

How are we doing? Please help us improve Stack Overflow. [Take our short survey](#)

Pass enums in angular2 view templates

Asked 3 years, 7 months ago Active 2 months ago Viewed 51k times



Can we use enums in an angular2 view template?

84

```
<div class="Dropdown" dropdownType="instrument"></div>
```



passes the string as input:



9

```
enum DropdownType {  
    instrument,  
    account,  
    currency  
}  
  
@Component({  
    selector: '[.Dropdown]',  
})  
export class Dropdown {  
  
    @Input() public set dropdownType(value: any) {  
  
        console.log(value);  
    };  
}
```

But how to pass an enum configuration? I want something like this in the template:

```
<div class="Dropdown" dropdownType="DropdownType.instrument"></div>
```

What would be the best practice?

Edited: Created an example:

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

```
import {Component, View, Input} from 'angular2/core';

export enum DropdownType {

    instrument = 0,
    account = 1,
    currency = 2
}

@Component({selector: '[.Dropdown]',})
@View({template: ''})
export class Dropdown {

    public dropdownTypes = DropdownType;

    @Input() public set dropdownType(value: any) {console.log(`-- dropdownType:
${value}`)};
    constructor() {console.log('-- Dropdown ready --');}
}

@Component({ selector: 'header' })
@View({ template: '<div class="Dropdown" dropdownType="dropdownTypes.instrument">
</div>', directives: [Dropdown] })
class Header {}

@Component({ selector: 'my-app' })
@View({ template: '<header></header>', directives: [Header] })
class Tester {}

bootstrap(Tester);
```

enums



angular

angular2-template

edited May 17 at 5:14



Moothor

994 2 12 32

asked Mar 10 '16 at 17:50



McLac

873 2 9 16

- 2 Better than both of the answers below, though similar but simpler than the accepted one, is: stackoverflow.com/a/42464835/358578 – pbarranis Jul 12 '18 at 17:51

Q A

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

▲
97

Create a property for your enum on the parent component to your component class and assign the enum to it, then reference that property in your template.



```
export class Parent {
  public dropdownTypes = DropdownType;
}

export class Dropdown {
  @Input() public set dropdownType(value: any) {
    console.log(value);
  };
}
```

This allows you to enumerate the enum as expected in your template.

```
<div class="Dropdown" [dropdownType]="dropdownTypes.instrument"></div>
```

edited May 17 at 7:52



Moother

994 2 12 32

answered Mar 10 '16 at 18:29



David L

24.7k 6 46 76

-
- 2 Based on your update, move your enum property declaration to the parent component. – [David L](#) Mar 10 '16 at 19:48
-
- 3 Downvoter, care to explain? – [David L](#) Mar 18 '16 at 18:08
-
- 6 Again, downvoter, please provide feedback on how this answer can be improved if you disagree with it. – [David L](#) Jul 13 '16 at 15:24
-
- 2 downvote is a mistake - answer is correct – [McLac](#) Jul 13 '16 at 17:12
-
- 2 Pretty sure `dropdownType` in the template should have square brackets on both ends (like so: `[dropdownType]`) since it takes a var and not text. – [Tom](#) Dec 27 '18 at 10:03
-



Create an enum

135

```
enum ACTIVE_OPTIONS {
```

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

Create your component, be sure your enum list will have the **typeof**

```
export class AppComponent {
  ACTIVE_OPTIONS = ACTIVE_OPTIONS;
  active:ACTIVE_OPTIONS;
}
```

Create your view

```
<li [ngClass]="{'active':active===ACTIVE_OPTIONS.HOME}">
  <a router-link="/in">
    <i class="fa fa-fw fa-dashboard"></i> Home
  </a>
</li>
```

edited Jul 23 at 15:04



Ploppy

7,658 2 28 41

answered Oct 14 '16 at 2:44



Oswaldo Alvarez

3,702 1 17 20

-
- 4 Better solution than the accepted one. I guess it uses some new TS feature. – [Greg Dan](#) Jan 14 '17 at 14:57
-
- 2 Not a specialist myself, so I really have to question: is this solution always better than David L.'s? This one takes less lines of code, but in terms of memory usage it may be creating one list per instance of the host component class... And if this is true (not saying it is!), there's not much of a problem when dealing with AppComponent, but the solution might not be the best in the case of a CustomerComponent or something more recurrent. Am I right? – [Rui Pimentel](#) Jan 26 '17 at 13:46
-
- 2 You could update the html as: [class.active]="active === ACTIVE_OPTIONS.HOME" – [Neil](#) Mar 15 '18 at 17:24
-
- 6 how and why is this better than the accepted solution @GregDan? – [Aditya Vikas Devarapalli](#) Jun 27 '18 at 0:38

Aditya, it is better for the simple reason, that it involves one class, not 2. I don't have a parent class, and not going to create it for that reason :) – [Yuri Gridin](#) Jan 18 at 19:03



Maybe you don't have to do this.

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

```
export enum DropdownType {  
    instrument = 0,  
  
    account = 1,  
    currency = 2  
}
```

In HTML Template:

```
<div class="Dropdown" [dropdownType]="1"></div>
```

result: dropdownType == DropdownType.account

or String Enum:

```
export enum DropdownType {  
    instrument = "instrument",  
    account = "account",  
    currency = "currency"  
}
```

```
<div class="Dropdown" [dropdownType]=" 'currency' "></div>
```

result: dropdownType == DropdownType.currency

If you want get Enum name :

```
val enumValue = DropdownType.currency  
DropdownType[enumValue] // print "currency", Even the "numeric enum" is also.
```

answered Jun 4 at 11:09



zbin

31 4