setNotification

reactive template with click event binding

<div class="form-group">

<div class="form-check form-check-inline">

<input class="form-check-input"

type="radio"

formControlName = "sendNotification"

id="emailnotification"

(click) = "setNotification('email')"

value="email">

<label class="form-check-label" for="emailnotification">

Email

</label>

</div>

<div class="form-check form-check-inline">

<input class="form-check-input"

type="radio"

formControlName = "sendNotification"

id="textnotification"

(click) = "setNotification('text')"

value="text">

<label class="form-check-label" for="textnotification">

Text

</label>

</div>

</div>

Reactive component.ts file

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

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

import { test } from '../core/service';

function ratingRange(min, max): ValidatorFn {

return function(c: AbstractControl): {[key: string]: boolean} | null {

console.log('rating validation');

if (c.value !== null && (isNaN(c.value) || c.value < min || c.value > max)) {

console.log('rating validation');

return {'range': true};

}

return null;

};

}

function emailMatcher(c:AbstractControl):{[key:string]:boolean} | null{

let confirmEmailControl = c.get('confirmEmail');

let emailControl = c.get('email');

console.log('email matcher',confirmEmailControl.value === emailControl.value);

if(confirmEmailControl.value === emailControl.value){

return null;

}

return {match:true};

}

@Component({

selector: 'app-reactiveform',

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

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

})

export class ReactiveformComponent implements OnInit {

reactiveformComponentInfo:string;

customersForm: FormGroup;

constructor(private fb: FormBuilder) { }

ngOnInit() {

this.reactiveformComponentInfo = 'ngOninit of reactiveformComponentInfo';

console.log(this.reactiveformComponentInfo);

// this.customersForm = new FormGroup({

// firstName: new FormControl('',[Validators.required,Validators.minLength(3)]),

// lastName: new FormControl('',),

// email: new FormControl(''),

// verify: new FormControl(true)

// });

this.customersForm = this.fb.group({

firstName: ['', [Validators.required, Validators.minLength(3)]],

lastName: ['', [Validators.required, Validators.minLength(3)]],

emailGroup:this.fb.group({

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

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

}),

phone: '',

sendNotification: 'email',

rating: ['', ratingRange(1, 5)],

verify: 'true'

});

}

save() {

console.log(this.customersForm);

}

setNotification(notifyVal: string) {

const phoneControl = this.customersForm.get('phone');

if (notifyVal === 'text') {

phoneControl.setValidators(Validators.required);

} else {

phoneControl.clearValidators();

}

phoneControl.updateValueAndValidity();

}

}

**Removing event binding from template**

reactive template after removing event binding

<!-- notification -->

<div class="form-group">

<div class="form-check form-check-inline">

<input class="form-check-input"

type="radio"

formControlName = "sendNotification"

id="emailnotification"

value="email">

<label class="form-check-label" for="emailnotification">

Email

</label>

</div>

<div class="form-check form-check-inline">

<input class="form-check-input"

type="radio"

formControlName = "sendNotification"

id="textnotification"

value="text">

<label class="form-check-label" for="textnotification">

Text

</label>

</div>

</div>

Subscribing to formcontrol value changes in ngOnit of reactiveComponent.ts

ngOnInit() {

this.reactiveformComponentInfo = 'ngOninit of reactiveformComponentInfo';

console.log(this.reactiveformComponentInfo);

// this.customersForm = new FormGroup({

// firstName: new FormControl('',[Validators.required,Validators.minLength(3)]),

// lastName: new FormControl('',),

// email: new FormControl(''),

// verify: new FormControl(true)

// });

this.customersForm = this.fb.group({

firstName: ['', [Validators.required, Validators.minLength(3)]],

lastName: ['', [Validators.required, Validators.minLength(3)]],

emailGroup:this.fb.group({

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

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

}),

phone: '',

sendNotification: 'email',

rating: ['', ratingRange(1, 5)],

verify: 'true'

});

let sendNotificationControl = this.customersForm.get('sendNotification');

sendNotificationControl.valueChanges.subscribe(value =>{

this.setNotification(value);

})

}

save() {

console.log(this.customersForm);

}

setNotification(notifyVal: string) {

const phoneControl = this.customersForm.get('phone');

if (notifyVal === 'text') {

phoneControl.setValidators(Validators.required);

} else {

phoneControl.clearValidators();

}

phoneControl.updateValueAndValidity();

}

}

Moving Validation rules out of the template

<div formGroupName = 'emailGroup'>

<!-- email -->

<div class="form-group">

<label for="exampleInputEmail1">Email address</label>

<input type="email"

formControlName = 'email'

class="form-control"

id="exampleInputEmail1"

[ngClass]="{'is-invalid':(customersForm.get('emailGroup.email').touched && customersForm.get('emailGroup.email').dirty) && !customersForm.get('emailGroup.email').valid}"

placeholder="Enter email">

<span class="invalid-feedback">

<span \*ngIf="customersForm.get('emailGroup.email').errors?.required">

Please enter your email address

</span>

<span \*ngIf="customersForm.get('emailGroup.email').errors?.email">

Please enter a valid email address

</span>

</span>

</div>

<!-- confirmEmail -->

<div class="form-group">

<label for="exampleInputEmail1">Email address</label>

<input type="email"

formControlName = 'confirmEmail'

class="form-control"

id="exampleInputEmail1"

[ngClass]="{'is-invalid': customersForm.get('emailGroup').errors ||

(customersForm.get('emailGroup.confirmEmail').touched &&

customersForm.get('emailGroup.confirmEmail').dirty) &&

!customersForm.get('emailGroup.confirmEmail').valid}"

placeholder="Enter confirmEmail">

<span class="invalid-feedback">

<span \*ngIf="customersForm.get('emailGroup.confirmEmail').errors?.required">

Please enter your confirm address

</span>

<span \*ngIf="customersForm.get('emailGroup').errors?.match">

Email address doesnot match .

</span>

</span>

</div>

</div>

Moving <span class="invalid-feedback"> validation messages to component ts