

# Can't bind to 'ngModel' since it isn't a known property of 'input'



I've got the following error when launching my Angular app, even if the component is not displayed.

1016

I have to comment out the `<input>` so that my app works.



106

```
zone.js:461 Unhandled Promise rejection: Template parse errors:  
Can't bind to 'ngModel' since it isn't a known property of 'input'. ("  
    <div>  
        <label>Created:</label>  
        <input type="text" [ERROR ->][(ngModel)]="test" placeholder="foo" />  
    </div>  
</div>"): InterventionDetails@4:28 ; Zone: <root> ; Task: Promise.then ; Value:
```

★

I'm looking at the Hero plunker, but I don't see any difference from my code.

Here is the component file:

```
import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core';  
import { Intervention } from '../../../../../model/intervention';  
  
@Component({  
    selector: 'intervention-details',  
    templateUrl: 'app/intervention/details/intervention.details.html',  
    styleUrls: ['app/intervention/details/intervention.details.css']  
})  
  
export class InterventionDetails  
{  
    @Input() intervention: Intervention;  
  
    public test : string = "toto";  
}
```



Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google





Rov

131 10



Anthony Brenelière

23.6k 11 29 44

- 6 What is Hero plucker? – [Daniel Shillcock](#) Aug 11 '16 at 9:53
- 3 I'm having the same problem since they updated to rc5 yesterday (interestingly it works for my colleague..) I think @abreneliere is talking about their tutorials - [angular.io/docs/ts/latest/tutorial](#) – [PetLahev](#) Aug 11 '16 at 10:43
- 3 @DanielShillcock, [Tour of Heroes plunker](#). – [Mark Rajcok](#) Aug 12 '16 at 2:28
- 2 Yes I referred to the Tour of heroes tutorial because it does use a ngModel. – [Anthony Brenelière](#) Aug 12 '16 at 7:43
- 2 I'm just staring Angular and saw this error when doing the Tour of Heroes tutorial. Sure enough, they now [call this out in the very next line](#) and tell you how/why to correct it. – [Matt Penner](#) Dec 20 '17 at 8:43

## 26 Answers



Yes that's it, in the app.module.ts, I just added :

1464

```
import { FormsModule } from '@angular/forms';

[...]

@NgModule({
  imports: [
    [...]
    FormsModule
  ],
  [...]
})
```

edited Mar 1 '17 at 14:51



Jason Swett

21.7k 47 166 291

answered Aug 11 '16 at 12:30



Anthony Brenelière

23.6k 11 29 44

[Join Stack Overflow](#) to learn, share knowledge, and build your career.

[Email Sign Up](#)

OR SIGN IN WITH

Google

Facebook

adam0101 Feb 14 '18 at 16:34

- 1 In my case, My use of braces were wrong. I was using `[{ngModel}]` instead of the correct one: `[(ngModel)]` – [Imtiaz](#) Apr 14 '18 at 19:40 
- 17 For those who find this because of a similar failure with Jasmine, you'll need to add these imports to the `component.spec.ts` file as well. – [theMayer](#) Apr 24 '18 at 15:03
- 3 And for where to put it (regarding the Jasmine error): [stackoverflow.com/questions/39584534/...](https://stackoverflow.com/questions/39584534/) – [bagsmode](#) Oct 11 '18 at 18:10 



In order to be able to use two-way data binding for form inputs you need to import the `FormsModule` package in your Angular module. For more info see the Angular 2 official tutorial [here](#) and the official documentation for [forms](#)

470

edited Aug 11 '16 at 11:06



Srijith

1,351 7 14

answered Aug 11 '16 at 10:58



Gabriele Ciech

5,325 3 10 13

- 8 I have to admit that I start with Angular on Monday (so 4 days ago) and was doing their tutorials so I'm pretty sure I did something wrong but for me importing `FormsModule` didn't work. Then I added `import { FORM_DIRECTIVES } from '@angular/forms';` and added the `FORM_DIRECTIVES` to directives and yeah my binding is working again (to be clear it was working before rc5 release) – [PetLahev](#) Aug 11 '16 at 13:05
- 2 Which seems to be related to this [stackoverflow.com/questions/31623879/...](https://stackoverflow.com/questions/31623879/) – [PetLahev](#) Aug 11 '16 at 13:51
- 1 @PetLahev the issue you're having with the tutorial is that on August 7th, when you started, the tutorials were running Release Candidate 4. As of August 8th, they are at Release Candidate 5, which has different syntax – [AJ Zane](#) Aug 17 '16 at 4:10
- 4 `FORM_DIRECTIVES` are dead just in case – [Toolkit](#) Jan 31 '17 at 11:19
- 6 It's worth nothing that, if you are using a `SharedModule`, you might need to import `FormsModule` there aswell. – [Difinity](#) Feb 14 '17 at 20:23



For using `[(ngModel)]` in **Angular 2, 4 & 5+**, You need to import **FormsModule** from Angular form...

