

What is the difference between FormControlName and FormControl?

Asked 2 years, 11 months ago Active 1 year, 2 months ago Viewed 45k times



I'm using `ReactiveFormsModule` of Angular2 to create a component that contains a form. Here is my code:

73

foo.component.ts:



23

```
constructor(fb: FormBuilder) {  
  this.myForm = fb.group({  
    'fullname': ['', Validators.required],  
    'gender': []  
  });  
}
```

foo.component.html (with `[formControl]`):

```
<div class="fields">  
  <div class="field">  
    <label>Fullname*</label>  
    <input type="text" [formControl]="myForm.controls.fullname"/>  
  </div>  
</div>  
  
<div class="inline fields">  
  <label for="gender">Gender</label>  
  <div class="field">  
    <div class="ui radio checkbox">  
      <input type="radio" name="gender" checked="" tabindex="0" class="hidden"  
[formControl]="myForm.controls.gender">  
      <label>Male</label>  
    </div>  
  </div>  
  <div class="field">  
    <div class="ui radio checkbox">  
      <input type="radio" name="gender" tabindex="0" class="hidden"  
[formControl]="myForm.controls.gender">  
      <label>Female</label>  
    </div>  
  </div>  
</div>
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

foo.component.html (with `formControlName`):

```
<div class="fields">
  <div class="field">
    <label>Fullname*</label>
    <input type="text" formControlName="fullname"/>
  </div>
</div>

<div class="inline fields">
  <label for="gender">Gender</label>
  <div class="field">
    <div class="ui radio checkbox">
      <input type="radio" name="gender" checked="" tabindex="0" class="hidden"
formControlName="gender">
      <label>Male</label>
    </div>
  </div>
  <div class="field">
    <div class="ui radio checkbox">
      <input type="radio" name="gender" tabindex="0" class="hidden"
formControlName="gender">
      <label>Female</label>
    </div>
  </div>
</div>
```

Both ways work. But i cannot figure out what is the difference between using `[formControl]` and `formControlName`.

[angular](#) [radio-button](#) [angular2-forms](#) [form-control](#) [angular2-formbuilder](#)

asked Oct 21 '16 at 8:33



[smartmouse](#)

5,122 19 62 118

-
- 1 I'd say that the main reason for using `formControlName` over `formControl` is when you don't want to maintain individual `FormControl` instances in the component. – [Paul Samsot](#) Oct 21 '16 at 8:46
-

5 Answers

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



132



I believe you missed an important point: `[formGroup]` directive in the second example. `formControlName` is used together with `[formGroup]` to save your form multiple dot navigations. For example:

```
<div>
  <input type="text" [formControl]="myForm.controls.firstName"/>
  <input type="text" [formControl]="myForm.controls.lastName"/>
  <input type="text" [formControl]="myForm.controls.email"/>
  <input type="text" [formControl]="myForm.controls.title"/>
</div>
```

Is equivalent to:

```
<div [formGroup]="myForm">
  <input type="text" formControlName="firstName"/>
  <input type="text" formControlName="lastName"/>
  <input type="text" formControlName="email"/>
  <input type="text" formControlName="title"/>
</div>
```

Now imagine nested `FormGroups` :)

answered Oct 21 '16 at 8:38



Harry Ninh

10.4k

3

47

47

using `[formControl]="form.get('Registration.Attributes.aboutme')"` caused issues.. but works fine with `formControlName="firstNRegistration.Attributes.aboutmeame"` – [Ricardo Saracino](#) Sep 14 '18 at 13:16

`[formControl]` causing issue while `form.valid` validation with `formGroup`, any comments – [Pardeep Jain](#) Nov 28 '18 at 8:03

how can i handle if the input elent is another component. how do I bind the fomrcontrol withe the component – [Ramakanth Reddy](#) Sep 25 at 9:10



17

`[formControl]` assigns a reference to the `FormControl` instance you created to the `FormControlDirective` .

`formControlName` assigns a string for the forms module to look up the control by name.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```

constructor(fb: FormBuilder) {
  this.fullName = new FormControl('', Validators.required);
  this.gender = new FormControl('');
  this.myForm = fb.group({
    'fullname': this.fullName,
    'gender': this.gender
  });
}

```

answered Oct 21 '16 at 8:42

**Günter Zöchbauer****373k** 88 1197 1084

when I add `this.fullName = new FormControl("", Validators.required);` I got an error like you cannot assign because it's read-only property or constant, but here I am taken as a variable. So please help – [Brijesh Mavani](#) Feb 5 '18 at 6:21

Please post the **exact** error message. It's probably even better to create a new question containing your code that allows to reproduce – [Günter Zöchbauer](#) Feb 5 '18 at 6:23

There is a 3rd equivalency to the two provided in the accepted answer, which is this (not recommended):

5

```

<div [formGroup]="myForm">
  <input type="text" [formControl]="firstName"/>
  <input type="text" [formControl]="lastName"/>
  <input type="text" [formControl]="email"/>
  <input type="text" [formControl]="title"/>
</div>

```

Notice that we are still using the `[formGroup]` directive.

However, for this template to compile without error, then your component needs to declare the controls as `AbstractControls` and not `FormControls`:

`myComponent.ts`

```
firstName: AbstractControl
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

However, please note that declaring AbstractControls is [not recommended](#), so if you get the error `Cannot find control with unspecified name attribute` then it is probable you have mixed the styles or have declared your controls as AbstractControls.

answered Feb 27 '18 at 14:05



[rmcsharry](#)

2,810 3 33 62

how can i handle if the input elent is another component. how do I bind the fomrcontrol withe the component – [Ramakanth Reddy](#) Sep 25 at 9:10

You cannot - even if there IS a way, you should not. The element should be bound to the control defined IN THAT COMPONENT. If you want to pass data to another component then use a service (or if it is parent component then event emitter). Google how to pass data between components – [rmcsharry](#) Sep 25 at 12:09

can you please look at this post stackoverflow.com/questions/58100248/... – [Ramakanth Reddy](#) Sep 25 at 14:05 ✎

From the Angular docs (<https://angular.io/guide/reactive-forms>):

1

Component

```
@Component({
  ...
})
export class ProfileEditorComponent {
  profileForm = new FormGroup({
    firstName: new FormControl(''),
    lastName: new FormControl(''),
  });
}
```

Template

```
<form [formGroup]="profileForm">
```

```
  <label>
    First Name:
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
<label>
  Last Name:
  <input type="text" formControlName="lastName">
</label>

</form>
```

Note that just as the `FormGroup` contains a group of controls, the `profileForm` `FormGroup` is bound to the form element with the `FormGroup` directive, creating a communication layer between the model and the form containing the inputs. The `formControlName` input provided by the `FormControlName` directive binds each individual input to the form control defined in the `FormGroup`

answered Aug 10 '18 at 3:18



[Chris Halcrow](#)

13.2k 6 87 104



0



with `[formControl]` you can use Reactive programming advantages because `FormControl` has a property named `valueChanges` (I know this one right now, maybe there is more than that) which returns an `Observable` which you can subscribe and use it. (for example, it is very useful in register scenarios which you want to check input email to be not repeated as soon as user change the value)

answered Feb 13 '18 at 20:59



[Seyed Ali Roshan](#)

656 8 22

Yes. But you still use `formControlName` in the template, even when using the model in your answer. Just assign the `formControlName="someFormControlName"` to a `FormControl` in the `component.ts` file like `someFormControlName: FormControl`; – [Charles Robertson](#)
Dec 17 '18 at 9:31