Conditionally make input field readonly in Angular 2 or 4: Advice + Best/which way to do it

21

5

I was attempting to answer someone elses question. And in doing so realised there was quite a bit of uncertainty in my mind about a few things. I'm hoping someone can provide feedback on the numbered points 1..4:

Task: Conditionally make input field readonly

Relevant section of HTML:

```
<input type="text" placeholder="Club Name" #clubName>
```

Add this to Typescript component.

```
// class properties
@ViewChild('clubName')
inp:HTMLInputElement; // Could also use interface Element

// conditionally set in some other methods of class
inp.setAttribute('readonly', 'readonly');
inp.removeAttribute('readonly');
```

Have to say this is a grey area for me.

1. Is referencing `HTMLInputElement` or `Element` directly with `@ViewChild` in Angular 2+ a bad practice? Only, I've often seen examples using `ElementRef` or chaining off to `nativeElement` from `ElementRef` .

Since VS Studio doesn't have Intelli-sense for those, I suddenly feel like I'm coding in the dark. i.e, you never get feedback about methods setAttribute or removeAttribute, their parameter requirements etc. (I'm aware of As to cast too)

IIRC I think you have to do this way with a property get in Typescript:

```
get isReadOnly() :boolean {
}
```

Is this way valid?

3. I'm wondering, can you do the method syntax too in the HTML template:

```
<input [attr.readonly]= "isReadOnly()">
```

Typescript

```
isReadOnly() :boolean {
}
```

Is this way valid?

4. In summary, what's the best approach?

Update: There is also *ngIF so you output one of two input elements with same name. But that sounds to me like a sledgehammer to crack a nut.

angular readonly attr html-input

edited Jul 21 '17 at 7:56

asked Jul 20 '17 at 22:43

JGFMK
4,016 4 32 66

3 Answers

46

✓

You need to use the following (Angular 4):

```
<input [readonly]="isReadOnly">
```

If you use `att.readonly` then the input will always be read-only because the `readonly` attribute will be present even if its value is false. By using `[readonly]` Angular will only place the attribute if `isReadOnly` is true.

In HTML, the following is sufficient to cause an input to be read-only:

```
<input readonly>
```

answered Nov 6 '17 at 13:42

phn
761 7 9

1

You can use `<input readonly="{{ variable }}>"` .

In your ***.component.ts**, initialize the variable:

```
private variable: boolean = true;
```

answered Oct 5 '18 at 19:51

Rodrigo Pauletti
126 3 9

All depends on what you want to achieve. At first look, I would say that pct. 2 is the simplest way to do it:

```
isReadOnly: boolean;
```

After, you assign the value as you wish:

```
// when you want to be read-only
isReadOnly = true;
// whe you want to be editable
isReadOnly = false;
```

answered Jul 20 '17 at 23:43
BogdanC
1,657 2 16 15