195

Also it is in this path under forms in Angular repo in [github](#):

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



Facebook

Also if you are using **ReactiveFormsModule**, needs to import it too.

So simply looks for **app.module.ts** and make sure you have `FormsModule` imported in...

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms'; //<<< import it here
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule, FormsModule //<<< and here
  ],
  providers: [],
  bootstrap: [AppComponent]
})

export class AppModule { }
```

Also this is the current starting comments for Angular4 `ngModel` in **FormsModule**:

```
/**
 * `ngModel` forces an additional change detection run when its inputs change:
 * E.g.:
 * ``
 * <div>{{myModel.valid}}</div>
 * <input [(ngModel)]="myValue" #myModel="ngModel">
 * ``
 * I.e. `ngModel` can export itself on the element and then be used in the template.
 * Normally, this would result in expressions before the `input` that use the exported
directive
 * to have an old value as they have been
 * dirty checked before. As this is a very common case for `ngModel`, we added this
second change
 * detection run.
 *
 * Notes:
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook

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

For more info, Look at **ng\_model** in Angular section [here](#)

edited Oct 15 '18 at 5:45

answered Oct 10 '17 at 12:46



Alireza

57.8k 14 192 126

- ▲ You need to import the **FormsModule**
- 77** Open **app.module.ts**
- ▼ and add line

```
import { FormsModule } from '@angular/forms';
```

and

```
@NgModule({
imports: [
  FormsModule
],
})
```

edited Nov 22 '18 at 6:10



Amit Chigadani

11.9k 5 36 58

answered Oct 6 '18 at 4:59



PRao

781 2 2

- 2 This worked perfect, just tested on Angular 6 on a CLI generated project – [siddharta](#) Jan 8 at 4:58

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



Facebook

If you'll be using the `FormsModule` ONLY in the `AuthModule` then you wouldn't need to import the `FormsModule` IN the default `AppModule`

So something like this in the `AuthModule`:

```
import { NgModule }      from '@angular/core';
import { FormsModule } from '@angular/forms';

import { authRouting } from './auth.routing';
import { LoginComponent, SignupComponent } from './auth.component';

@NgModule({
  imports: [
    authRouting,
    FormsModule
  ],
  declarations: [
    SignupComponent,
    LoginComponent
  ]
})
export class AuthModule { }
```

Then forget about importing in `AppModule` if you don't use the `FormsModule` anywhere else.

answered Sep 24 '16 at 13:06



Rexford

3,743 4 40 85

---

I needed it at sub-module where I was using the Form features. Higher up the hierarchy was not sufficient. – [Zarepheth](#) Nov 29 '18 at 21:33

you saved me!!! – [Hagai L](#) Jan 17 at 18:39

---

There are two steps you need to follow to get rid of this error

- 33 1. import `FormsModule` in your app module

[Join Stack Overflow](#) to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



```

import { NgModule }      from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule }   from '@angular/forms';
import { AppComponent }  from './app.component';
import {AppChildComponent} from './appchild.component';
@NgModule({
  imports:      [ BrowserModule,FormsModule ],
  declarations: [ AppComponent, AppChildComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }

```

Hope it helps

answered Dec 9 '16 at 7:00




---

Thank you! I'd not declared it in the imports section and it was driving me insane when I moved a component into a sub-module. – [Richard](#) Mar 4 '17 at 15:49

---

import **FormsModule** in you app module.

