

Cannot approach Typescript enum within HTML

Asked 2 years, 3 months ago Active 3 months ago Viewed 22k times



I made an enum with Typescript to use in MyService.service.ts MyComponent.component.ts and MyComponent.component.html.

39



```
export enum ConnectionResult {  
    Success,  
    Failed  
}
```



4

I can easily get and compare a defined enum variable from MyService.service.ts:

```
this.result = this.myService.getConnectionResult();  
  
switch(this.result)  
{  
    case ConnectionResult.Failed:  
        doSomething();  
        break;  
    case ConnectionResult.Success:  
        doSomething();  
        break;  
}
```

I also wanted to use the enum for a comparison within my HTML using the *ngIf statement:

```
<div *ngIf="result == ConnectionResult.Success; else failed">  
      
</div>  
<ng-template #failed>  
      
</ng-template>
```

The code compiles but the browser gives me an error:

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

```

✖ ERROR TypeError: Cannot read property 'Success' of undefined
    at Object.View_ConnectResultComponent_0.co [as updateDirectives] (ConnectResultComponent.html:11)
    at Object.debugUpdateDirectives [as updateDirectives] (core.es5.js:12784)
    at checkAndUpdateView (core.es5.js:12122)
    at callViewAction (core.es5.js:12485)
    at execComponentViewsAction (core.es5.js:12417)
    at checkAndUpdateView (core.es5.js:12128)
    at callViewAction (core.es5.js:12485)
    at execEmbeddedViewsAction (core.es5.js:12443)
    at checkAndUpdateView (core.es5.js:12123)
    at callViewAction (core.es5.js:12485)
ConnectResultComponent.html:11

```

With the following html indication error line:

```

<div class="container">
  <div *ngIf="result == ConnectionResult.Success; else failed">✖
    
  </div>
  <ng-template #failed>
    
  </ng-template>
</div>

```

Does anyone know why the enum cannot be approached like this?

html



angular

typescript

enums

asked May 18 '17 at 10:30



Klyner

881 2 10 28

3 Answers



71

The scope of the template is limited to the component instance members. If you want to refer to something it needs to be available there

```

class MyComponent {
  ...
}

```

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



then you can use

```
*ngIf="connectionResult.Success"
```

See also [Angular2 access global variables from HTML template](#)

edited Feb 11 at 19:23

answered May 18 '17 at 10:33



Günter Zöchbauer

365k 83 1167 1056

Since I am strictly following the coding standards, what data type , I have to give for the connectionResult – [Nasrul Bharathi](#) Jun 28 '18 at 10:59

I don't know. I only used TypeScript in Plunker and there were no type checks. I'd expect the error message tells you the expected type when you use an incompatible one, doesn't it? – [Günter Zöchbauer](#) Jun 28 '18 at 11:08

Thanks, let me start a new conversation in stack overflow – [Nasrul Bharathi](#) Jun 28 '18 at 11:10

1 The main approach doesn't work for me, but the one presented on the link does... – [Juan](#) Jul 30 '18 at 16:54

1 Yes, just a plain property member didn't work for me, but setting it as a getter worked. – [Kon](#) Feb 11 at 19:08

You will have to write it in the following way in .ts file.

18

```
enum Tenure { day, week, all }
```

```
export class AppComponent {
  tenure = Tenure.day
  TenureType = Tenure
}
```

And now in html you can use this like

```
*ngIf = "tenure == TenureType.day ? selectedStyle : unSelectedStyle"
```

I hope it is more clear now. :)

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



You can just add the enum to your component as property **and use the same name** of the enum (Quarters) in your templates:

4

```
enum Quarters{ Q1, Q2, Q3, Q4}
```



```
export class AppComponent {  
  quarter = Quarters.Q1  
  Quarters = Quarters  
}
```

In your template

```
<div *ngIf="quarter == Quarters.Q1">I=am only visible for Q1</div>
```

The reason why this works is that the new property is basically of this type:

```
TileType: typeof TileType
```

answered May 17 at 7:57



[Ondrej Peterka](#)

2,159 3 27 45