



[REACTIVE FORM]

What is Reactive Form ?

In this Article we see about Reactive Form, uses of Reactive form, Necessary Things also we can apply some basic validations on reactive form. What is ReactiveFormsModule? Where should can we apply it? Difference between Template driven form and Reactive form.

Reactive Forms

Q.1) What is Reactive Forms? When we use Reactive Form?

- If we have a requirement to create a **Complex form** that time we use Reactive form. It is Also known as Modern-Driven Form.
- Reactive forms are more Robust (Strong).
- Reactive form does **not support Two way data binding**.
- Reactive form is **Synchronous**.
- Synchronous means execution of code **line by line**.
- When first statement executes successfully after that second statement will be execute.
- Main thing is Most of the code we write in **TypeScript side**.

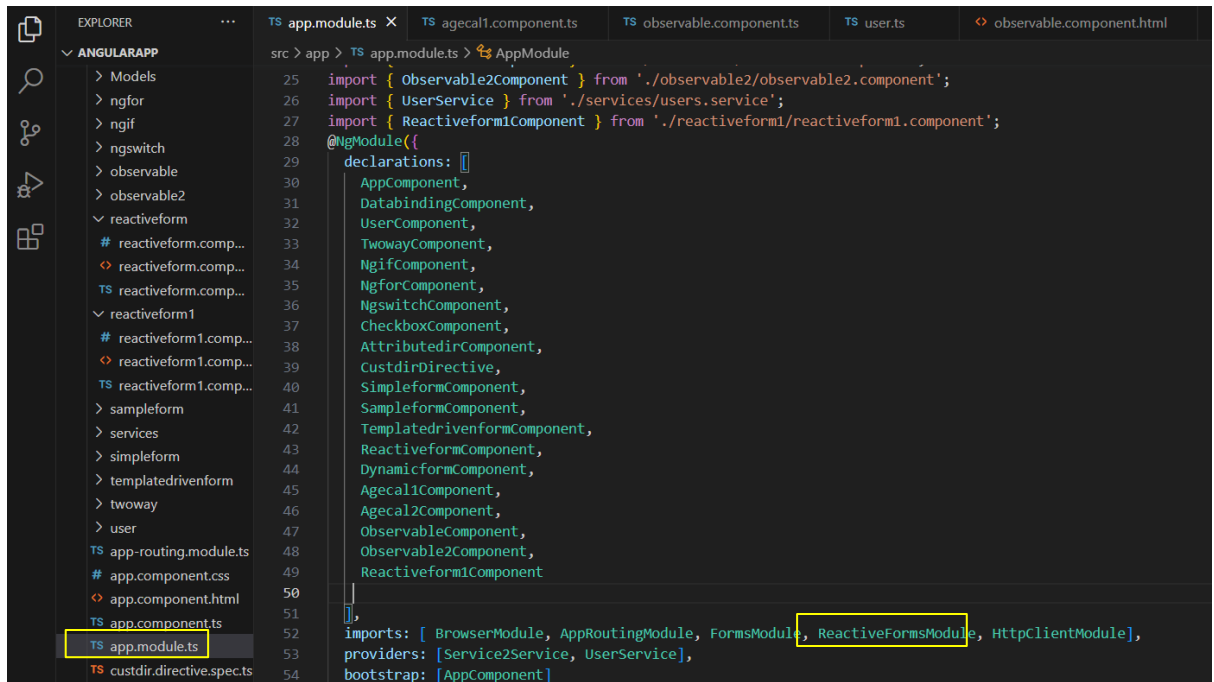
Q.2) Difference between Template Driven form and Reactive form.

