

Use Stack Overflow for Teams at work to find answers in a private and secure environment. Get your first 10 users free. [Sign up.](#)

# Angular2 @Input to a property with get/set

Asked 3 years, 5 months ago   Active 6 months ago   Viewed 138k times



123



20

I have an Angular2 component in that component it currently has a bunch fields that have @Input() applied before them to allow binding to that property, i.e.

```
@Input() allowDay: boolean;
```

What I would like to do is actually bind to a property with get/set, so that I can do some other logic in the setter, something like the following

```
_allowDay: boolean;  
get allowDay(): boolean {  
    return this._allowDay;  
}  
set allowDay(value: boolean) {  
    this._allowDay = value;  
    this.updatePeriodTypes();  
}
```

how would I do this in Angular2?

Based on Thierry Templier suggestion I changed it to, but that throws the error Can't bind to 'allowDay' since it isn't a known native property :

```
//@Input() allowDay: boolean;  
_allowDay: boolean;  
get allowDay(): boolean {  
    return this._allowDay;  
}  
@Input('allowDay') set allowDay(value: boolean) {  
    this._allowDay = value;  
    this.updatePeriodTypes();  
}
```

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



edited Mar 13 at 10:29



simbabque

45.2k

7

59

108

asked Apr 15 '16 at 17:44



Paul Cavacas

1,122

2

16

38

How and where to do you bind to `[allowDay]="...."`. If the field (setter) name and the property name you want to use for binding are the same, you can omit the parameter for `@Input(...)`. – [Günter Zöchbauer](#) Apr 16 '16 at 10:29

I would be curious to see how yo set up your unit test if you went the route of using get set as shown in the accepted answer. – [Winnemucca](#) Jan 8 '18 at 22:12

- 1 Whatever you end up doing make sure to put a breakpoint, or debug statement, or counter inside your setter to make sure it is only firing once as expected. I just found mine was being updated for every change detection run causing horrible performance and quirky behavior. – [Simon\\_Weaver](#) Feb 14 '18 at 23:53

## 4 Answers



You could set the `@Input` on the setter directly, as described below:

193



```
_allowDay: boolean;
get allowDay(): boolean {
  return this._allowDay;
}

@Input('allowDay')
set allowDay(value: boolean) {
  this._allowDay = value;
  this.updatePeriodTypes();
}
```

See this plunker: <https://plnkr.co/edit/6miSutgTe9sfEMCb8N4p?p=preview>.

edited Apr 16 '16 at 6:57

answered Apr 15 '16 at 17:48



Thierry Templier

157k

29

342


311

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

Are you sure? It works for me. I added a plunkr. Perhaps you forgot to add the directive into the `directives` attribute of the component where you want to use it... I updated my answer. – [Thierry Templier](#) Apr 16 '16 at 6:57

1 This is a bad idea because if you use the setter, `ngOnChanges` doesn't fire. – [user2867288](#) Feb 14 '18 at 15:44

3 [On angular Fundamentals pages](#) – [PaulCo](#) Mar 15 '18 at 12:40

7 **WARNING:** The `setter` will **NOT** be triggered by mutations to values which are passed by reference (i.e. pushing to an array, mutating an object, etc.). You would need to replace the whole value being passed as an `Input` for the `setter` to trigger again. – [Nickofthyme](#) Nov 16 '18 at 14:54 

If you are mainly interested in implementing logic to the **setter only**:

44

```
import { Component, Input, OnChanges, SimpleChanges } from '@angular/core';
```

```
// [...]
```

```
export class MyClass implements OnChanges {
  @Input() allowDay: boolean;

  ngOnChanges(changes: SimpleChanges): void {
    if(changes['allowDay']) {
      this.updatePeriodTypes();
    }
  }
}
```

The import of `SimpleChanges` is not needed if it doesn't matter which input property was changed or if you have only one input property.

[Angular Doc: OnChanges](#)

**otherwise:**

```
private _allowDay: boolean;

@Input() set allowDay(value: boolean) {
  this._allowDay = value;
  this.updatePeriodTypes();
}
```

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

edited Apr 12 '18 at 7:56

answered Nov 23 '16 at 12:04

**Martin Schneider**

5,636 1 38 48

Just curious, is there any benefit for using ngOnChanges vs not using the set property if you are only interested on a setter logic? – Mese Jun 22 '17 at 8:52

- 4 There is no difference in "using ngOnChanges vs not using set"... ;) Joking aside: One benefit is, if your component has multiple @Input properties and you want to call a routine when any of them has changed. So less code needed. – Martin Schneider Jun 22 '17 at 14:35 ✎

Ups, had a typo hehe. But ok, thought it might had more relevance. Thanks for the answer tho :) – Mese Jun 23 '17 at 8:25

- 1 @MA-Maddin I suppose you could also set a debounced observable if you were expecting multiple changes all at the same time that would each result in a routine needing to be run. – Simon\_Weaver Jan 18 '18 at 23:02

ngOnChanges approach is great!! Good answer. If the value being set cannot be private e.g. it's used as binding in the template, The \_propertyName setter/private naming convention becomes inconsistent . ngOnChanges gets around this perfectly – Drenai May 15 '18 at 19:32

@Paul Cavacas, I had the same issue and I solved by setting the Input() decorator above the getter.

6

```
@Input('allowDays')
get in(): any {
  return this._allowDays;
}

//@Input('allowDays')
// not working
set in(val) {
  console.log('allowDays = '+val);
  this._allowDays = val;
}
```

See this plunker: <https://plnkr.co/edit/6miSutgTe9sfEMCb8N4p?p=preview>

edited Nov 21 '16 at 20:18

**George Kagan**

answered Nov 21 '16 at 19:52

**maxi-code**

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

9/18/2019

angular - Angular2 @Input to a property with get/set - Stack Overflow

0 This bug got me crazy, I finally found that you should define the input() first (getter or setter but the input decorator must go first) – [maxi-code](#) Nov 21 '16 at 20:23

1 Here's other reference that may help <https://github.com/angular/angular/issues/5477> – [maxi-code](#) Nov 21 '16 at 20:23



1



Updated accepted answer to angular 7.0.1 on stackblitz here: <https://stackblitz.com/edit/angular-inputsetter?embed=1&file=src/app/app.component.ts>

directives are no more in Component decorator options. So I have provided sub directive to app module.

thank you [@thierry-templier](#)!

answered Dec 17 '18 at 13:41



[Luckylooke](#)

1,921 1 19 29

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