**Angular - Forms**

Forms are used to handle user input data. Angular 8 supports two types of forms. They are **Template driven forms** and **Reactive forms**. This section explains about Angular 8 forms in detail.

## **Template driven forms**

Template driven forms is created using directives in the template. It is mainly used for creating a simple form application. Let’s understand how to create template driven forms in brief.

### **Configure Forms**

Before understanding forms, let us learn how to configure forms in an application. To enable template driven forms, first we need to import **FormsModule** in **app.module.ts.** It is given below −

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

],

Once, **FormsModule** is imported, the application will be ready for form programming.

### **Create simple form**

Let us create a sample application **(template-form-app)** in Angular 8 to learn the template driven form.

Open command prompt and create new Angular application using below command −

cd /go/to/workspace

ng new template-form-app

cd template-form-app

Configure **FormsModule** in **AppComponent** as shown below −

...

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

@NgModule({

declarations: [

AppComponent,

TestComponent

],

imports: [

BrowserModule,

FormsModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

Create a test component using Angular CLI as mentioned below −

ng generate component test

The above create a new component and the output is as follows −

CREATE src/app/test/test.component.scss (0 bytes)

CREATE src/app/test/test.component.html (19 bytes)

CREATE src/app/test/test.component.spec.ts (614 bytes)

CREATE src/app/test/test.component.ts (262 bytes)

UPDATE src/app/app.module.ts (545 bytes)

Let’s create a simple form to display user entered text.

Add the below code in **test.component.html** file as follows −

<form #userName="ngForm" (ngSubmit)="onClickSubmit(userName.value)">

<input type="text" name="username" placeholder="username" ngModel>

<br/>

<br/>

<input type="submit" value="submit">

</form>

Here, we used **ngModel** attribute in **input** text field.

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

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

@Component({

selector: 'app-test',

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

styleUrls: ['./test.component.scss']

})

export class TestComponent implements OnInit {

ngOnInit() {

}

onClickSubmit(result) {

console.log("You have entered : " + result.username);

}

}

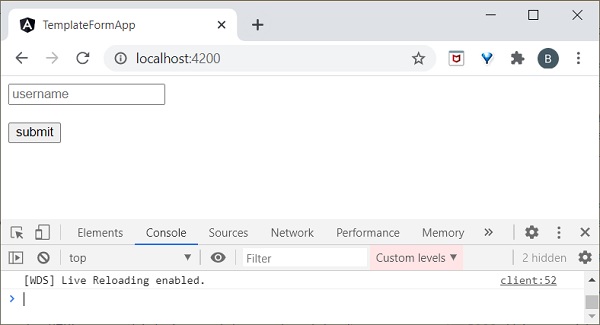
Open app.component.html and change the content as specified below −

<app-test></app-test>

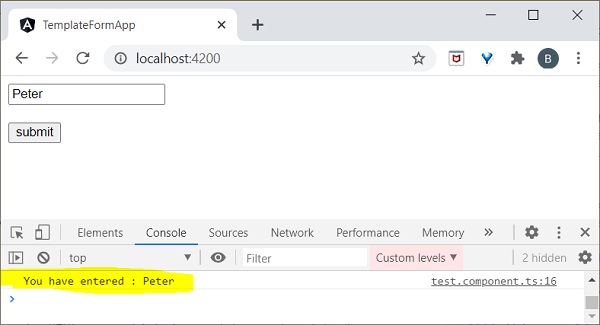
Finally, start your application (if not done already) using the below command −

ng serve

Now, run your application and you could see the below response −



Enter **Peter** in input text field and enter submit. **onClickSubmit** function will be called and user entered text **Peter** will be send as an argument. **onClickSubmit** will print the user name in the console and the output is as follows −



## **Reactive Forms**

**Reactive Forms** is created 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 the form programming. **Reactive Form** is based on structured data model. Let’s understand how to use Reactive forms in angular.

### **Configure Reactive forms**

To enable reactive forms, first we need to import **ReactiveFormsModule** in **app.module.ts**. It is defined below

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

import { NgModule } from '@angular/core';

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

import { AppComponent } from './app.component';

import { TestComponent } from './test/test.component';

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

//import ReactiveFormsModule here

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

imports: [

BrowserModule,

AppRoutingModule,

FormsModule,

ReactiveFormsModule //Assign here

]

### **Create Reactive forms**

Before moving to create Reactive forms, we need to understand about the following concepts,

* **FormControl** − Define basic functionality of individual form control
* **FormGroup** − Used to aggregate the values of collection form control
* **FormArray** − Used to aggregate the values of form control into an array
* **ControlValueAccessor** − Acts as an interface between Forms API to HTML DOM elements.

Let us create a sample application **(reactive-form-app)** in Angular 8 to learn the template driven form.

Open command prompt and create new Angular application using below command −

cd /go/to/workspace

ng new reactive-form-app

cd reactive-form-app

Configure **ReactiveFormsModule** in **AppComponent** as shown below −

...

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

@NgModule({

declarations: [

AppComponent,

TestComponent

],

imports: [

BrowserModule,

ReactiveFormsModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

Create a **test** component using Angular CLI as mentioned below −

ng generate component test

The above create a new component and the output is as follows −

CREATE src/app/test/test.component.scss (0 bytes)

CREATE src/app/test/test.component.html (19 bytes)

CREATE src/app/test/test.component.spec.ts (614 bytes)

CREATE src/app/test/test.component.ts (262 bytes)

UPDATE src/app/app.module.ts (545 bytes)

Let’s create a simple form to display user entered text.

We need to import **FormGroup, FormControl** classes in **TestComponent**.

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

Create **onClickSubmit()** method inside **test.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;

ngOnInit() {

this.formdata = new FormGroup({

userName: new FormControl("Tutorialspoint")

});

}

onClickSubmit(data) {this.userName = data.userName;}

}

Here,

* Created an instance of **formGroup** and set it to local variable, formdata.
* Crete 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.

Add the below code in **test.component.html** file.

<div>

<form [formGroup]="formdata" (ngSubmit)="onClickSubmit(formdata.value)" >

<input type= text" name="userName" placeholder="userName"

formControlName = "userName">

<br/>

<br/>

<input type="submit" value="Click here">

</form>

</div>

<p> Textbox result is: {{userName}} </p>

Here,

* New form is created and set it’s **formGroup** property to formdata.
* New input text field is created and set is **formControlName to username.**
* **ngSubmit** event property is used in the form and set onClickSubmit() method as its value.
* **onClickSubmit()** method gets formdata values as its arguments.

Open **app.component.html** and change the content as specified below −

<app-test></app-test>

Finally, start your application (if not done already) using the below command −

ng serve