Template Driven Form	Reactive Form
Easier to create Template Driven Form	Complex to create Reactive Form
Most of the code we write in HTML Side	Most of the code we write in TypeScript side.
Form Structure and Logic is mainly implemented in HTML	Form Structure and Logic is mainly implemented in TypeScript.
Support Two Way DataBinding [(ngModel)]	Does Not Support Two Way DataBinding formControlName
Need to Implement FormsModule in imports array → app.module.ts	Need to Implement ReactiveFormsModule in imports array → app.module.ts
It is Asynchronous	It is Synchronous.
Apply Validations in HTML side.	Apply validations in TypeScript Side.
Complex for Unit Testing.	Easier for Unit Testing
Not Suitable for creation of forms with dynamic structure at run time	Suitable for creation of forms with dynamic structure at run time

Reactive Forms

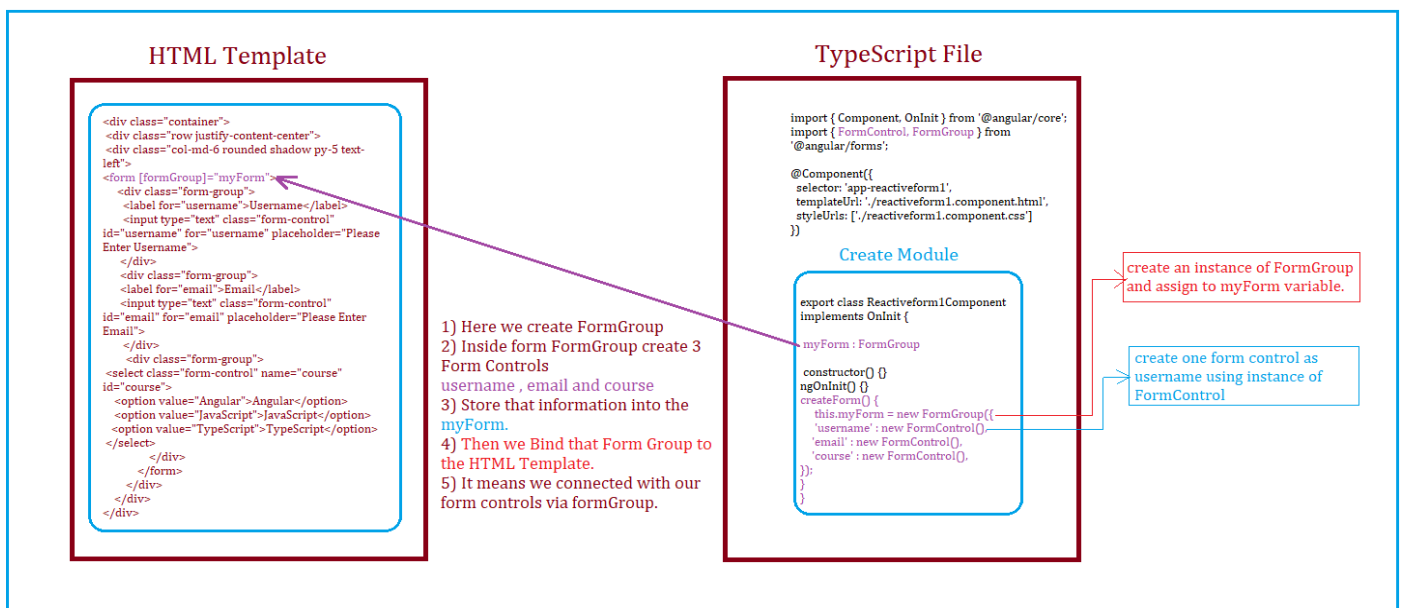
Q.3) Execution flow of Reactive form.