21

it would let your application running well.

```

import { NgModule }      from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import {ContactListCoppmonent} from './contacts/contact-list.component';
import { FormsModule }   from '@angular/forms';

import { AppComponent }  from './app.component';

@NgModule({
  imports: [
    BrowserModule,

```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Facebook

```
})
export class AppModule { }
```

answered Dec 25 '16 at 9:43



Shashikant Pandit  
1,049 9 15

 You need to import **FormsModule** in your **root** module if this component is in the root i.e. **app.module.ts**

**21** Kindly open app.module.ts

 Import **FormsModule** from **@angular/forms**

Ex:

```
import { FormsModule } from '@angular/forms';
```

and

```
@NgModule({
  imports: [
    FormsModule
  ],
})
```

answered Apr 24 at 8:06



WapShivam  
280 2 9

1 Thanks a lot. it saved my time – [Rejwanul Reja](#) May 16 at 10:51

Most Welcome!!! – [WapShivam](#) Jun 7 at 10:16

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



Facebook

**19** *app.module.ts*

1. import import {FormsModule} from "@angular/forms";  
 2. add in imports imports: [  
   BrowserModule,  
   FormsModule  
   ],

*app.component.ts*

1. Example: <input type="text" [(ngModel)]="name" >
2. and then <h1>your name is: {{name}} </h1>

edited Nov 14 '17 at 9:40


**Massimiliano Kraus**  
 2,482 4 16 33

answered Oct 29 '17 at 5:05


**iGhost**  
 199 1 7
**15** I'm using Angular 5.

In my case, I needed to import ReactiveFormsModule too.

*app.module.ts* (or anymodule.module.ts)

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    ReactiveFormsModule
  ],
  declarations: []
})
```

edited May 9 '18 at 18:44

answered May 4 '18 at 22:19

Join Stack Overflow to learn, share knowledge, and build your career.

[Email Sign Up](#)

OR SIGN IN WITH



ReactiveFormsModule is required when you need FormGroup, FormControl ,Validators etc. For using ngModel , you don't require ReactiveFormsModule . – Amit Chigadani Nov 22 '18 at 6:08

@AmitChigadani Adding ReactiveFormsModule is the only thing that made the error go away in my case. I'm using Angular 7. – Eternal21 May 29 at 0:25

## I am using Angular 7

**15** I have to import ReactiveFormsModule because I am using FormBuilder class to create reactive form.

```
import {
FormsModule,
ReactiveFormsModule } from '@angular/forms';

@NgModule({
imports: [
CommonModule,
FormsModule,
ReactiveFormsModule
], declarations: []})
```

answered Feb 5 at 8:29



334 2 9

## Simple Solution : In app.module.ts

**13**

```
***Example 1***
import {FormsModule} from "@angular/forms";
// add in imports

imports: [
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook

If you want to use [(ngModel)] then you have to import FormsModule in app.module.ts

```
import { FormsModule } from "@angular/forms";
@NgModule({
  declarations: [
    AppComponent, videoComponent, tagDirective,
  ],
  imports: [
    BrowserModule, FormsModule
  ],
  providers: [ApiServices],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

edited Feb 18 '18 at 2:58



Paresh Gami

3,771 3 16 32

answered Jan 25 '18 at 11:09



ngCourse

1,050 9 12

17 Where exactly is the difference in the two examples? – Philipp Meissner Apr 6 '18 at 12:35

12

If someone is still getting errors after applying the accepted solution, it could be possibly because you have a separate module file for the component in which you want to use the ngModel property in input tag. In that case, apply the accepted solution in the component's module.ts file as well.

answered May 11 '17 at 11:18



Subham Pasari

121 1 5

I know this question is about Angular 2, but I am on Angular 4 and none of these answers helped.

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



Hope this helps.

answered Aug 4 '17 at 5:49



Matt

16.5k 13 50 64

 if you are still getting the error after importing FormsModule correctly then check in your terminal or (windows console) because your project is not compiling (because of another error that could be anything) and your solution has not been reflected in your browser!

12

 In my case, my console had the following unrelated error: Property 'retrieveGithubUser' does not exist on type ' ApiService'.

edited Nov 30 '18 at 15:03

answered Jul 26 '17 at 16:50



mruanova

2,171 2 15 36

 In the module you are willing to use **ngModel** you have to import **FormsModule**

10

```
import { FormsModule } from '@angular/forms';

