

How to set value in template-driven forms in Angular

5 Comments / 8 minutes of reading / March 9, 2023



[Angular Tutorial](#)

[Reactive Forms in Angular](#)



[Angular Template Driven Forms](#)

In this tutorial, we will learn how to set value in [template-driven forms in Angular](#). We will learn how to set the default or initial value to form controls, dynamically set values, reset the value of the form, etc. Learn how to set the value of individual FormControl or a FormGroup and nested FormGroup .

We have covered [how to create template-driven forms in the angular tutorial](#). We will continue from there and in this tutorial, we will show you

Table of Contents

[Template](#)

[Set value in template-driven forms](#)

[Two-way data binding](#)

[Template reference variable](#)

[Summary](#)

Template

The following is the app.component.html from the [angular template-driven forms tutorial](#).

```
1
2 <form #contactForm="ngForm" (ngSubmit)="onSubmit(contactForm)">
3
4   <p>
5     <label for="firstname">First Name </label>
6     <input type="text" id="firstname" name="firstname" ngModel>
7   </p>
8
9   <p>
10    <label for="lastname">Last Name </label>
11    <input type="text" id="lastname" name="lastname" ngModel>
12  </p>
13
14  <p>
15    <label for="email">Email </label>
16    <input type="text" id="email" name="email" ngModel>
17  </p>
18
19  <p>
20    <label for="gender">Geneder </label>
21    <input type="radio" value="male" id="gender" name="gender" ngModel> Male
22    <input type="radio" value="female" id="gender" name="gender" ngModel> Female
23  </p>
24
25  <p>
26    <label for="isMarried">Married </label>
27    <input type="checkbox" id="isMarried" name="isMarried" ngModel>
28  </p>
29
30  <p>
31    <label for="country">country </label>
32    <select id="country" name="country" ngModel>
33      <option [ngValue]="c.id" *ngFor="let c of countryList">
34        {{c.name}}
35      </option>
36    </select>
37  </p>
38
39  <div ngModelGroup="address">
40
41    <p>
42      <label for="city">City</label>
43      <input type="text" id="city" name="city" ngModel>
```

```
44     </p>
45
46     <p>
47       <label for="street">Street</label>
48       <input type="text" id="street" name="street" ngModel>
49     </p>
50     <p>
51       <label for="pincode">Pin Code</label>
52       <input type="text" id="pincode" name="pincode" ngModel>
53     </p>
54
55   </div>
56
57   <p>
58     <button type="submit">Submit</button>
59   </p>
60
61 </form>
62
```

Before we set the default value, it is better to create a model class for the above form. Open the `app.component.ts` and add the following class

```
1
2 export class contact {
3   firstname:string;
4   lastname:string;
5   email:string;
6   gender:string;
7   isMarried:boolean;
8   country:string;
9   address: {
10     city:string;
11     street:string;
12     pincode:string;
13   }
14 }
15
```

Set value in template-driven forms

There are two ways you can set the value of the form elements

- Two-way data binding
- Use the template reference variable

Two-way data binding

The [two-way data binding](#) is the recommended way to set the value in the template-driven forms.

The following code uses the `[(ngModel)]="contact.firstname"` to bind the `firstname` HTML element to the `contact.firstname` field in the component class. The advantage here is that any changes made in the form are automatically propagated to the component class and changes made in component class are immediately shown in the form.

```
1  
2 <label for="firstname">First Name </label>  
3 <input type="text" id="firstname" name="firstname" [(ngModel)]="contact.firstname">  
4
```

Set the default/initial value

To set the initial or default value all you need to populate the contact model in the [ngOnInit](#) method as shown below

```
1
```

```
2 ngOnInit() {  
3  
4   this.contact = {  
5     firstname: "Sachin",  
6     lastname: "Tendulkar",  
7     email: "sachin@gmail.com",  
8     gender: "male",  
9     isMarried: true,  
10    country: "2",  
11    address: { city: "Mumbai", street: "Perry Cross Rd", pincode: "400050" }  
12  };  
13  
14 }  
15
```

Set the value individually or dynamically

```
1  
2 changeCountry() {  
3   this.contact.country = "1";  
4 }  
5
```

