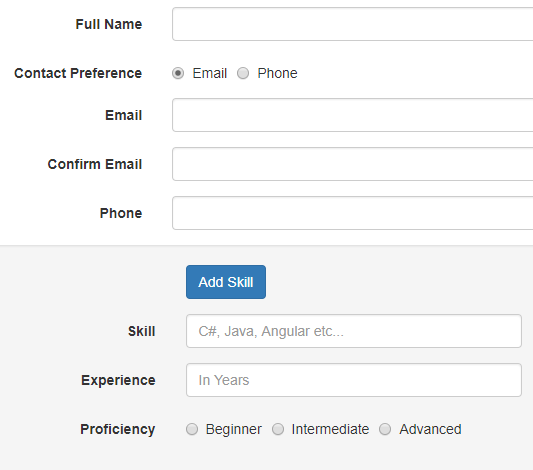
**Lesson20 Creating form array of form group objects in Angular**

**Notes:-**

**1-In the following form, when we click the button Add Skill it will show the two form control Skill, Experience, it will create new radio button**



**Steps:-**

**1-In the employee-comp.ts we write the following code**

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

// Import FormGroup and FormControl classes

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

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

import {emailDomain,emailDomain1,matchEmails} from '../../Validators/email-validators';

import { format } from 'url';

@Component({

selector: 'app-employees-comp',

templateUrl: './employees-comp.component.html',

styleUrls: ['./employees-comp.component.css']

})

export class EmployeesCompComponent implements OnInit {

// Include phone property

formErrors = {

'fullName': '',

'emailGroup':'',

'email': '',

'confirmEmail':'',

'phone': '',

//we declare the following validation binding

'skillName': '',

'experienceInYears': '',

'proficiency': ''};

validationMessages = {

'fullName': {

'required': 'Full Name is required.',

'minlength': 'Full Name must be greater than 2 characters',

'maxlength': 'Full Name must be less than 10 characters.'},

'emailGroup':{

'matchEmailsMsg': 'email and confirm email must match'},

'email': {

'required': 'Email is required.',

'emailDomain': 'Email domian should be hotmail.com',

'emailDomain1': 'Email domian should be prgaim123.com'},

'confirmEmail': {'required': 'Confirm Email is required.'},

'phone': {'required': 'Phone is required.'},

//we declare the following validation messages

'skillName': {'required': 'Skill Name is required.'},

'experienceInYears': {'required': 'Experience is required.'},

'proficiency': {'required': 'Proficiency is required.'}};

employeeForm:FormGroup;

constructor(private fb:FormBuilder){}

ngOnInit() {

// Include FormControls for contactPreference, email & phone

// contactPreference has email as the default value

this.employeeForm = this.fb.group({

fullName: ['', Validators.compose([Validators.required, Validators.minLength(2), Validators.maxLength(10)])],

contactPreference: ['email'],

emailGroup: this.fb.group({

email: ['', Validators.compose([Validators.required,emailDomain('hotmail.com')])],

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

}, { validator: matchEmails }),

phone: [''],

//we declare the function that add the form group into the form array

skills: this.fb.array([this.AddSkillFormGrouping()])});

this.employeeForm.valueChanges.subscribe((data) => {

this.logValidationErrors(this.employeeForm);

});

this.employeeForm.get('contactPreference').valueChanges.subscribe((data) =>{this.onContactPrefernceChange(data);})

}

//when we click add button we call the method

addSkillButtonClick(): void {

(<FormArray>this.employeeForm.get('skills')).push(this.AddSkillFormGrouping());}

//The reason we use the method is that we use it to bind form array and click ADD button   
AddSkillFormGrouping():FormGroup{

return this.fb.group({skillName: ['', Validators.required],

experienceInYears: ['', Validators.required],

proficiency: ['', Validators.required]});}

logValidationErrors(group: FormGroup = this.employeeForm): void {

Object.keys(group.controls).forEach((key: string) => {

const abstractControl = group.get(key);

this.formErrors[key] = '';

if (abstractControl && !abstractControl.valid

&& (abstractControl.touched || abstractControl.dirty)) {

const messages = this.validationMessages[key];

for (const errorKey in abstractControl.errors) {

if (errorKey) {

this.formErrors[key] += messages[errorKey] + ' ';}}}

if (abstractControl instanceof FormGroup) {

this.logValidationErrors(abstractControl);}

// We need this additional check to get to the FormGroup

// in the FormArray and then recursively call this

// logValidationErrors() method to fix the broken validation

if (abstractControl instanceof FormArray) {

for (const control of abstractControl.controls) {

if (control instanceof FormGroup) {

this.logValidationErrors(control);

