



CJC

Complete Java Classes

by **Kunal Sir**
Forms

Angular

❖ What is forms in Angular?

Angular forms are used to log in, update a profile, enter sensitive information, and perform many other data-entry tasks.

❖ Types of Angular Forms:-

1. Template-driven Forms.
2. Reactive Forms.

❖ Template Driven Forms:-

Template-driven forms use two-way data binding to update the data model in the component as changes are made in the template and vice versa.

Template-driven forms are very straightforward to implement in app. But they are not scale as well as reactive forms. Because reactive forms have lot of features as compare to template-driven forms.

All tho, you can include template-driven form in your app if you have very basic requirements.

Example on Template-driven forms:-

1. Create component using command: `ng g c Template-driven --skip-test`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS E:\Angular Workspace\AngularForms> ng g c Template-driven --skip-tests
```



CJC

Complete Java Classes

by Kunal Sir

Angular

Forms

2. Add the component selector in app.component.html file.

```
app.component.html M X
src > app > app.component.html > app-template-driven
Go to component
1 <app-template-driven></app-template-driven>
```

3. Import FormsModule in app.module.ts

```
app.module.ts M X
src > app > app.module.ts > AppModule
4 import { AppComponent } from './app.component';
5 import { TemplateDrivenComponent } from './template-driven/template-driven.component';
6 import { FormsModule } from '@angular/forms';
7
8 @NgModule({
9   declarations: [
10     AppComponent,
11     TemplateDrivenComponent
12   ],
13   imports: [
14     BrowserModule,
15     FormsModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }
21
```



by Kunal Sir

Angular

Forms

4. In `TemplateDrivenComponent` define on template driven form, also create on function to log the value of form properties.

```
template-driven.component.ts U X
src > app > template-driven > template-driven.component.ts > TemplateDrivenComponent
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-template-driven',
5    templateUrl: './template-driven.component.html',
6    styleUrls: ['./template-driven.component.css']
7  })
8  export class TemplateDrivenComponent {
9
10   login={
11     username:'',
12     password:''
13   }
14
15   onLogin()
16   {
17     console.log(this.login.username);
18     console.log(this.login.password);
19   }
20 }
21
```

5. Bind the properties of login form into html using two-way-data binding.

