Reactive Form in Angular

Create app: ng new formdemo

Move: cd formdemo

Open VS Code: code .

Run app: ng serve, see output check,localhost:4200

Let’s implement Reactive Form Module.

Open app.module.ts and import reactiveFormsModule

Check imports too

import { FormsModule, ReactiveFormsModule } from '@angular/forms';

  imports: [

    BrowserModule,

    AppRoutingModule,

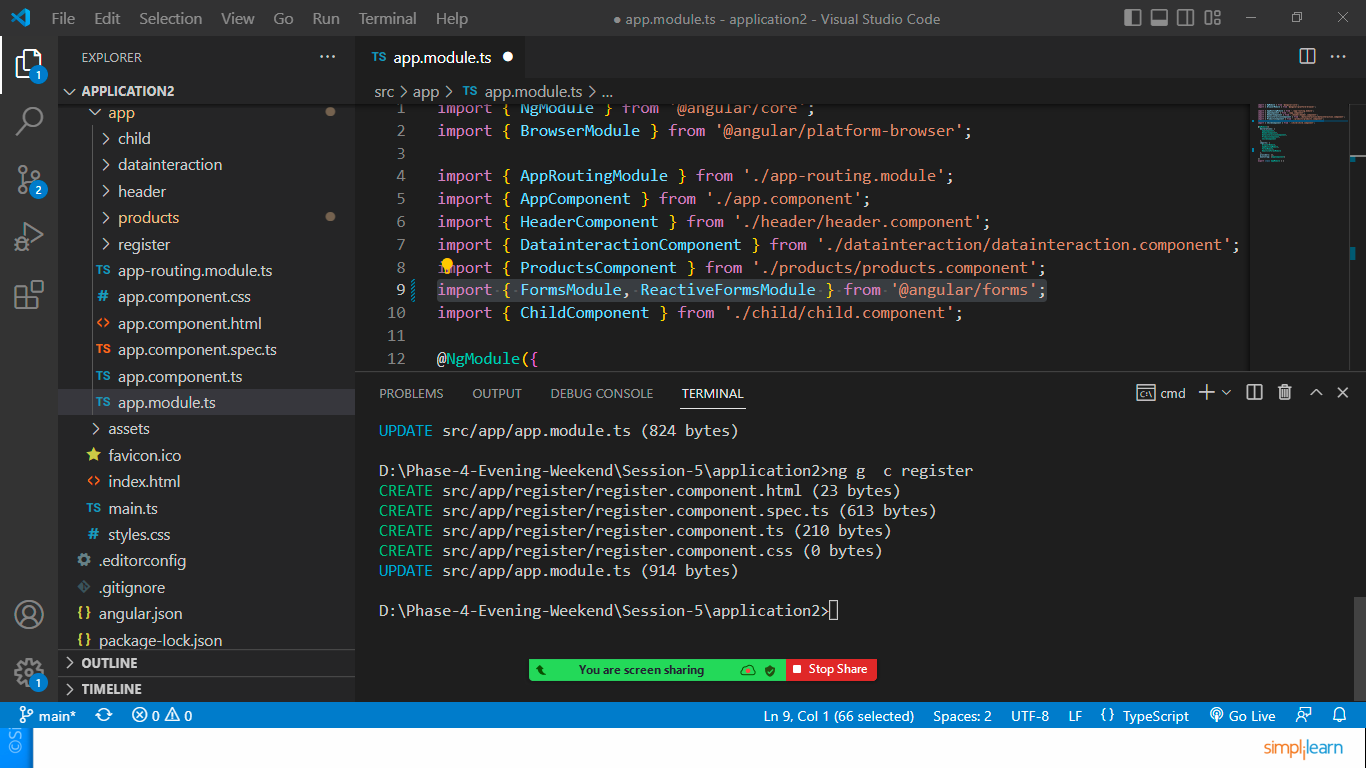
    FormsModule,

    ReactiveFormsModule

  ],

Let’s create a component called register

ng g c register



Let’s include this component in app.component.html file

<div class="container">

  <app-register></app-register>

</div>

Note: whenever we are declaring a variable without its initialization it can give you error. To resolve this error open **tsconfig.json** file add below mentioned property.

{

  "compileOnSave": false,

  "compilerOptions": {

    "strictPropertyInitialization": false,

    "baseUrl": "./",

    "outDir": "./dist/out-tsc",

    "forceConsistentCasingInFileNames": true,

    "strict": true,

    "noImplicitOverride": true,

    "noPropertyAccessFromIndexSignature": true,

    "noImplicitReturns": true,

When you update this the error will removed.

Register.component.ts

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

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

@Component({

  selector: 'app-register',

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

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

})

export class RegisterComponent {

  registerForm:FormGroup

  submitted:boolean=false //default the form is not submitted

  //injected formbuilder dependency in constructor to create a form

  constructor(private builder:FormBuilder){}

  //loads when the component intialized

  ngOnInit():void{

    this.registerForm = this.builder.group({

      firstName:["",Validators.required],

      lastName:["",Validators.required],

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

      password:["",[Validators.required,Validators.minLength(8)]]

    })

  }

  //this f we are using in ngClass attribute for validation error

  get f(){

    return this.registerForm.controls

  }

  onsubmit():void{

    this.submitted=true;

    if(this.registerForm.invalid)

      return

    alert("Form Submitted successfully")

  }

}

Register.component.html file

<h1 class="bg-warning">Registration Form</h1>

<div class="row">

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

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

            <div class="form-group">

                <label>FirstName</label>

                <input type="text" formControlName="firstName" class="form-control"

                [ngClass]="{'is-invalid':submitted && f['firstName'].errors}">

                <div \*ngIf="submitted && f['firstName'].errors">

                    <div \*ngIf="submitted && f['firstName'].errors['required']">

                        FirstName should not be empty

                    </div>

                </div>

            </div>

            <div class="form-group">

                <label>LastName</label>

                <input type="text" formControlName="lastName" class="form-control"

                [ngClass]="{'is-invalid':submitted && f['lastName'].errors}">

                <div \*ngIf="submitted && f['lastName'].errors">

                    <div \*ngIf="submitted && f['lastName'].errors['required']">

                        LastName should not be empty

                    </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">

                    <div \*ngIf="submitted && f['email'].errors['required']">

                        email should not be empty

                    </div>

                    <div \*ngIf="submitted && f['email'].errors['email']">

                        Invalid Email Id

                    </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">

                    <div \*ngIf="submitted && f['password'].errors['required']">

                        Password should not be empty

                    </div>

                    <div \*ngIf="submitted && f['password'].errors['minlength']">

                        Password must be 8 characters long

                    </div>

                </div>

            </div>

            <button type="submit" class="btn btn-primary">Register</button>

        </form>

    </div>

</div>