**Lesson04 Angular 6 Reactive Form**

**Notes:-**

**1-There are 2 ways to create forms in Angular: -**

**A-Template Driven Forms B-Reactive Forms (Also called Model Driven Forms)**

**2-Template Driven Forms are heavy on the template meaning we create the form completely in HTML. Template driven forms are easy to build and understand. They are great for creating simple forms. However, creating complex forms using template driven approach is not recommended as the HTML can get very complicated and messy. It is not easy to unit test template forms as the logic is in the HTML.**

**3-Reactive forms on the other hand allow us to build the form completely in code.**

**A-This is more flexible and has many benefits over template forms**

**B-Make Dynamic Validation More easily and make unit testing**

**Syntax of the FormGroup**

constructor(controls: { [key: string]: AbstractControl;},

validatorOrOpts?: ValidatorFn | ValidatorFn[] | AbstractControlOptions | null,

asyncValidator?: AsyncValidatorFn | AsyncValidatorFn[] | null);

**1-The First parameter is primary and the 2 parameter are optional**

**The controls parameter is used to pass the collection of child controls.**

**Which means that the controls define in the form group**

**Syntax of the FormControl**

constructor(controls: { [key: string]: AbstractControl;}

**Note:-**

**1-Both Form Control and Form Group classes inherit from Abstract Control class**

**2-Form Group can have a nested Form Groups**

**Steps:-**

**1-in the app.module.ts we set the following modules**

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

imports: [

ReactiveFormsModule]

**2-in the employee-react.comp.ts we set the following**

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

// Import FormGroup and FormControl classes

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

@Component({

selector: 'app-employee-react-comp',

templateUrl: './employee-react-comp.component.html',

styleUrls: ['./employee-react-comp.component.css']

})

export class EmployeeReactCompComponent implements OnInit {

employeeForm: FormGroup;

constructor() { }

ngOnInit() {

//we initialize the form controls in the form group

this.employeeForm = new FormGroup({

fullName: new FormControl(),

email: new FormControl()

});}

onSubmit(): void {

console.log(this.employeeForm.value);}}

**3-in the employee-react.comp.html we set the following code**

//we see that we apply the form Group to variable of type form Group

<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">

</div>

</div>

<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">

</div>

</div>

</div>

<div class="panel-footer">

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

</div>

</div>

</form>