Reactive Forms

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

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

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

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

import { HelloWorldComponent } from './hello-world/hello-world.component';

import { HeaderComponent } from './header/header.component';

import { FooterComponent } from './footer/footer.component';

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

@NgModule({

declarations: [

AppComponent,

HelloWorldComponent,

HeaderComponent,

FooterComponent

],

imports: [

BrowserModule,

AppRoutingModule,

ReactiveFormsModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

Create a new interface file User.ts

export interface User{

name:string;

contact:string;

email:string;

}

Import User.ts in app.component.ts

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

import {User } from './User';

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent implements OnInit{

userList:User[]=[];

form:FormGroup;

constructor(){}

ngOnInit(){

this.form=new FormGroup({

name: new FormControl(''),

contact: new FormControl(''),

email:new FormControl('')

});

}

}

<h1>Reactive Forms</h1>

<form class="form-group">

<div class="">

<label for="">Name:</label>

<input type="text" name="" class="form-control">

</div>

<div class="">

<label for="">Contact:</label>

<input type="text" name="" class="form-control">

</div>

<div class="">

<label for="">Email:</label>

<input type="text" name="" class="form-control">

</div>

</form>

Associate FormGroup & FormControl into html controls.

<h1>Reactive Forms</h1>

<form novalidate [formGroup]="form" class="form-group" (ngSubmit)="addUser(form)">

<div class="">

<label for="">Name:</label>

<input type="text" name="" class="form-control" formControlName="name">

</div>

<div class="">

<label for="">Contact:</label>

<input type="text" name="" class="form-control" formControlName="contact">

</div>

<div class="">

<label for="">Email:</label>

<input type="text" name="" class="form-control" formControlName="email">

</div>

</form>

Now add submit button form submit.

<button type="submit" > Add User</button>

addUser(form){

console.log(form.value);

this.userList.push(form.value);

}

<ul class="list-group">

<li \*ngFor="let user of userList" class="list-group-item">

{{user.name | json}}

</li>

</ul>

Perform validation

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

ngOnInit(){

this.form=new FormGroup({

name: new FormControl('',Validators.required),

contact: new FormControl('',Validators.required),

email:new FormControl('',Validators.required)

});

}

<button type="submit" [disabled]="form.invalid" class="btn btn-success"> Add User</button>

Other Validation types

ngOnInit(){

this.form=new FormGroup({

name: new FormControl('',[Validators.required,Validators.pattern('[a-zA-Z][a-zA-Z]+')]),

contact: new FormControl('',[Validators.required,Validators.pattern('[0-9]\*'),Validators.minLength(3),Validators.maxLength(10)]),

email:new FormControl('',[Validators.required,Validators.email])

});

}

<h1>Reactive Forms</h1>

<form novalidate [formGroup]="form" class="form-group" (ngSubmit)="addUser(form)">

<div class="">

<label for="">Name:</label>

<input type="text" name="" class="form-control" formControlName="name">

</div>

<div \*ngIf="form.get('name').touched && form.get('name').invalid" class="alert alert-danger">

<div \*ngIf="form.get('name').hasError('required')">

Name is required

</div>

<div \*ngIf="form.get('name').hasError('pattern')">

Invalid pattern

</div>

</div>

<div class="">

<label for="">Contact:</label>

<input type="text" name="" class="form-control" formControlName="contact">

</div>

<div \*ngIf="form.get('contact').touched && form.get('contact').invalid" class="alert alert-danger">

<div \*ngIf="form.get('contact').hasError('required')">

Contact is required

</div>

<div \*ngIf="form.get('contact').hasError('pattern') ">

Invalid pattern

</div>

<div \*ngIf="form.get('contact').hasError('minlength') && !form.get('contact').hasError('pattern')">

Less than 3 digits

</div>

<div \*ngIf="form.get('contact').hasError('maxlength') && !form.get('contact').hasError('pattern')">

Greater than 10 digits

</div>

</div>

<div class="">

<label for="">Email:</label>

<input type="text" name="" class="form-control" formControlName="email">

</div>

<div \*ngIf="form.get('email').touched && form.get('email').invalid" class="alert alert-danger">

<div \*ngIf="form.get('email').hasError('required')">

Email is required

</div>

<div \*ngIf="form.get('email').hasError('email') && !form.get('email').hasError('required') ">

Invalid Email

</div>

</div>

<button type="submit" [disabled]="form.invalid" class="btn btn-success"> Add User</button>

</form>

<ul class="list-group">

<li \*ngFor="let user of userList" class="list-group-item">

{{user.name | json}}

</li>

</ul>