@NgModule({
  imports: [
    FormsModule,
  ],
})

export class AbcModule {}
```

answered Jul 17 '18 at 4:59



Malindu Sandaruwan

1,130 1 5 18

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



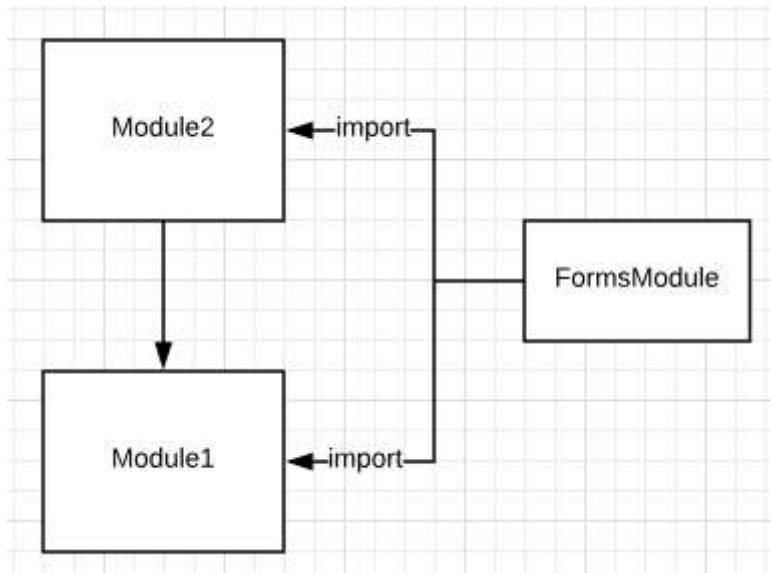
Google



Facebook

8

1. You didn't import the FormsModule into import array of module where your component is declared, component in which the ngModel is used.
2. You have import the FormsModule into one module which is inherited of another module. In this case you have two options:
  - let the FormsModule imported into the import array from both modules:module1 and module2. **As rule: Importing a module does NOT provide access to its imported modules.(Imports are not inherited)**



- declare the FormsModule into the import and export arrays in module1 to be able to see it in module2 also

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

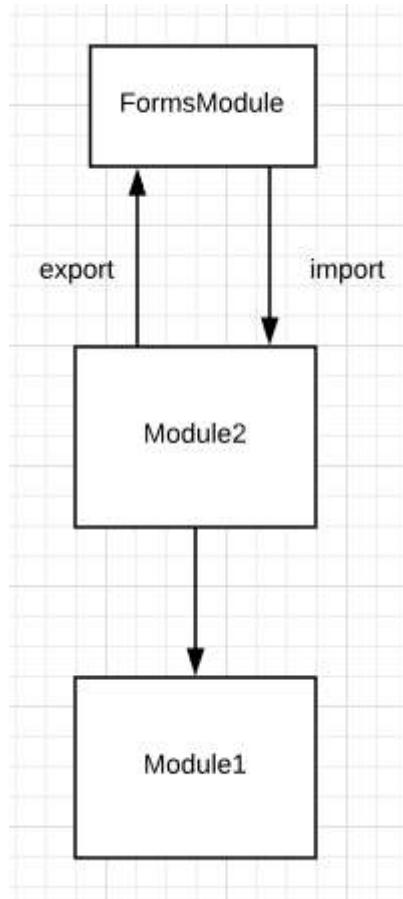
OR SIGN IN WITH



Google



Facebook



3. (In some version I faced this problem) You have imported correctly the FormsModule but the problem is on the input HTML tag. You must add the name tag attribute for input and the object bound name in [(ngModel)] must be the same as the name into the *name* attribute

```
<input type="text" [(ngModel)]="test" name="test" placeholder="foo" />
```

answered Nov 13 '18 at 16:20



Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



7

```

import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule {}

```

edited Nov 14 '17 at 9:40



Massimiliano Kraus  
2,482 4 16 33

answered Oct 11 '17 at 3:56



William Pourmajidi  
181 1 6

6

```

import { NgModule } from '@angular/core'
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { MyComponent } from './mycomponent'

@NgModule({
  imports: [
    CommonModule,
    FormsModule
  ],
  declarations: [
    MyComponent
  ]
})

```

Join Stack Overflow to learn, share knowledge, and build your career.

[Email Sign Up](#)

OR SIGN IN WITH



Google



Facebook



387 5 15

This is for the folks who use plain JavaScript instead of Type Script. In addition to referencing the forms script file on top of the page like below

5