Reset form

```
1  
2 <button type="button" (click)="reset(contactForm)">Reset</button>  
3
```

```
1  
2 reset(contactForm :NgForm) {  
3   contactForm.resetForm();  
4 }  
5
```

[tabby title="app.component.ts"]

```
1  
2 import { Component, ViewChild, ElementRef, OnInit } from '@angular/core';  
3 import { NgForm } from '@angular/forms';  
4
```

```
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.css']
9 })
10 export class AppComponent implements OnInit {
11   title = 'Template driven forms';
12
13
14   countryList: country[] = [
15     new country("1", "India"),
16     new country('2', 'USA'),
17     new country('3', 'England')
18   ];
19
20   contact: contact;
21
22   ngOnInit() {
23
24     this.contact = {
25       firstname: "Sachin",
26       lastname: "Tendulkar",
27       email: "sachin@gmail.com",
28       gender: "male",
29       isMarried: true,
30       country: "2",
31       address: { city: "Mumbai", street: "Perry Cross Rd", pincode: "400050" }
32     };
33
34   }
35
36   onSubmit() {
37     console.log(this.contact);
38   }
39
40   setDefaults() {
41     this.contact = {
42       firstname: "Sachin",
43       lastname: "Tendulkar",
44       email: "sachin@gmail.com",
45       gender: "male",
46       isMarried: true,
47       country: "2",
48       address: { city: "Mumbai", street: "Perry Cross Rd", pincode: "400050" }
49     };
50   }
51
52   changeCountry() {
53     this.contact.country = "1";
```

```
54   }
55
56   reset(contactForm :NgForm) {
57     contactForm.resetForm();
58   }
59
60 }
61
62 export class contact {
63   firstname: string;
64   lastname: string;
65   email: string;
66   gender: string;
67   isMarried: boolean;
68   country: string;
69   address: {
70     city: string;
71     street: string;
72     pincode: string;
73   }
74 }
75
76
77 export class country {
78   id: string;
79   name: string;
80
81   constructor(id: string, name: string) {
82     this.id = id;
83     this.name = name;
84   }
85 }
86
```

[tabby title="app.component.html"]

```
1
2 <form #contactForm="ngForm" (ngSubmit)="onSubmit(contactForm)">
3
4   <p>
5     <label for="firstname">First Name </label>
6     <input type="text" id="firstname" name="firstname" [(ngModel)]="contact.firstname">
7   </p>
8
9   <p>
10    <label for="lastname">Last Name </label>
```

```
11 <input type="text" id="lastname" name="lastname" [(ngModel)]="contact.lastname">
12 </p>
13
14 <p>
15 <label for="email">Email </label>
16 <input type="text" id="email" name="email" [(ngModel)]="contact.email">
17 </p>
18
19 <p>
20 <label for="gender">Gender </label>
21 <input type="radio" value="male" id="gender" name="gender" [(ngModel)]="contact.g
22 <input type="radio" value="female" id="gender" name="gender" [(ngModel)]="contact.
23
24 </p>
25
26 <p>
27 <label for="isMarried">Married </label>
28 <input type="checkbox" id="isMarried" name="isMarried" [(ngModel)]="contact.isMarried
29 </p>
30
31 <p>
32 <label for="country">country </label>
33 <select id="country" name="country" [(ngModel)]="contact.country">
34 <option [ngValue]="c.id" *ngFor="let c of countryList">
35 {{c.name}}
36 </option>
37 </select>
38 </p>
39
40 <div ngModelGroup="address">
41
42 <p>
43 <label for="city">City</label>
44 <input type="text" id="city" name="city" [(ngModel)]="contact.address.city">
45 </p>
46
47 <p>
48 <label for="street">Street</label>
49 <input type="text" id="street" name="street" [(ngModel)]="contact.address.street">
50 </p>
51
52 <p>
53 <label for="pincode">Pin Code</label>
54 <input type="text" id="pincode" name="pincode" [(ngModel)]="contact.address.pincode
55 </p>
56
57 </div>
58
59 <p>
```



```
60     <button type="submit">Submit</button>
61 </p>
62
63 <p>
64     <button type="button" (click)="changeCountry()">Change Country</button>
65     <button type="button" (click)="setDefaults()">Set Defaults</button>
66     <button type="button" (click)="reset(contactForm)">Reset</button>
67 </p>
68
69 <b>valid</b> {{contactForm.valid}}
70 <b>touched</b> {{contactForm.touched}}
71 <b>pristine</b> {{contactForm.pristine}}
72 <b>dirty</b> {{contactForm.dirty}}
73
74 </form>
75
76
```

