

tslint / codelyzer / ng lint error: "for (... in ...) statements must be filtered with an if statement"

Asked 2 years, 11 months ago Active 10 months ago Viewed 93k times



Lint error message:

193

src/app/detail/edit/edit.component.ts[111, 5]: for (... in ...) statements must be filtered with an if statement



Code snippet (It is a working code. It is also available at [angular.io form validation section](https://angular.io/docs/api/forms/FORMS#validation)):



26

```
for (const field in this.formErrors) {  
  // clear previous error message (if any)  
  this.formErrors[field] = '';  
  const control = form.get(field);  
  
  if (control && control.dirty && !control.valid) {  
    const messages = this.validationMessages[field];  
    for (const key in control.errors) {  
      this.formErrors[field] += messages[key] + ' ';  
    }  
  }  
}
```

Any idea how to fix this lint error?

 angular angular2-forms angular-cli tslint

edited Nov 24 '16 at 0:59

asked Nov 23 '16 at 17:08



[choopage - Jek Bao](#)

2,114 4 20 51

4 Answers



A neater way of applying @Helzgate's reply is possibly to replace your 'for .. in' with

242



```
for (const field of Object.keys(this.formErrors)) {
```

answered Mar 29 '17 at 2:49



Maxxx

2,916

1

16

18

-
- 5 This should be the accepted answer as not only it solves the problem, it also reduces the amount of boilerplate code compared to additional conditionals such as `if (this.formErrors.hasOwnProperty(field))` . – Denialos Jul 28 '17 at 8:04
-
- 1 Be careful with the answer, it might break your codes. Test after you "fix" it. – ZZZ May 19 '18 at 7:22
-
- 3 This doesn't actually remove the tslint error for me. – HammerN'Songs Nov 16 '18 at 15:27
-
- 4 @HammerN'Songs check that you changed to `for of` instead of `for in` – Tom Nov 20 '18 at 10:35
-
- same problem here. error is not removed after using this – llamerr Nov 27 '18 at 20:44
-



To **explain the actual problem** that tslint is pointing out, a quote from the JavaScript documentation of the [for...in statement](#):

201



The loop will iterate over all enumerable properties of the object itself and those the object inherits from its constructor's prototype (properties closer to the object in the prototype chain override prototypes' properties).

So, basically this means you'll get properties you might not expect to get (from the object's prototype chain).

To solve this we need to iterate only over the objects own properties. We can do this in two different ways (as suggested by @Maxxx and @Qwertiy).

First solution

```
for (const field of Object.keys(this.formErrors)) {  
    ...  
}
```

Here we utilize the [Object.Keys\(\)](#) method which returns an array of a given object's own enumerable properties, in the same order as that provided by a `for...in` loop (the difference being that a `for-in` loop enumerates properties in the prototype chain as well).

Second solution

```
for (var field in this.formErrors) {
  if (this.formErrors.hasOwnProperty(field)) {
    ...
  }
}
```

In this solution we iterate all of the object's properties including those in it's prototype chain but use the [Object.prototype.hasOwnProperty\(\)](#) method, which returns a boolean indicating whether the object has the specified property as own (not inherited) property, to filter the inherited properties out.

edited Sep 6 '17 at 19:38

answered Aug 30 '17 at 11:59



akrabi

2,447 2 12 15

2 I'd like to notice that `Object.keys` is ES5. The only thing from ES6 there is for-of loop. We can iterate array in usual loop from 0 to its length and it would be ES5. – [Qwertiy](#) Sep 6 '17 at 16:26

4 once more notice: if somehow `this.formErrors` is null, `for...in` just do nothing, while `for ... of Object.keys()` would throw error. – [user3448806](#) Oct 12 '17 at 6:56

i am following the second solution but still i see the lint message. Disabled lint for time being. – [raj240](#) Feb 2 at 7:37

1 Why don't you recommend `Object.keys(obj).forEach(key => {...})` ? – [Ben Carp](#) Feb 24 at 15:06



69



```
for (const field in this.formErrors) {
  if (this.formErrors.hasOwnProperty(field)) {
```

```
for (const key in control.errors) {
  if (control.errors.hasOwnProperty(key)) {
```

answered Nov 24 '16 at 14:53



Qwertiy

8,574 6 27 72

▲ use Object.keys:

13

▼

```
Object.keys(this.formErrors).map(key => {
  this.formErrors[key] = '';
  const control = form.get(key);

  if(control && control.dirty && !control.valid) {
    const messages = this.validationMessages[key];
    Object.keys(control.errors).map(key2 => {
      this.formErrors[key] += messages[key2] + ' ';
    });
  }
});
```

answered Jan 15 '17 at 6:33



[Helzgate](#)

6,207

2

31

38