- First we need to import **ReactiveFormsModule**
- **app.module.ts** → **@NgModule** directive → **imports** [ReactiveFormsModule]
- **@NgModule** directive provides all the metadata to the components.
- **imports :[]** → get the properties of third party modules inside angular framework.



```
src > app > TS app.module.ts X TS agecal1.component.ts TS observable.component.ts TS user.ts observable.component.html
EXPLORER
ANGULARAPP
  Models
  ngfor
  ngif
  ngswitch
  observable
  observable2
  reactiveform
    # reactiveform.comp...
    < reactiveform.comp...
    TS reactiveform.comp...
  reactiveform1
    # reactiveform1.comp...
    < reactiveform1.comp...
    TS reactiveform1.comp...
  sampleform
  services
  simpleform
  templatedrivenform
  twoway
  user
  TS app-routing.module.ts
  # app.component.css
  < app.component.html
  TS app.component.ts
  TS custdir.directive.spec.ts
TS app.module.ts
25 import { Observable2Component } from './observable2/observable2.component';
26 import { UserService } from './services/users.service';
27 import { Reactiveform1Component } from './reactiveform1/reactiveform1.component';
28 @NgModule({
29   declarations: [
30     AppComponent,
31     DatabindingComponent,
32     UserComponent,
33     TwowayComponent,
34     NgifComponent,
35     NgforComponent,
36     NgswitchComponent,
37     CheckboxComponent,
38     AttributedirComponent,
39     CustdirDirective,
40     SimpleformComponent,
41     SampleformComponent,
42     TemplatedrivenformComponent,
43     ReactiveformComponent,
44     DynamicformComponent,
45     Agecal1Component,
46     Agecal2Component,
47     ObservableComponent,
48     Observable2Component,
49     Reactiveform1Component
50   ],
51   imports: [ BrowserModule, AppRoutingModule, FormsModule, ReactiveFormsModule, HttpClientModule ],
52   providers: [ Service2Service, UserService ],
53   bootstrap: [ AppComponent ]
54 })
```

- Create one Component using CLI
- Now we are creating a FormGroup so we need to import **FormGroup**.
- **FormGroup** allow us to create and manage groups.
- Now we create one module in TypeScript File and Bind that form to HTML Template.



Reactive Forms

- In above diagram we can see we have create one module for creating FormGroup.
- Inside Module define one variable **myForm :FormGroup** having type FormGroup.
- Then we create one method **createForm ()**.
- Inside createForm method call the of **myForm** we use **this.myForm**
- Then we Create a FormGroup instance using **new** keyword and assign that instance to myForm variable.
- For creating FormControl we use **FormControl()** method
- **FormControl()** → allow you to create new formControl. (e.g. input, select)
- Like this we created three FormControls username, email and course.
- Also we created one form in HTML Template having same form controls but using DOM elements.
- Now we need to bind them.
- So for binding we use **property binding []** and assign myForm to form element.
- **<form [formGroup]="myForm">**
- Then angular form how to know which property is assign for which formControl.
- For that we use **formControlName** directive
- **formControlName** → allowing you to bind form controls to HTML elements.
- **in template driven form we use name attribute**
- **in reactive form we use formControlName directive**
- **FormGroup** → **[formGroup] = "myForm"** (Bind FormGroup)
- **FormControl** → **formControlName = "username"** (Bind FormControl)
- Now need to send data to console we can use **ngSubmit()** and pass a method which is created in ts file. (OnSubmit)
- Here we perform click event so we need to use **event binding()**.
- **<form [formGroup]="myForm" (ngSubmit)="OnSubmit">**
- Now assign FormControl to form element will use **this.createForm** in constructor.
- Because Wherever your DOM elements render on browser then it will process your TypeScript file. Inside ts file constructor will execute first. So our formControl will execute at the run time.
- After click on submit button my OnSubmit method will be call and in response
- I'm getting response on console.
- See the Second Diagram as below.

Reactive Forms

Point number 4

HTML Template

```
<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-6 rounded shadow py-5 text-left">
      <form [formGroup]="myForm"
      (ngSubmit)="onSubmit()">
        <div class="form-group">
          <label for="username">Username</label>
          <input type="text" class="form-control"
            formControlName="username">
        </div>
        <div class="form-group">
          <label for="email">Email</label>
          <input type="text" class="form-control"
            formControlName="email">
        </div>
        <div class="form-group mt-2">
          <label for="course">Course</label>
          <select class="form-control" id="course"
            formControlName="course">
            <option value="Angular">Angular</option>
            <option value="JavaScript">JavaScript</option>
            <option value="TypeScript">TypeScript</option>
          </select>
        </div>
        <button type="submit" class="btn btn-success mt-2">
          Submit</button>
        </div>
      </div>
    </div>
  </div>
```