[tabbyending]

Template reference variable

We have a `#contactForm` reference variable, which is an instance of `ngForm`.

```
1
2 <form #contactForm="ngForm" (ngSubmit)="onSubmit(contactForm)">
3
```

We can get the reference to the `#contactForm` in the `app.component.ts`, using the `ViewChild`

```
1  
2 @ViewChild('contactForm', null) contactForm: NgForm;  
3
```

Once we have the reference, we can use the `setValue` method of the `ngForm` to set the initial value

Set the default or initial value

```
1  
2 ngOnInit() {  
3  
4   this.contact = {  
5     firstname: "Sachin",  
6     lastname: "Tendulkar",  
7     email: "sachin@gmail.com",  
8     gender: "male",  
9     isMarried: true,  
10    country: "2",  
11    address: {  
12      city: "Mumbai",  
13      street: "Perry Cross Rd",  
14      pincode: "400050"  
15    }  
16  };  
17  
18  setTimeout(() => {  
19    this.contactForm.setValue(this.contact);  
20  });  
21  
22  }  
23
```

Note that we are using the `setTimeout`. That is because the form controls are yet initialized when the `OnInit` is fired. We will get the following error message

There are no form controls registered with this group yet. If you're using `ngModel`, you may want to check next tick (e.g. use `setTimeout`).

Set the value individually or dynamically

You can also set the value individually using the `setValue` method of the individual `FormControl`.

You will get the reference to the individual `FormControl` from the `controls` collection of the `ngForm`. Once you get the reference use the `setValue` on the `FormControl` instance to change the value.

For Example, this code will change the country to India

```
1  
2 changeCountry() {
```

```
3  this.contactForm.controls["country"].setValue("1");  
4  }  
5
```

Call the `changeCountry` method from the Template.

```
1  
2  <button type="button" (click)="changeCountry()">Change Country</button>  
3
```

Reset values

You can reset the form to empty value using the `reset` or `resetForm` method of the `ngForm`. These also resets the form status like `dirty`, `valid`, `pristine` & `touched`, etc

```
1  
2  reset() {  
3    this.contactForm.reset();  
4  }  
5
```

```
1  
2  resetForm() {  
3    this.contactForm.resetForm();  
4  }  
5
```

Set Default Value

You can invoke the `setValue` anytime to set the form back to the default value. This will set the entire form to the value held by the contact form.

```
1  
2  setDefaults() {  
3    this.contactForm.setValue(this.contact);  
4  }  
5
```

patch value

You can make use of the [patchValue](#) to change the only few fields anytime. The control property of the ngForm returns the reference to the top level FormGroup . Then, you can make use of the patchValue method to change only firstname , lastname & email fields

```
1
2  patchValue() {
3    let obj = {
4      firstname: "Rahul",
5      lastname: "Dravid",
6      email: "rahul@gmail.com",
7    };
8
9    this.contactForm.control.patchValue(obj);
10
11  }
12
```

Set value of nested FormGroup

You can update nested [FormGroup](#) by getting a reference to the nested FormGroup from the controls collection of ngForm .

```
1
2  changeAddress() {
3    let obj = {
```

```
4     city: "Bangalore",
5     street: "Brigade Road",
6     pincode: "600100"
7   };
8   let address= this.contactForm.controls["address"] as FormGroup
9   address.patchValue(obj);
10
11 }
12
```

The complete code.

