

# Can an angular 2 component be used with an attribute selector?

Asked 3 years, 1 month ago   Active 4 months ago   Viewed 35k times



We currently have an existing small Angular 1 project that is used in an on premises Sharepoint 2013 environment. For a large part of our content, we use publishing pages on the Sharepoint environment.

25



With Angular 1, we could define directives to be restricted to: *match attribute name, tag name, comments, or class name*. Most of the directives we created were attribute or tag name. The preference would have been tag name, but the publishing platform on Sharepoint strips out **unknown** elements. So that means we were left with using attributes in order to bring our directives in to the publishing pages.



With Angular 2 though, I've only seen components implemented by tag name.

9

Is it possible with Angular 2 to use attribute names in order to use our components? This is a requirement for us because of the restrictions in the Sharepoint publishing platform.

Thanks.



edited May 4 at 22:05



cale\_b

20.7k

6

52

89

asked Jul 20 '16 at 3:29



TehOne

1,539

3

20

36

## 4 Answers



Yes, the `selector` property of the `@Component` decorator is a [CSS selector](#) (or a subset of):

47



```
selector : '.cool-button:not(a)'
```



Specifies a CSS selector that identifies this directive within a template. Supported selectors include `element` , `[attribute]` , `.class` ,

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

**Source:** [Angular Cheat Sheet / Directive Configuration](#), which `@Component` inherits.

That way you can use `[name-of-the-attribute]` (namely, the [CSS attribute selector](#)), such as:

```
@Component({
  selector: "[other-attr]",
  ...
})
export class OtherAttrComponent {
```

See [demo plunker here](#).

The usual way is the [CSS type \(AKA element or tag\) selector](#):

```
@Component({
  selector: "some-tag",
  ...
})
```

And it matches a tag with name `some-tag`.

You can even have a component that matches [both a tag or an attribute](#):

```
@Component({
  selector: "other-both,[other-both]",
  template: `this is other-both ({ { value } })`
})
export class OtherBothComponent {
```

[Demo plunker](#) contains examples of all three.

## Is `[attributeName="attributeValue"]` supported?

Yes. But mind the quotes. In the current implementation, the selector `[attributeName="attributeValue"]` actually matches `<some-tag attributeName='attributeValue'>`, so test around before committing to this approach.

edited Apr 13 '17 at 1:29

answered Jul 20 '16 at 4:12

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

Note: since any CSS selector is valid, you could even use class selectors such as `selector: '.my-class-name'`, which would match any elements with such class. – [acdcjunior](#) Jul 20 '16 at 4:27

Great. thanks. I didn't see any reference to being able to pull this off when going through the Angular 2 site. So glad this is possible. – [TehOne](#) Jul 20 '16 at 20:47

2 is `[attributeName="attributeValue"]` supported? – [Brad Kent](#) Nov 15 '16 at 17:27

2 The linter that comes with Angular CLI complains about components as attributes. Is there a way to disable that rule on a per file basis? – [Votemike](#) Jan 25 '17 at 18:27

2 @Votemike I haven't found a way to disable the error but if you add an element selector, in addition to the attribute selector, the error goes away. e.g., `selector: "my-selector,[my-selector]"` – [mfink](#) Jun 14 '18 at 19:43



9



Yes, according to this <https://angular.io/docs/ts/latest/guide/cheatsheet.html> (Components and Directives are very similar in general). Instead of using element selector:

```
selector: 'custom-element-name'
```

Use:

```
selector: '[custom-attribute-name]'
```

And in your parent component's template:

```
<div custom-attribute-name></div>
```

answered Jul 20 '16 at 3:41



[Harry Ninh](#)

10.1k 3 46 46



Selector property of `@Component` decorator support, element selector, attribute selector and class selector:

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

```
@Component({  
  selector: 'app-servers'  
})
```

**Usage:** <app-servers></app-servers>

## 2. Attribute selector:

```
@Component({  
  selector: '[app-servers]'  
})
```

**Usage:** <div app-servers></div>

## 3. Class selector:

```
@Component({  
  selector: '.app-servers'  
})
```

**Usage:** <div class="app-servers"></div>

**Note:** Angular 2 does not support id and pseudo selectors

edited Apr 3 at 1:49



JulienD

4,147

5

31

61

answered Dec 16 '17 at 10:43



Ankit Chaurasia

301

3

5

---

Well explained thanks :) – [Brahim LAMJAGUAR](#) Jun 14 '18 at 9:47

---

▲ Absolutely. Essentially this is just a CSS selector, so if you need to use attribute you just do this:

6

```
@Component({
```

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

answered Jul 20 '16 at 3:43

[Alexander Leonov](#)

3,214 1 12 22

---

And that would be selected by: `<my-tag with-my-attribute></my-tag>` but not by `<my-tag></my-tag>` Correct? – [Roger](#) Jun 6 at 0:09 

---

Yes, you are correct. – [Alexander Leonov](#) Jun 6 at 10:39

---

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