}}}});}

onContactPrefernceChange(frmGroup:string){

debugger;

const obj = this.employeeForm.get('phone');

if(frmGroup === 'phone'){

obj.setValidators(Validators.required);}

else{obj.clearValidators();}

obj.updateValueAndValidity();}

onLoadDataClick(){

const formArray = this.fb.array([

new FormControl('John',Validators.required),

new FormGroup({

country:new FormControl('',Validators.required)}),

new FormArray([])])}}

**2-In the employee-comp.html we type the following code**

<form class="form-horizontal" [formGroup]="employeeForm" (ngSubmit)="onSubmit()">

<div class="panel panel-primary">

<div class="panel-heading">

<h3 class="panel-title">Create Employee</h3></div>

<div class="panel-body">

<div class="form-group">

<label class="col-sm-2 control-label" for="fullName">Full Name</label>

<div class="col-sm-8">

<input id="fullName" type="text" class="form-control" formControlName="fullName" (blur)="logValidationErrors()">

<span class="help-block" \*ngIf="formErrors.fullName">

{{formErrors.fullName}}</span></div></div>

<div class="form-group">

<label class="col-md-2 control-label">Contact Preference</label>

<div class="col-md-8">

<label class="radio-inline">

<input type="radio" value="email" formControlName="contactPreference">Email</label>

<label class="radio-inline">

<input type="radio" value="phone"formControlName="contactPreference">Phone

</label></div></div>

<div formGroupName="emailGroup">

<!-- Email input element -->

<div class="form-group">

<label class="col-sm-2 control-label" for="email">Email</label>

<div class="col-sm-8">

<input id="email" type="text" class="form-control" formControlName="email" (blur)="logValidationErrors()">

<span class="help-block" \*ngIf="formErrors.email">

{{formErrors.email}}

</span></div></div>

<div class="form-group">

<label class="col-sm-2 control-label" for="confirmEmail">Confirm Email</label>

<div class="col-sm-8">

<input id="confirmEmail" type="text" class="form-control" formControlName="confirmEmail" (blur)="logValidationErrors()">

<span class="help-block" \*ngIf="formErrors.confirmEmail || formErrors.emailGroup">

{{formErrors.confirmEmail ? formErrors.confirmEmail : formErrors.emailGroup}}

</span></div></div></div>

<!-- Phone input element -->

<div class="form-group">

<label class="col-sm-2 control-label" for="phone">Phone</label>

<div class="col-sm-8">

<input id="phone" type="text" class="form-control" formControlName="phone" (blur)="logValidationErrors()">

<span class="help-block" \*ngIf="formErrors.phone">

{{formErrors.phone}}

</span></div></div>

//we create button and apply the addSkillButtonClick() to create new form group to the form array

<div class="form-group">

<div class="col-md-offset-2 col-md-4">

<button type="button" class="btn btn-primary" (click)="addSkillButtonClick()">Add Skill

</button></div></div>

//we create form array that make looping on the skills form array that

//based on the form group and assign unique id -> group number

<div class="well">

<div formArrayName="skills" \*ngFor="let skill of employeeForm.get('skills').controls; let i = index"><div [formGroupName]="i">

<div class="form-group">

//we make dynamic validator ID = ‘'txtskillName'+groupNo

<label class="col-md-2 control-label" [attr.for]="'txtskillName'+i">skill Name</label>

<div class="col-md-8">

//we make dynamic validator ID = ‘'txtskillName'+groupNo

<input [id]="'txtskillName'+i" type="text" class="form-control" formControlName="skillName" (blur)="logValidationErrors()">

<span class="help-block" \*ngIf="formErrors.skillName">

{{formErrors.skillName}}</span></div></div>

<div class="form-group">

//we make dynamic validator ID = 'txtexperienceInYears'+groupNo

<label class="col-md-2 control-label" [attr.for]="'txtexperienceInYears'+i">experience In Years</label>

//we make dynamic validator ID = 'txtexperienceInYears'+groupNo

<div class="col-md-8">

<input [id]="'txtexperienceInYears'+i" type="text" class="form-control" formControlName="experienceInYears" (blur)="logValidationErrors()">

<span class="help-block" \*ngIf="formErrors.experienceInYears">

{{formErrors.experienceInYears}}</span></div></div>

<div class="form-group">

<label class="col-md-2 control-label" >Proficiency</label>

<div class="col-md-8">

<label class="radio-inline">

<input type="radio" value="email" formControlName="proficiency">Beginner

</label>

<label class="radio-inline">

<input type="radio" value="phone" formControlName="proficiency">Intermediate

</label>

<label class="radio-inline">

<input type="radio" value="phone" formControlName="proficiency">Advance

</label>

<span class="help-block" \*ngIf="formErrors.proficiency">

{{formErrors.proficiency}}</span></div></div>

</div></div></div></div>

<div class="panel-footer">

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

<button class="btn btn-primary" type="button" (click)="onLoadDataClick()">Load Data</button></div>

</div>

</form>