Contents

* Forms
* Angular Material



**Forms**

* **Types of Angular Forms?**
  1. **Template-Driven Forms**
     + Form tag is used to create forms.
     + Angular automatically interprets and creates a form object representation for the tag.
     + Controls can be added to the form using the **NGModel** tag.
     + Form value can be generated using the **form.value** object.
     + Template-Driven form use when unstructured data.
  2. **Reactive Forms**
     + Create inside component class so it is also referred as model driven forms.
     + Every form control will have an object in the component and this provides greater control and flexibility in form programming.
     + This form is based on structured data model.
* **Template Driven Forms**
  1. **Configure Forms**
     + First, we need to import **FormsModule** in **app.module.ts**

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

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

**//import FormsModule here**

**import {FormsModule} from '@angular/forms';**

**imports: [**

**BrowserModule,**

**AppRoutingModule,**

**FormsModule //Assign FormsModule**

**],**

* + - Add the below code in .comonent.html file

**<form #userForm=” ngForm” (ngSubmit)=” onClickSubmit (userForm)”>**

**<input type=”text” name=”username” placeholder=”username” ngModel />**

**<br/>**

**<br/>**

**<input type=”submit” value=”Submit” />**

**</form>**

* + - Create **onClickSubmit ()** method inside .component**.ts** file as shown below.

**export class TestComponent implements OnInit {**

**ngOnInit () {}**

**onClickSubmit(result: NgForm) {**

**console.log ("You have entered: ");**

**console.log(‘result.value’);**

**}**

**}**

* **Reactive forms**
  1. **FormControl**
     + Define basic functionality of individual form control
  2. **FormGroup**
     + Used to aggregate the values of collection form control.
  3. **FormArray**
     + Used to aggregate the values of form control into an array.
  4. **Configure forms**
     + First, we need to import **ReactiveFormsModule** in **app.module.ts**
     + Inside .component.ts file as shown below.

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

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

**@Component ({**

**selector: 'app-test',**

**templateUrl: './test.component.html',**

**styleUrls: ['./test.component.css']**

**})**

**export class TestComponent implements OnInit {**

**userName;**

**formData: FormGroup;**

**ngOnInit () {**

**this. formdata = new FormGroup (**

**{userName: new FormControl("Tutorialspoint") }**

**);**

**}**

**onClickSubmit(data) {**

**this. userName = data. userName;**

**}**

**}**

* + - Created an instance of **formGroup** and set it to local variable, formData.
    - Created an instance of **FormControl** and set it one of the entry in formData.
    - Created a **onClickSubmit ()** method, which sets the local variable, **userName** with its argument.
    - Below code in **.component.html**

**<div>**

**<form [FormGroup]=” formData” (ngSubmit)=” onClickSubmit (formData.value)” >**

**<input type=”text” formControlName=” userName” /> <br/>**

**<br/>**

**<input type=” submit” value=” Click here”/> </form> </div> <p> Text Box result is: {{userName}} </p>**

* **Form Validation**

**Angular Material**

**Steps:**

1. Match CLI Version



1. Command for install

**ng add @angular/material**

1. Add separate modules for each component in app.module.ts.