[tabby title="app.component.ts"]

```
1
2 import { Component, ViewChild, ElementRef, OnInit } from '@angular/core';
3 import { NgForm, FormGroup } from '@angular/forms';
4
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.css']
9 })
10 export class AppComponent implements OnInit {
11   title = 'Template driven forms';
12
13   @ViewChild('contactForm', null) contactForm: NgForm;
14
15   countryList: country[] = [
16     new country("1", "India"),
17     new country('2', 'USA'),
18     new country('3', 'England')
19   ];
20
21   contact: contact;
22
23   ngOnInit() {
24
25     this.contact = {
26       firstname: "Sachin",
27       lastname: "Tendulkar",
28       email: "sachin@gmail.com",
29       gender: "male",
30       isMarried: true,
```

```
31     country: "2",
32     address: {
33       city: "Mumbai",
34       street: "Perry Cross Rd",
35       pincode: "400050"
36     }
37   };
38
39   setTimeout(() => {
40     this.contactForm.setValue(this.contact);
41   });
42
43 }
44
45 onSubmit() {
46   console.log(this.contactForm.value);
47 }
48
49 setDefaults() {
50   this.contactForm.setValue(this.contact);
51 }
52
53 changeCountry() {
54   this.contactForm.controls["country"].setValue("1");
55 }
56
57 patchValue() {
58   let obj = {
59     firstname: "Rahul",
60     lastname: "Dravid",
61     email: "rahul@gmail.com",
62   };
63
64   this.contactForm.control.patchValue(obj);
65
66 }
67
68 changeAddress() {
69   let obj = {
70     city: "Bangalore",
71     street: "Brigade Road",
72     pincode: "600100"
73   };
74   let address= this.contactForm.controls["address"] as FormGroup
75   address.patchValue(obj);
76
77 }
78
79 reset() {
```

```
80     this.contactForm.reset();
81   }
82
83   resetForm() {
84     this.contactForm.resetForm();
85   }
86 }
87
88
89 export class contact {
90   firstname: string;
91   lastname: string;
92   email: string;
93   gender: string;
94   isMarried: boolean;
95   country: string;
96   address: {
97     city: string;
98     street: string;
99     pincode: string;
100  }
101 }
102
103
104 export class country {
105   id: string;
106   name: string;
107
108   constructor(id: string, name: string) {
109     this.id = id;
110     this.name = name;
111   }
112 }
113
114
```

[tabby title="app.component.html"]

```
1
2
3 <form #contactForm="ngForm" (ngSubmit)="onSubmit(contactForm)">
4
5   <p>
6     <label for="firstname">First Name </label>
7     <input type="text" id="firstname" name="firstname" ngModel>
8   </p>
```



```
9
10 <p>
11   <label for="lastname">Last Name </label>
12   <input type="text" id="lastname" name="lastname" ngModel>
13 </p>
14
15 <p>
16   <label for="email">Email </label>
17   <input type="text" id="email" name="email" ngModel>
18 </p>
19
20 <p>
21   <label for="gender">Geneder </label>
22   <input type="radio" value="male" id="gender" name="gender" ngModel> Male
23   <input type="radio" value="female" id="gender" name="gender" ngModel> Female
24 </p>
25
26 <p>
27   <label for="isMarried">Married </label>
28   <input type="checkbox" id="isMarried" name="isMarried" ngModel>
29 </p>
30
31 <p>
32   <label for="country">country </label>
33
34   <select id="country" name="country" ngModel>
35     <option [ngValue]="c.id" *ngFor="let c of countryList">
36       {{c.name}}
37     </option>
38   </select>
39
40 </p>
41
42 <div ngModelGroup="address">
43
44   <p>
45     <label for="city">City</label>
46     <input type="text" id="city" name="city" ngModel>
47   </p>
48
49   <p>
50     <label for="street">Street</label>
51     <input type="text" id="street" name="street" ngModel>
52   </p>
53   <p>
54     <label for="pincode">Pin Code</label>
55     <input type="text" id="pincode" name="pincode" ngModel>
56   </p>
57
```

```
58 </div>
59
60 <p>
61   <button type="submit">Submit</button>
62 </p>
63
64 <p>
65   <button type="button" (click)="changeCountry()">Change Country</button>
66   <button type="button" (click)="setDefaults()">Set Defaults</button>
67   <button type="button" (click)="patchValue()">Patch Value</button>
68   <button type="button" (click)="changeAddress()">Change Address</button>
69   <button type="button" (click)="reset()">Reset</button>
70   <button type="button" (click)="resetForm()">Reset Form</button>
71 </p>
72
73 <b>valid</b> {{contactForm.valid}}
74 <b>touched</b> {{contactForm.touched}}
75 <b>pristine</b> {{contactForm.pristine}}
76 <b>dirty</b> {{contactForm.dirty}}
77
78 </form>
79
80
```

[tabbyending]