TypeScript File

```
import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup } from
  '@angular/forms';
@Component({
  selector: 'app-reactiveform1',
  templateUrl: './reactiveform1.component.html',
  styleUrls: ['./reactiveform1.component.css']
})
export class Reactiveform1Component implements
  OnInit {
  myForm : FormGroup
  constructor() {
    this.createForm();
  }
  ngOnInit() {
  }
  onSubmit() {
    console.log("Submit method called");
  }
  createForm() {
    this.myForm = new FormGroup({
      'username': new FormControl(),
      'email': new FormControl(),
      'course': new FormControl()
    })
  }
}
```

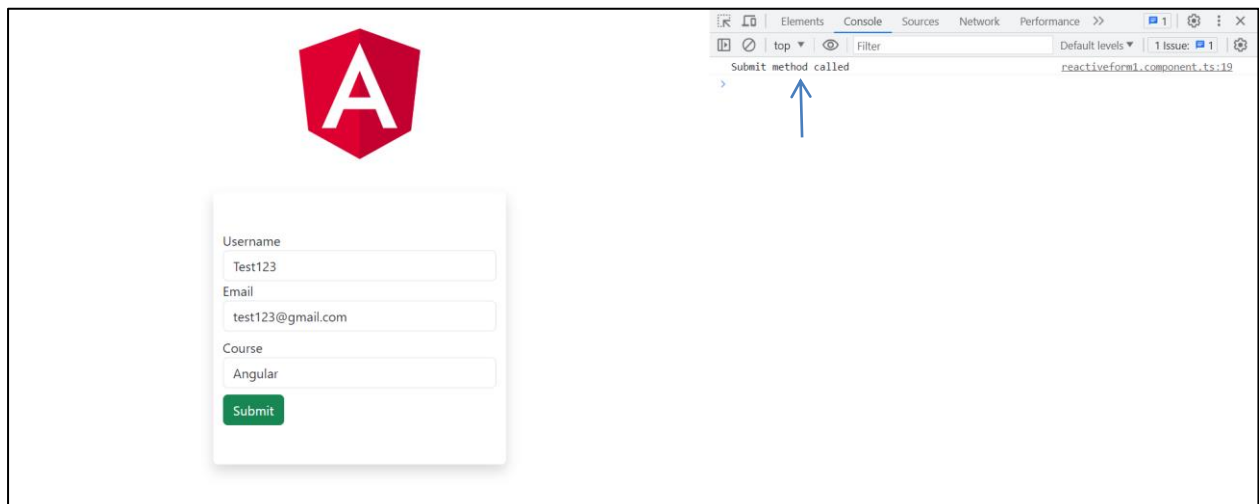
1) We have create a FormGroup.
2) Create one method **createForm**.
3) inside that method create an instance of **FormGroup** and assign to **myForm**.
4) Then Bind **myForm** to HTML form.
5) For that we use Property Binding [].
6) [formGroup] = "**myForm**"
7) Assign all properties to HTML form.
8) But how to know username property is for username formControl ?
9) For that we use **formControlName**.
10) **formControlName="username"**
11) Now HTML template form is know which formControl instance is use for particular formControl.
12) When u click on Submit Button your **onSubmit** method will call.
13) and We assign one method to it.
14) angular will call that method also.
15) and then "**Submit method called**" log on console.

Assign instance of **createForm** method to constructor. Because constructor will call first when we run program. So Constructor have instance of **FormGroup** and **FormControls**.

create an instance of **FormGroup** and assign to **myForm** variable.

create one form control as **username** using instance of **FormControl**

We can see the output like we get message "Submit method called"



4) Important point :

- Need to import **ReactiveFormsModule** in imports array.
- Need to import **FormGroup** and **FormControl** in **component.ts**
- If we have complex form then we use **Reactive form**.
- **formControlName** → used to bind **FormControl** instance to DOM element.
- **FormGroup** → Collection of Form Controls and used to create instance of **FormGroup**
- **FormControl** → used to handle the user input and create new instance of **FormControl**
- **ngSubmit** → is a event, When user click on Submit button after that **ngSubmit** will call.