

# Angular 2 creating reactive forms with nested components

Asked 2 years, 7 months ago   Active 1 year, 4 months ago   Viewed 16k times



31



11

My requirement is that I need to create a form with nested components. I am creating components for each form field means for textbox there will be one component, for radio button there will be another component like wise.

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

```
  <textbox-component></textbox-component>
```

```
  <radioButton-component></radioButton-component>
```

```
</form>
```

And I want to use Reactive forms for creating this form as I want my html to be untouched and have my form validations through typescript only.

But I cant find any solution how can we have reactive forms nested with components.

 angular    typescript    angular2-forms

asked Mar 1 '17 at 12:28



[Mahesh Saibalwar](#)

747   1   6   13

Maybe each component will keep it's own validators. At submit, each component emit the value to the parent component through service or event emitter... Not tested, but this should work – [mickdev](#) Mar 1 '17 at 12:35

I am not able to add FormControlName property(which is required for reactive forms) in the html inside child components, it is throwing error of parent directive FormGroup is not present(as it is present in the parent) – [Mahesh Saibalwar](#) Mar 1 '17 at 12:40

That's why each component should keep it's own FormControlName and validators. The parent component role will be instantiate the form and get all data at submit. The children component will fire valid or error event and inform the parent to allow or not the submission ... Take a look to [Nested Model Driven Form](#) – [mickdev](#) Mar 1 '17 at 12:48

I have been through this post, but it is very static example. I am generating forms by rendering components dynamically on the fly. Searched a lot but not

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

## 4 Answers



After my research & experiments I found one answer to my question, so answering it myself. If it saves someone's time then I will be happy.

47



If you want to create reactive forms with nested components then you can do as below

Here I am creating a form with two nested components one for textbox & other for radio button



Your parent component can be like this

```
<form [formGroup]="myForm">
  <child-textbox-component [parentFormGroup]="myForm">
  </child-textbox-component>
  <child-radio-button-component [parentFormGroup]="myForm">
  </child-radio-button-component>
</form>
```

We are passing FormGroup object as input to child components which has been created in the parent component as input to the child components, they will use this FormGroup object in their component to design specific control of the class

Your child components will be like this

child-textbox-component

```
<div class="form-group" [formGroup]="parentFormGroup">
  <label>
    {{control.caption}}
  </label>
  <input class="form-control" type="text" [title]="control.toolTip"
    [attr.maxlength]="control.width" [name]="control.name"
    [value]="control.defaultValue" [formControlName]="control.name"/>
</div>
```

child-radio-button-component

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

```

    {{control.caption}}
  </label>
  <div>
    <label *ngFor="let value of control.values; let idx = index"
      class="radio-inline" [title]="control.tooltip">
      <input type="radio" [name]="control.name" [formControlName]="control.name"/>
      {{ value }}
    </label>
  </div>
</div>

```

Here control is the model class holding data to be displayed for these child components.

This way you can have your form to be generated using nested components, so that you need not have your form (can say large form) in single component. You can break it down to as many sub components & form will be easy to create & maintain also using reactive forms of angular 2. You can also easily add validations too.

I followed these links before answering this

1. [something similar on stackoverflow](#)
2. [angular 2 dynamic forms](#)

edited May 23 '17 at 12:10



Community ♦

1 1

answered Mar 6 '17 at 7:03



**Mahesh Saibalwar**

**747** 1 6 13

9 How are you doing validations at the child components? Can we add validations to the form at child component or define all validations at parent? Can you post your .ts file code – [Manu Sharma](#) May 18 '17 at 0:32 ✎

16 This answer looks useful, but it is incomplete. Can you please post your TS code? Namely, how and where you set up the FormGroup object and its controls. – [ell](#) Jun 29 '17 at 17:46 ✎

2 You can follow this tutorial this awesome [toddmotto.com/...](#) – [Tabares](#) Aug 29 '17 at 19:10

I made a simple example based on this solution: [stackblitz.com/edit/angular-1drscm](#) (based on the switch example at [coryryan.com/blog/...](#)). – [bernardn](#) Mar 8 at 15:36

Additional note to Mhesh's answer you can build this same solution without injecting [ParentFormGroup] in the HTML. You can do this

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



This is really nice.

## Example

To take the existing solution, you can do the same thing, except:

Your parent component can be like this, without any additional parameters passed in

```
<form [formGroup]="myForm">
  <child-textbox-component></child-textbox-component>
  <child-radio-button-component></child-radio-button-component>
</form>
```

Note additionally you can set formgroups like this:

```
<form [formGroup]="myForm">
  <child-textbox-component></child-textbox-component>
  <child-radio-button-component formGroupName="myGroup"></child-radio-button-
component>
</form>
```

child-textbox-component

```
<div class="form-group" [formGroup]="controlContainer.control">
  <label>
    {{control.caption}}
  </label>
  <input class="form-control" type="text" [title]="control.toolTip"
    [attr.maxLength]="control.width" [name]="control.name"
    [value]="control.defaultValue" [formControlName]="control.name"/>
</div>
```

To enable this you want to inject a `ControlContainer` into your `@Component`

```
@Component({
  moduleId: `MODULE_ID_HERE`,
  selector: "child-textbox-component",
  templateUrl: "childTextbox.component.html",
})
```

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

```
}  
}
```

answered Jun 11 '18 at 17:13

**Stormswept****184** 2 12

1



To extend the list of possible answers, [this article by Alexey Zuev](#) suggests using `provide:ControlContainer` and `useExisting:NgForm` in the component decorator as a way to pass the `ngForm` directive into a child component.

```
@Component({  
  selector: 'registrant',  
  templateUrl: 'app/register/registrant.component.html',  
  **viewProviders: [ { provide: ControlContainer, useExisting: NgForm } ]**  
})
```

answered Jun 13 '18 at 20:35

**PoorInRichfield****428** 4 9

0



just pass same or sub `formGroup` in form block with `[formGroup]` binding.

edited Jan 5 '18 at 8:07

answered Jan 5 '18 at 7:42

**xsilen T****342** 3 5