

# ngModel cannot be used to register form controls with a parent formGroup directive

Asked 3 years, 1 month ago   Active 1 month ago   Viewed 69k times



After upgrading to RC5 we began getting this error:

60

ngModel cannot be used to register form controls with a parent formGroup directive. Try using formGroup's partner directive "formControlName" instead. Example:



7

```
<div [formGroup]="myGroup">
  <input formControlName="firstName">
</div>
```

In your class:

```
this.myGroup = new FormGroup({
  firstName: new FormControl()
});
```

Or, if you'd like to avoid registering this form control, indicate that it's standalone in ngModelOptions:

Example:

```
<div [formGroup]="myGroup">
  <input formControlName="firstName">
  <input [(ngModel)]="showMoreControls" [ngModelOptions]="{standalone: true}">
</div>
```

It looks like in RC5 the two can no longer be used together, but I could not find an alternative solution.

Here is the component producing the exception:

```
<select class="field form-control" [formGroup]="form" [(ngModel)]="cause_id"
```

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



angular2-forms

angular2-formbuilder

asked Aug 24 '16 at 14:50



user2363245

580 1 5 13

Did you import `FormsModule` and `ReactiveFormsModule` ? – [Günter Zöchbauer](#) Aug 24 '16 at 14:54

Sure, both of them – [user2363245](#) Aug 24 '16 at 15:25

AFAIK, the only existing explanation is the following: [blog.angular-university.io/...](http://blog.angular-university.io/...) – [user2363245](#) Aug 24 '16 at 15:26

Found source of the change: [github.com/angular/angular/pull/10314](https://github.com/angular/angular/pull/10314) – [user2363245](#) Aug 24 '16 at 15:40

## 8 Answers



The answer is right on the error message, you need to indicate that it's standalone and therefore it doesn't conflict with the form controls:

119

```
[ngModelOptions]="{standalone: true}"
```



answered Nov 14 '16 at 12:36



Avenir Çokaj

1,424 1 11 10

1 what does standalone means? – [Jas](#) Oct 11 '17 at 11:48

1 It means that its not handled by the form model/data so you can pass data by any object/model you want as it used to be in AngularJS 1 – [Avenir Çokaj](#) Oct 12 '17 at 18:38

I only see this problem in my test setup. What is exactly missing? `[ngModelOptions]="{standalone: true}"` fixes test but changes logic. `ngModel` is inherited from parent component, that declares `ngForm` in my case – [aholbreich](#) Dec 21 '17 at 13:13

1 just tested... doesn't work with Angular 5 – [Gatschet](#) May 2 '18 at 6:38

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

▲ Expanding on @Avenir Çokaj's answer

22

Being a novice even I did not understand the error message clearly at first.

▼

What the error message indicates is that in your formGroup you have an element that doesn't get accounted for in your formControl. (Intentionally/Accidentally)

If you intend on not validating this field but still want to use the ngModel on this input element please add the flag to indicate it's a standalone component without a need for validation as mentioned by @Avenir above.

answered Aug 3 '17 at 18:31



saNiks

517 3 12

5 What if you want to use if for validation AND use it with ngModel at the same time though? – Paul Nov 23 '17 at 17:34

In my case I missed out to declare formControl for an element inside formGroup ! – Sagar Khatri Nov 28 '18 at 10:32

▲

OK, finally got it working: see <https://github.com/angular/angular/pull/10314#issuecomment-242218563>

10

In brief, you can no longer use `name` attribute within a `formGroup`, and must use `formControlName` instead

▼

answered Aug 25 '16 at 9:48



user2363245

580 1 5 13

▲

when you write **formcontrolname** Angular 2 do not accept. You have to write **formControlName**. it is about uppercase second words.

8

```
<input type="number" [(ngModel)]="myObject.name" formcontrolname="nameFormControl"/>
```

▼

if the error still continue try to set form control for all of object(myObject) field

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

between start `<form>` `</form>` for example: `<form [formGroup]="myForm" (ngSubmit)="submitForm(myForm.value)">` set form control for all input field `</form>`.

edited Jan 13 '17 at 14:53

answered Jan 13 '17 at 12:42



ethemsulan

1,899 21 19

2

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

    this.userInfoForm = new FormGroup({
        userInfoUserName: new FormControl({ value: '' },
        Validators.compose([Validators.required])),
        userInfoName: new FormControl({ value: '' },
        Validators.compose([Validators.required])),
        userInfoSurName: new FormControl({ value: '' },
        Validators.compose([Validators.required]))
    });

    <form [formGroup]="userInfoForm" class="form-horizontal">
        <div class="form-group">
            <label class="control-label"><i>*</i> User Name</label>
            <input type="text" formControlName="userInfoUserName" class="form-
control" [(ngModel)]="userInfo.userName">
        </div>
        <div class="form-group">
            <label class="control-label"><i>*</i> Name</label>
            <input type="text" formControlName="userInfoName" class="form-
control" [(ngModel)]="userInfo.name">
        </div>
        <div class="form-group">
            <label class="control-label"><i>*</i> Surname</label>
            <input type="text" formControlName="userInfoSurName" class="form-
control" [(ngModel)]="userInfo.surName">
        </div>
    </form>
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

edited Aug 15 '17 at 8:58

answered Feb 3 '17 at 8:34



barış çıracı

882 9 14

**If component has more than 1 form, register all controls and forms**

1

I needed to know why this was happening in a certain component and not in any other component.



The issue was that I had 2 forms in one component and the second form didn't yet have `[formGroup]` and wasn't registered yet since I was still building the forms.

I went ahead and completed writing both forms complete without leaving a input not registered which solve the issue.

edited Apr 29 '18 at 7:37

answered Apr 29 '18 at 7:20



Rust

5,013 6 31 56



I just got this error because I did not enclose all my form controls within a `div` with a `formGroup` attribute.

1

For example, this will throw an error



```
<div [formGroup]='formGroup'>
</div>
<input formControlName='userName' />
```

This can be quite easy to miss if its a particularly long form.

answered Mar 15 at 16:57



Stephen Paul

20.3k 10 54 52

0

Example:

This does not work because it uses the `[formGroup]` and `name` attribute.

```
<div [formGroup]="myGroup">
  <input name="firstName" [(ngModel)]="firstName">
</div>
```

You should replace the `name` attribute by `formControlName` and it will work fine like this following:

```
<div [formGroup]="myGroup">
  <input formControlName="firstName" [(ngModel)]="firstName">
</div>
```

edited Sep 5 at 14:35

[ProblemAnswerQue](#)

415 1 3 16

answered Feb 7 at 12:43

[SANA Abdoul Aziz](#)

120 1 10

---

a solution for this could also be if you're using a `[formGroup]="myGroup"` and internally reference with `[(ngModel)]` a property that is not in the `formGroup`. try adding `[ngModelOptions]="{ standalone: true }"` so you tell your compiler it may be excluded and it is a standalone property – [ProblemAnswerQue](#) Sep 30 at 14:06

---