```
template-driven.component.html U X
src > app > template-driven > template-driven.component.html > form > br
Go to component
1  <p>template-driven works!</p>
2  <form (ngSubmit)="onLogin()">
3    <label for="un">Enter Username:-</label>
4    <input type="text" id="un" [(ngModel)]="login.username" name="username">
5    <br>
6    <label for="ps">Enter Username:-</label>
7    <input type="text" id="ps" [(ngModel)]="login.password" name="password">
8    <br>
9    <button type="submit">LOGIN</button>
10 </form>
11
```



CJC

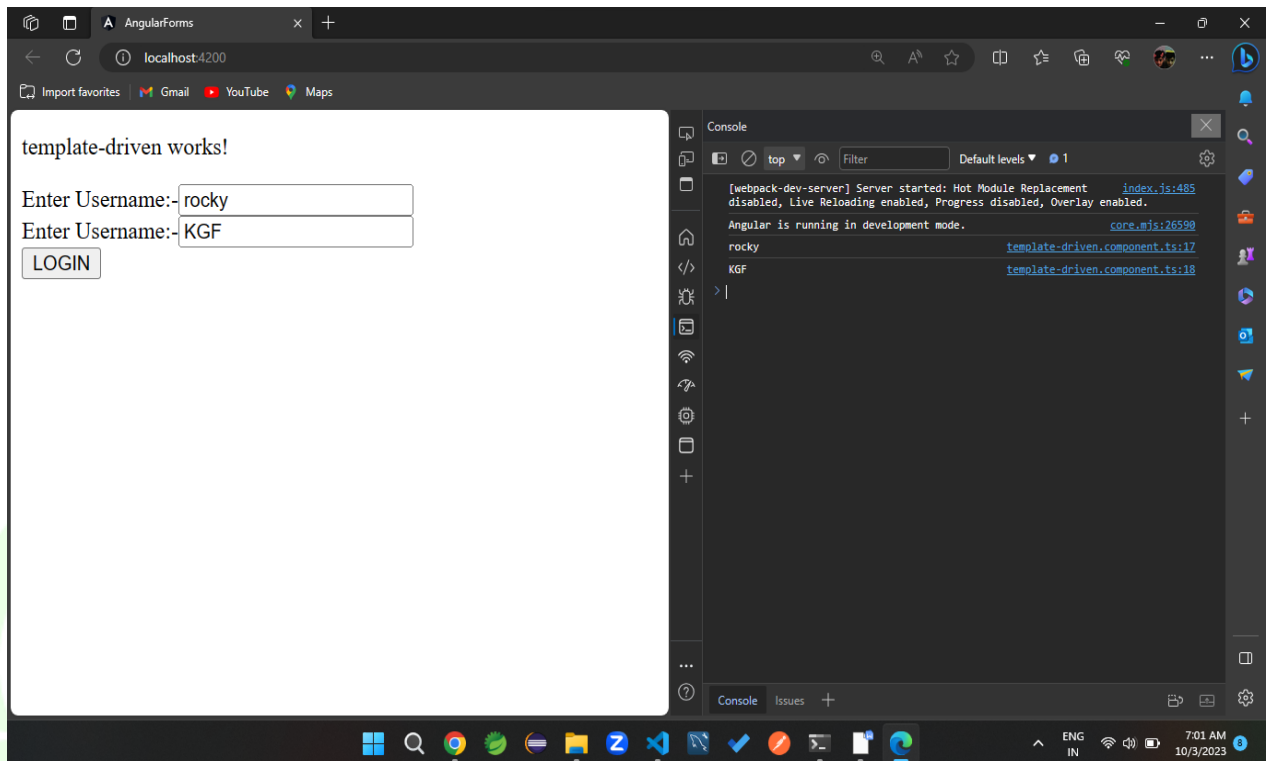
Complete Java Classes

by Kunal Sir

Forms

Angular

OUTPUT :-



❖ Reactive Forms:-

Reactive form is the structural way to define forms in angular. Reactive forms provide controls for inputs.

We use FormGroup class to declare Reactive Form.

We use FormBuilder class to build Reactive form in angular.

Reactive forms can have multiple controls.

Controls are use to emit input values in component.

Reactive forms supports validation for each form input.

Reactive Forms provide patchValue and setValue function to add/update form control values.



CJC

Complete Java Classes

Angular

by Kunal Sir
Forms

Example on Reactive forms:-

1. Create component using command: `ng g c Reactive --skip-test`

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
UPDATE src/app/app.module.ts (430 bytes)
PS E:\Angular Workspace\AngularForms> ng g c reactive --skip-tests
```

2. Add the component selector in app.component.html file.

```
app.component.html M X
src > app > app.component.html > app-reactive
Go to component
1 <!-- <app-template-driven>c</app-template-driven> -->
2 <app-reactive>*/app-reactive>
```

3. Import ReactiveFormsModule in app.module.ts

```
app.component.html M app.module.ts M X
src > app > app.module.ts > AppModule
4 import { AppComponent } from './app.component';
5 import { TemplateDrivenComponent } from './template-driven/template-driven.component';
6 import { FormsModule } from '@angular/forms';
7 import { ReactiveComponent } from './reactive/reactive.component';
8 import { ReactiveFormsModule } from '@angular/forms';
9 @NgModule({
10   declarations: [
11     AppComponent,
12     TemplateDrivenComponent,
13     ReactiveComponent
14   ],
15   imports: [
16     BrowserModule,
17     FormsModule,
18     ReactiveFormsModule
19   ],
20   providers: [],
21   bootstrap: [AppComponent]
22 })
23 export class AppModule { }
24
```



