**Reactive Forms**

Reactive forms are forms where we define the structure of the form in the component class.

we create the form model with Form Groups, Form Controls, and Form Arrays. We also define the validation rules in the component class.

Then, we bind it to the HTML form in the template.

**How to use Reactive Forms**

1. Import ReactiveFormsModule
2. Create Form Model in component class using Form Group, Form Control & Form Arrays
3. Create the HTML Form resembling the Form Model.
4. Bind the HTML Form to the Form Model.

**Import ReactiveFormsModule**

The ReactiveFormsModule contains all the form directives and constructs for working with angular reactive forms.

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule

  ]

**Initial template**

<div class="container">

    <span><h2>{{ title }}</h2></span>

    <div class="form">

        <form>

            <p>

                <label for="bookName">Book Name</label>

                <input type="text" id="bookName" name="bookName">

            </p>

            <p>

                <label for="authorName">Author</label>

                <input type="text" id="authorName" name="authorName">

            </p>

            <p>

                <label for="bookPrice">Price </label>

                <input type="text" id="bookPrice" name="bookPrice">

            </p>

            <p>

                <label for="publisher">Publisher </label>

                <input type="text" id="publisher" name="publisher">

            </p>

            <p>

                <label for="genre">Genre</label>

                <input type="radio" value="fiction" id="genre" name="genre"> Fiction

                <input type="radio" value="science" id="genre" name="genre"> Science

                <input type="radio" value="novel" id="genre" name="genre"> Novel

                <input type="radio" value="self help" id="genre" name="genre"> Sefl Help

                <input type="radio" value="autobiography" id="genre" name="genre"> Autobiography

                <input type="radio" value="technology" id="genre" name="genre"> Technology

            </p>

            <div formGroupName = "publisher\_contact">

                <p>

                  <label for="city">City</label>

                  <input type="text" name="city" >

                </p>

                <p>

                  <label for="phone">Phone</label>

                  <input type="text" name="phone" >

                </p>

                <p>

                  <label for="email">email</label>

                  <input type="text" name="email" >

                </p>

            </div>

            <p>

                <button type="submit">Submit</button>

            </p>

        </form>

    </div>

</div>

1. Template is associated with model with the help of formGroup directive in the following manner
2. [In Reactive](https://www.tektutorialshub.com/angular/angular-formarray-example-in-reactive-forms/) Forms approach, It is our responsibility to build the Model using FormGroup, FormControl and FormArray.
3. FormControl encapsulates the state of a *single form element*
4. FormGroup represents a collection of form Controls.
5. It can also contain form groups and form arrays.
6. Angular Form is a FormGroup.

Code for the Component

Syntax to create the FormGroup

bookForm = new FormGroup({})

The FormGroup takes 3 arguments. a collection of a child FormControl, a validator, and an asynchronous validator. The validators are optional.

bookForm = new FormGroup({

    bookName: new FormControl(),

    authorName: new FormControl(),

    bookPrice: new FormControl(),

    genre: new FormControl(),

    publisher: new FormControl(),

    publisher\_contact: new FormGroup({

      city: new FormControl(),

      phone: new FormControl(),

      email: new FormControl()

    })

  });

We need to associate the model with the template

<form [formGroup]="bookForm">

We also need to bind form fields with the FormControl models.

We can use formControlName directive and add it to each field in the html form.

< input type="text"

id="bookName" name="bookName"

formControlName="bookName" >

**Submit Form**

1. We submit the form using ngSubmit directive.
2. ngSubmit directive binds itself to the click event of the submit button.

<form [formGroup]="bookForm" (ngSubmit) = "onSubmit()">

**Final Template**

Following is how the final template looks like.

<div class="container">

    <span><h2>{{ title }}</h2></span>

    <div class="form">

        <form [formGroup]="bookForm" (ngSubmit) = "onSubmit()">

            <p>

                <label for="bookName">Book Name</label>

                <input type="text" id="bookName" name="bookName" formControlName="bookName">

            </p>

            <p>

                <label for="authorName">Author</label>

                <input type="text" id="authorName" name="authorName" formControlName="authorName">

            </p>

            <p>

                <label for="bookPrice">Price </label>

                <input type="text" id="bookPrice" name="bookPrice" formControlName="bookPrice">

            </p>

            <p>

                <label for="publisher">Publisher </label>

                <input type="text" id="publisher" name="publisher" formControlName="publisher">

            </p>

            <p>

                <label for="genre">Genre</label>

                <input type="radio" value="fiction" id="genre" name="genre"  formControlName="genre"> Fiction

                <input type="radio" value="science" id="genre" name="genre"  formControlName="genre"> Science

                <input type="radio" value="novel" id="genre" name="genre"  formControlName="genre"> Novel

                <input type="radio" value="self help" id="genre" name="genre"  formControlName="genre"> Sefl Help

                <input type="radio" value="autobiography" id="genre" name="genre"  formControlName="genre"> Autobiography

                <input type="radio" value="technology" id="genre" name="genre"  formControlName="genre"> Technology

            </p>

            <div formGroupName = "publisher\_contact">

                <p>

                  <label for="city">City</label>

                  <input type="text" name="city"  formControlName="city" >

                </p>

                <p>

                  <label for="phone">Phone</label>

                  <input type="text" name="phone"  formControlName="phone" >

                </p>

                <p>

                  <label for="email">email</label>

                  <input type="text" name="email"  formControlName="email" >

                </p>

            </div>

            <p>

                <button type="submit">Submit</button>

            </p>

        </form>

    </div>

</div>

Adding onSubmit() method to the component.

onSubmit(){

     console.log(this.bookForm.value);

   }

# **Using FormBuilder**

1. FormBuilder API makes it easier to build reactive forms.
2. Provides shortcuts to create the instance of the FormControl, [FormGroup](https://www.tektutorialshub.com/angular/formgroup-in-angular/) or FormArray.
3. Reduces the code required to write the complex forms.
4. We need to import formBuilder and inject it into the Component

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

constructor(private fb: FormBuilder)

1. we will use group() method to build Form Group.
2. We can pass Form Controls, or other Form Group as a key-value pair.
3. Key is the name of formControl, formGroup
4. Value is the configuration of the control

bookForm: FormGroup;

  constructor(private fb: FormBuilder) {

    this.bookForm = this.fb.group({

    bookName: ['', [Validators.required]],

    authorName: ['',[Validators.required]],

    bookPrice: ['',[Validators.required]],

    genre: ['',[Validators.required]],

    publisher: ['',[Validators.required]],

    publisher\_contact: this.fb.group({

      city: ['',[Validators.required]],

      phone: ['',[Validators.required]],

      email: ['',[Validators.required,Validators.email]]

    })

    });