```
<script src="node_modules/@angular/forms/bundles/forms.umd.js"></script>
```

you should also tell the module loader to load the `ng.forms.FormsModule`. After making the changes my `imports` property of `NgModule` method looked like below

```
imports: [ng.platformBrowser.BrowserModule, ng.forms.FormsModule],
```

Happy coding!

answered Dec 12 '16 at 0:37



James Poulose

1,846 16 21

Sometimes you get this error when you try to use a component from a module, which is not shared, in a different module.

5

For example, you have 2 modules with `module1.componentA.component.ts` and `module2.componentC.component.ts` and you try to use the selector from `module1.componentA.component.ts` in a template inside `module2` (e.g. `<module1-componentA [someInputVariableInModule1]="variableFromHTTPRequestInModule2">` ), it will throw the error that the `someInputVariableInModule1` is not available inside `module1.componentA.component.ts` - even though you have the `@Input() someInputVariableInModule1` in the `module1.componentA`.

If this happens, you want to share the `module1.componentA` to be accessible in other modules. So if you share the `module1.componentA` inside a `sharedModule`, the `module1.componentA` will be usable inside other modules (outside from `module1`), and every module importing the `sharedModule` will be able to access the selector in their templates injecting the `@Input()` declared variable.

Join Stack Overflow to learn, share knowledge, and build your career.

[Email Sign Up](#)

OR SIGN IN WITH



Google



- 1 I have this exact problem, and you say, you should share the module so you can use componentA in ComponentC. But.. How do I share the module1? Which is the syntax to "share a module"? I guessed exporting module1 and then importing it in the module2 was enough sharing, but still having this problem – [Agorilla](#) Sep 5 '17 at 11:25 
  - 2 So to answer myself, this is how you share a module [angular-2-training-book.rangle.io/handout/modules/...](#) – [Agorilla](#) Sep 5 '17 at 12:06 
  - 2 Hey sorry i didn't see your comment that fast :) Yes, this is how you can create a SharedModule and import your own modules there, which you want to use in multiple other modules. Then you import the SharedModule where you want to use the module which is imported in the shared one. – [Guntram](#) Sep 5 '17 at 13:02
- 



You need to import the FormsModule

5



Open **app.module.ts**

and add line

```
import { FormsModule } from '@angular/forms';
```

and

```
@NgModule({
  imports: [
    FormsModule
  ],
})
```

answered Apr 22 at 10:41



When I first did the tutorial, main.ts looked slightly different from what it is now. It looks very similar, but note the differences (the top one is correct).

[Join Stack Overflow](#) to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Facebook

```
platformBrowserDynamic().bootstrapModule(AppModule);
```

Old tutorial code:

```
import { bootstrap }    from '@angular/platform-browser-dynamic';
import { AppComponent } from './app.component';
bootstrap(AppComponent);
```

answered Aug 12 '16 at 2:52



Jordan Lapp  
822 8 18



I upgraded from RC1 to RC5 and received this error.

4

I completed my migration (introducing a new `app.module.ts` file, changing `package.json` to include new versions and missing modules, and finally changing my `main.ts` to boot accordingly, based on the [Angular2 quick start example](#)).



I did an `npm update` and then an `npm outdated` to confirm the versions installed were correct, still no luck.

I ended up completely wiping the `node_modules` folder and reinstalling with `npm install` - Voila! Problem solved.

answered Aug 24 '16 at 4:42



Rots  
4,823 2 35 47



In `NgModule` you need to import `FormsModule`, because `ngModel` is coming from `FormsModule`. Please modify your `app.module.ts` like the below code I have shared

4

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
```

[Join Stack Overflow](#) to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



Facebook

```
imports: [
  BrowserModule,
  AppRoutingModule,
  FormsModule
],
bootstrap: [AppComponent]
})
export class AppModule { }
```

edited Jun 21 at 5:21

answered Jun 21 at 5:00



Vicky Khichar

108 6

For any version from Angular 2, you need to import FormsModule in your app.module.ts file and it will fix the issue.

0

answered Aug 3 '18 at 14:41



Siddhartha Thota

472 4 14

**protected** by Günter Zöchbauer Jan 22 '18 at 8:42

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these [unanswered questions](#) instead?

Join Stack Overflow to learn, share knowledge, and build your career.

[Email Sign Up](#)

OR SIGN IN WITH



Google



Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Facebook