

Angular2, TypeScript, How to read/bind attribute value to component class (undefined in ngOnInit) [duplicate]

Asked 3 years, 7 months ago Active 1 year, 5 months ago Viewed 43k times



15

This question already has an answer here:

[Angular 2 external inputs](#) 4 answers



can someone please advice me how to read/bind attribute value to @component class, which seems to be undefined in ngOnInit method?



4

Here's a plunker demo: <http://plnkr.co/edit/4FoFNBFsOEvvOkyfn0lw?p=preview>

I'd like to read value of "someattribute" attribute

```
<my-app [someattribute]='somevalue'>
```

inside the App class (src/app.ts) ngOninit method.

Thanks!

typescript



angular

angular2-template

angular2-directives

edited Feb 25 '16 at 14:06

asked Feb 25 '16 at 12:59



Tomino

3,489

5

24

40

marked as duplicate by [Günter Zöchbauer](#)

angular2

Nov 29 '16 at 7:47

This question has been asked before and already has an answer. If those answers do not fully address your question, please [ask a new question](#).

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

3 Answers



15



You can notice that **such parameters can't be used for root component**. See this question for more details:

- [Angular 2 input parameters on root directive](#)

The workaround consists in leveraging the `ElementRef` class. It needs to be injected into your main component:

```
constructor(elm: ElementRef) {  
  this.someattribute = elm.nativeElement.getAttribute('someattribute');  
}
```

We need to use the component this way in the HTML file:

```
<my-app someattribute="somevalue"></my-app>
```

edited May 23 '17 at 11:46



Community ♦

1 1

answered Feb 25 '16 at 13:00



Thierry Templier

158k 29 342 311

Sorry, it was a typo. This doesn't work as well. Please see the plunker. – [Tomino](#) Feb 25 '16 at 13:01

No worries! I updated my answer... The problem is because you try to use this on the main component of your application (the one you use to bootstrap it) – [Thierry Templier](#) Feb 25 '16 at 13:04

Yes! That's it... I did not realized that I cannot use attributes in a root component. Can you please remove 'interpolation' part of your answer and keep just part below 'code'? Just to avoid confusion of users like me. – [Tomino](#) Feb 25 '16 at 13:07

This is a bad idea because of the security ricks and coupling to the DOM as noted in the API. angular.io/docs/js/latest/api/core/index/ElementRef-class.html use the `@Input` decorator – [Michael Warner](#) Feb 24 '17 at 21:47

The security risks pertain to DOM manipulation. If you read further into the security guide ([angular.io/guide/...](https://angular.io/guide/)) it's the `ElementRef`'s `node` that has unsafe methods. Simply reading attributes from a DOM element is not an inherit security risk. You can also use a third-party library like StencilJS which supports creating custom properties on components: stenciljs.com/docs/properties – [CallieLinux](#) Aug 18 at 15:13

14

This is an example:

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

@Component({
  selector: 'user-menu',
  templateUrl: 'user-menu.component.html',
  styleUrls: ['user-menu.component.scss'],
})
export class UserMenuComponent {

  /**
   * userName Current username
   */
  @Input('userName') userName: string;

  constructor() {

  }

  sayMyName() {
    console.log('My name is', this.userName);
  }
}
```

And to use it

```
<user-menu userName="John Doe"></user-menu>
```

edited Apr 23 '18 at 6:55



Sunil Garg

5,249 9 60 90

answered Oct 14 '16 at 13:39



Germanaz0

701 4 17

1 this is a much better approach than reading the getAttribute if not used on the root element – [darwin](#) Mar 20 '18 at 9:30 ✎

1 This must be accepted answer. A proper way to get attribute. – [Rafique Mohammed](#) Aug 30 '18 at 11:10

This answer doesn't address the issue with creating a custom attribute on a root component. While this solution is file for a parent/child component property binding, the accepted answer is correct for what the scenario (root component). – [CallieLinux](#) Aug 18 at 15:10

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

4

Inputs aren't supported in the root component as a workaround you can use

```
constructor(elementRef:ElementRef) {  
  console.log(elementRef.nativeElement.getAttribute('someattribute'));  
}
```

See also <https://github.com/angular/angular/issues/1858>

See also the fixed [Plunker](#)

original

You need to either use

```
[property]="value"
```

or

```
property="{{value}}"
```

or if it's an attribute

```
[attr.property]="value"
```

or

```
attr.property="{{value}}"
```

edited Feb 25 '16 at 13:07

answered Feb 25 '16 at 13:00



Günter Zöchbauer

369k 85 1179 1070

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