by Kunal Sir

Angular

Forms

4. In ReactiveComponent define one User registration form using FormBuilder, also create one function to log the values of form controls.

```
reactive.component.ts U X
src > app > reactive > reactive.component.ts > ReactiveComponent > ngOnInit
1  import { Component,OnInit } from '@angular/core';
2  import { FormBuilder,FormGroup } from "@angular/forms";
3  @Component({
4    selector: 'app-reactive',
5    templateUrl: './reactive.component.html',
6    styleUrls: ['./reactive.component.css']
7  })
8  export class ReactiveComponent implements OnInit{
9
10     constructor(private fb:FormBuilder){}
11     userRegistration:FormGroup;
12     ngOnInit(): void {
13       this.userRegistration=this.fb.group([
14         {
15           firstName:[],
16           lastName:[],
17           emailId:[],
18           username:[],
19           password:[]
20         }
21       ])
22     }
23
24     onRegister()
25     {
26       console.log('First Name :- '+this.userRegistration.controls['firstName'].value);
27       console.log('last Name :- '+this.userRegistration.controls['lastName'].value);
28       console.log('Email ID :- '+this.userRegistration.controls['emailId'].value);
29       console.log('Username :- '+this.userRegistration.controls['username'].value);
30       console.log('Password :- '+this.userRegistration.controls['password'].value);
31     }
32   }
33 }
34
```



CJC

Complete Java Classes

by Kunal Sir

Forms

Angular

5. Bind this user registration form into html.

```
reactive.component.html U X
src > app > reactive > reactive.component.html > form > input#email
Go to component
1 <p>reactive works!</p>
2 <form [formGroup]="userRegistration" (ngSubmit)="onRegister()">
3
4   <label for="fn">Enter First Name:-</label>
5   <input type="text" id="fn" formControlName="firstName">
6   <br>
7   <label for="ln">Enter Last Name:-</label>
8   <input type="text" id="ln" formControlName="lastName">
9   <br>
10  <label for="email">Enter Email ID:-</label>
11  <input type="text" id="email" formControlName="emailId">
12  <br>
13  <label for="un">Enter Username:-</label>
14  <input type="text" id="un" formControlName="username">
15  <br>
16  <label for="ps">Enter Password:-</label>
17  <input type="text" id="ps" formControlName="password">
18  <br>
19  <button type="submit">REGISTER</button>
20 </form>
21
```



