<router-outlet></router-outlet>

* Open the **signup.component.ts** file and replace the source code available in **SignupComponent** class with the following source code:

import { Component, OnInit } from '@angular/core';

import {FormBuilder, FormGroup, Validators } from '@angular/forms';

@Component({

selector: 'app-signup',

templateUrl: './signup.component.html',

styleUrls: ['./signup.component.css']

})

export class SignupComponent implements OnInit {

submitted = false;

registerForm: FormGroup;

constructor(private formBuilder: FormBuilder) { }

ngOnInit() {

this.registerForm = this.formBuilder.group({

firstName: ['', Validators.required],

lastName: ['', Validators.required],

email: ['', [Validators.required, Validators.email]],

password: ['', [Validators.required, Validators.minLength(6)]]

});

}

get f() { return this.registerForm.controls; }

onSubmit() {

this.submitted = true;

// stop here if form is invalid

if (this.registerForm.invalid) {

return;

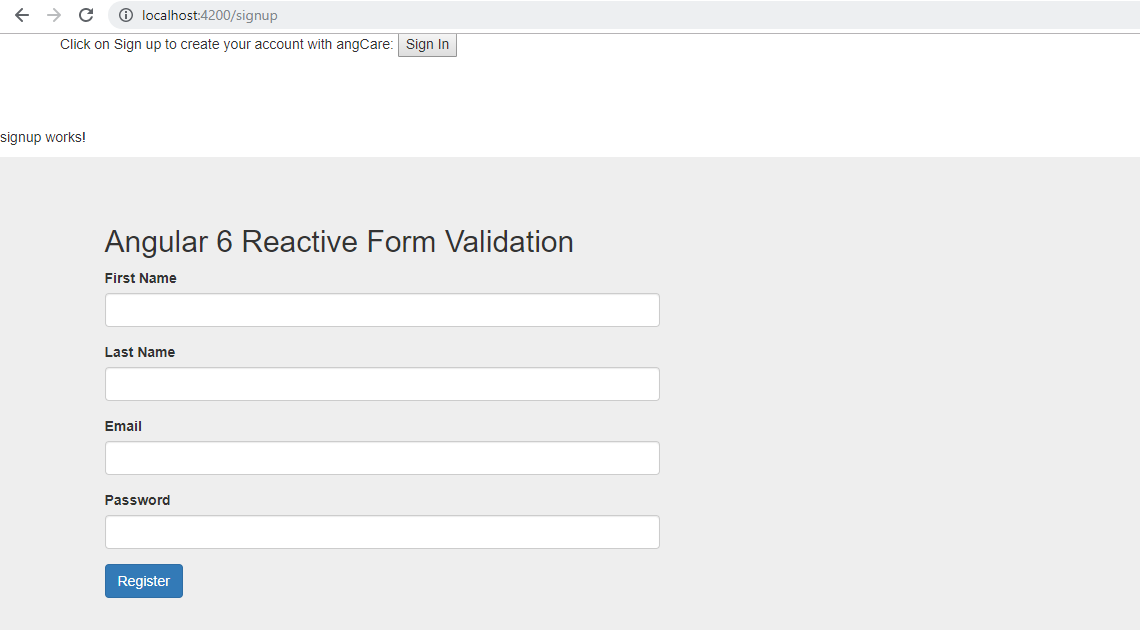
}

alert('Your request has been submitted for approval')

}

}

* Use **ctrl+s** to save the changes. The Angular application will automatically load. Once you click on **Signup with angCare**, you should be able to see the below form in the path localhost:4200/signup as shown in the screenshot:



**Step 2.6.4:** Pushing the code to GitHub repositories

Open your command prompt and navigate to the folder where you have created your files

**cd <folder path>**

Initialize your repository using the following command:

**git init**

Add all the files to your git repository using the following command:

**git add .**

Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

Push the files to the folder you initially created using the following command:

**git push -u origin master**