## Reactive Forms

A form building technology for angular Applications

Uses the **ReativeFormsModule** from **@angular/forms Library**

With reactive forms, you create a tree of Angular form control objects in the component class and bind them to native form control elements in the component template,

Advantage is that value and validity updates are [always synchronous and under your control](https://angular.io/guide/reactive-forms#async-vs-sync)

**Steps to work with reactive forms**

1. Import ReactiveFormsModule and FormsGroup at appModule.ts
2. Create a FormGroup at .ts file and instantiate the controls by using formControlName
3. formControlName at .ts file should match the formControlName at html file
4. use formBuilder for creating the controls easily
   1. if Using FormBuilder Instantiate the control of the formGroup using fromBuilder
5. all validations can be checked by using formGroup.valid
6. all the values can get by using formGroup.value | json or formGroup.value.Controlname

Example

Add a employee class in models folder

export class employee{

public eid : number;

public ename : string;

public city : string;

public salary : number ;

constructor(){

}

}

Create a component employee component in components folder

**At appModule.ts**

import { ReactiveFormsModule, FormGroup } from '@angular/forms'

@NgModule({

imports: [

BrowserModule,

ReactiveFormsModule

],

})

**At Employee Componenet.ts**

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

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

export class EmployeeComponent implements OnInit {

private frmgrp: FormGroup;

private idv: string;

private namev : string;

constructor(private fb: FormBuilder) {

}

frmSubmit() {

if (this.frmgrp.valid) {

alert('From is valid...')

this.idv = this.frmgrp.value.id

this.namev = this.frmgrp.value.name;

}

else {

this.idv = this.frmgrp.value.id

this.namev = this.frmgrp.value.name;

}

}

ngOnInit() {

this.frmgrp = this.fb.group({

id: ['', Validators.required],

name: ['', Validators.required],

city: ['', Validators.required],

salary: ['', Validators.required]

})

}

}

**At Employee.html template**

<form [formGroup] ="frmgrp" >

<table>

<tr>

<td>Id : </td>

<td><input type="text" formControlName="id" /></td>

<td><p \*ngIf="idv===undefined" >Id is required</p></td>

</tr>

<tr>

<td>Name : </td>

<td><input type="text" formControlName="name" /></td>

<td><p \*ngIf="namev===undefined" >Name is required</p></td>

</tr>

<tr>

<td>City : </td>

<td><input type="text" formControlName="city" /></td>

</tr>

<tr>

<td>Salary : </td>

<td><input type="text" formControlName="salary" /></td>

</tr>

<tr>

<td colspan="2" align="center">

<button (click)="frmSubmit()" >Submit</button>

</td>

</tr>

</table>

<p>

{{frmgrp.value|json}}

</p>

</form>