CJC

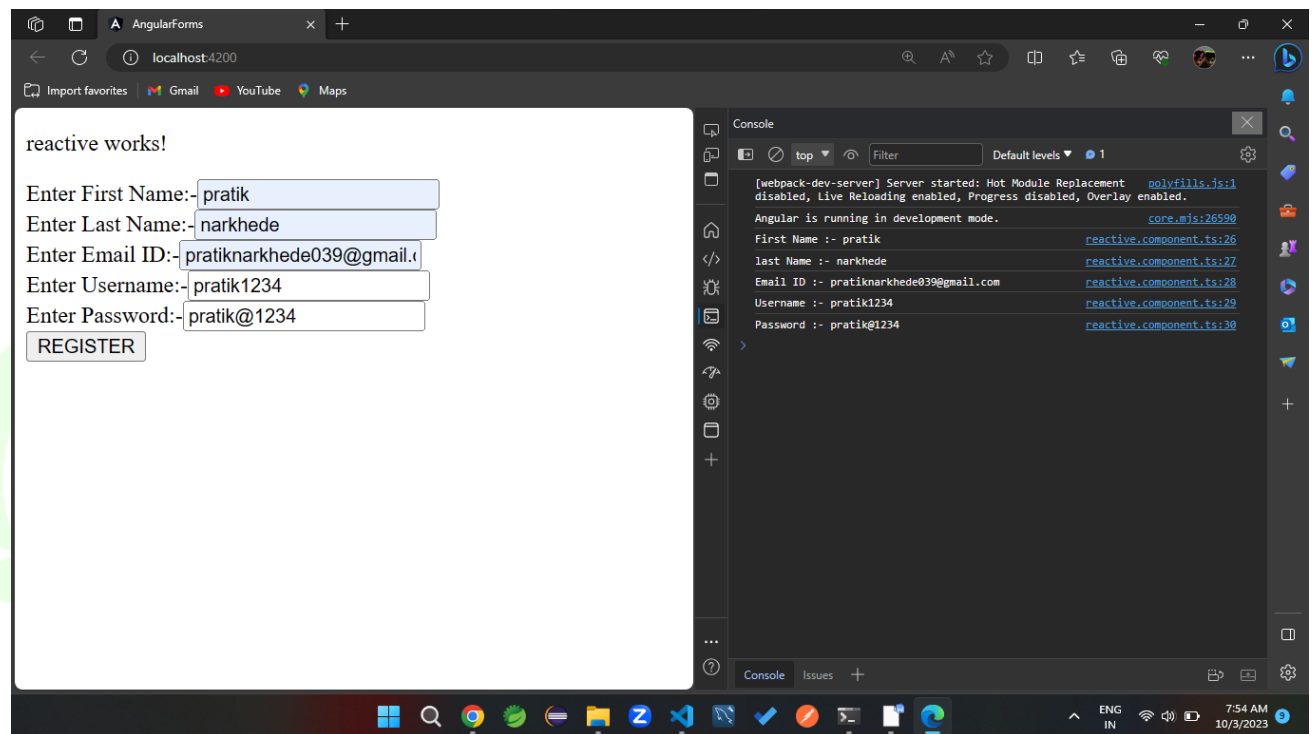
Complete Java Classes

by Kunal Sir

Forms

Angular

OUTPUT :-





CJC

Complete Java Classes

by Kunal Sir

Forms

Angular

◆ How to use setValue and patchValue functions?

➤ Snippet of reactive.component.ts

```
reactive.component.ts U x
src > app > reactive > reactive.component.ts > ReactiveComponent
1  import { Component,OnInit } from '@angular/core';
2  import { FormBuilder,FormGroup } from "@angular/forms";
3  @Component({
4    selector: 'app-reactive',
5    templateUrl: './reactive.component.html',
6    styleUrls: ['./reactive.component.css']
7  })
8  export class ReactiveComponent implements OnInit{
9    constructor(private fb:FormBuilder){}
10   userRegistration:FormGroup;
11   ngOnInit(): void {
12     this.userRegistration=this.fb.group(
13       {
14         firstName:[],
15         lastName:[],
16         emailId:[],
17         username:[],
18         password:[]
19       }
20     )
21   }
22
23   onRegister()
24   {
25     console.log('First Name :- '+this.userRegistration.controls['firstName'].value);
26     console.log('last Name :- '+this.userRegistration.controls['lastName'].value);
27     console.log('Email ID :- '+this.userRegistration.controls['emailId'].value);
28     console.log('Username :- '+this.userRegistration.controls['username'].value);
29     console.log('Password :- '+this.userRegistration.controls['password'].value);
30   }
31
32   onSetValue()
33   {
34     this.userRegistration.controls['firstName'].setValue('Shubham');
35     this.userRegistration.controls['lastName'].setValue('ingale');
36     this.userRegistration.controls['emailId'].setValue('shubhamingale@gmail.com');
37     this.userRegistration.controls['username'].setValue('Shubham123');
38     this.userRegistration.controls['password'].setValue('Shubham@123');
39   }
```



CJC

Complete Java Classes

by Kunal Sir

Angular

Forms

```
40
41     onPatchValue()
42     {
43         this.userRegistration.patchValue({
44             firstName:"Sagar",
45             lastName:"sarode",
46             emailId:"sagarsarode@gmail.com",
47             username:"sagar123",
48             password:"sagar@123"
49         })
50     }
51 }
52
```

➤ Snippet of reactive.component.html

```
reactive.component.html U X
src > app > reactive > reactive.component.html > form > label
1 <p>reactive works!</p>
2 <form [formGroup]="userRegistration" (ngSubmit)="onRegister()">
3   <label for="fn">Enter First Name:-</label>
4   <input type="text" id="fn" formControlName="firstName">
5   <br>
6   <label for="ln">Enter Last Name:-</label>
7   <input type="text" id="ln" formControlName="lastName">
8   <br>
9   <label for="email">Enter Email ID:-</label>
10  <input type="text" id="email" formControlName="emailId">
11  <br>
12  <label for="un">Enter Username:-</label>
13  <input type="text" id="un" formControlName="username">
14  <br>
15  <label for="ps">Enter Password:-</label>
16  <input type="text" id="ps" formControlName="password">
17  <br>
18  <button type="submit">REGISTER</button>
19  <button (click)="onSetValue()">SET VALUE</button>
20  <button (click)="onPatchValue()">Patch VALUE</button>
21 </form>
```



CJC

Complete Java Classes

by Kunal Sir

Angular

Forms

◆ How to add validation in Reactive form?

➤ Snippet of reactive.component.ts

```
reactive.component.ts U X
src > app > reactive > reactive.component.ts > ReactiveComponent > ngOnInit
1  import { Component,OnInit } from '@angular/core';
2  import { FormBuilder,FormGroup, Validators } from "@angular/forms";
3  @Component({
4    selector: 'app-reactive',
5    templateUrl: './reactive.component.html',
6    styleUrls: ['./reactive.component.css']
7  })
8  export class ReactiveComponent implements OnInit{
9    constructor(private fb:FormBuilder){}
10   userRegistration:FormGroup;
11   ngOnInit(): void {
12     this.userRegistration=this.fb.group(
13       {
14         firstName:['',[Validators.required,Validators.maxLength(10),
15         | Validators.minLength(3)]],
16         lastName:[],
17         emailId:['',[Validators.required,Validators.email]],
18         username:[],
19         password:[]
20       }
21     )
22   }
```



CJC

Complete Java Classes

by Kunal Sir

Forms

Angular

➤ Snippet of reactive.component.html

```
reactive.component.html U X
src > app > reactive > reactive.component.html > p
1 <p>reactive works!</p>
2 <form [formGroup]="userRegistration" (ngSubmit)="onRegister()">
3
4   <div *ngIf="userRegistration.controls['firstName'].touched && userRegistration.controls['firstName'].invalid">
5
6     <span *ngIf="userRegistration.controls['firstName'].hasError('required')" >
7       * Please Enter Firstname.
8     </span>
9     <span *ngIf="userRegistration.controls['firstName'].hasError('minlength')" >
10      * Firstname must have at least 3 characters.
11    </span>
12    <span *ngIf="userRegistration.controls['firstName'].hasError('maxlength')" >
13      * Firstname should have characters less than 10.
14    </span>
15  </div>
16  <label for="fn">Enter First Name:-</label>
17  <input type="text" id="fn" formControlName="firstName">
18  <br>
19
20  <label for="ln">Enter Last Name:-</label>
21  <input type="text" id="ln" formControlName="lastName">
22
23
24  <br>
25  <div *ngIf="userRegistration.controls['emailId'].touched && userRegistration.controls['emailId'].invalid">
26
27    <span *ngIf="userRegistration.controls['emailId'].hasError('required')" >
28      * Please Enter Email Id.
29    </span>
30    <span *ngIf="userRegistration.controls['emailId'].hasError('email')" >
31      * Please Enter valid Email Id.
32    </span>
33  </div>
34  <label for="email">Enter Email ID:-</label>
35  <input type="text" id="email" formControlName="emailId">
36  <br>
37  <label for="un">Enter Username:-</label>
38  <input type="text" id="un" formControlName="username">
39  <br>
40  <label for="ps">Enter Password:-</label>
41  <input type="text" id="ps" formControlName="password">
42  <br>
43  <button type="submit">REGISTER</button>
44  <button (click)="onSetValue()">SET VALUE</button>
45  <button (click)="onPatchValue()">Patch VALUE</button>
46
47 </form>
```