! operator in typescript after object method

Asked 3 years, 2 months ago Active 1 year, 4 months ago Viewed 18k times

38

9

I have an object `X` with a method `getY()` returning an object `Y` with a method `a()`, in typescript. What does it mean an expression like this one:

```
X.getY()!.a()
```

I guess the `!` operator is used to check against null, but how does it work concretely? Where is defined in the language?

object typescript operators

edited Jul 13 '17 at 18:27
msanford
7,565 7 47 69

asked Aug 10 '16 at 13:32
user2032922
193 1 2 6

1 Answer

65

✓

It's called the "Non-null assertion operator" and it tells the compiler that `x.getY()` is not null.

It's a new typescript 2.0 feature and you can read about it in the what's new page, here's what it says:

> A new ! post-fix expression operator may be used to assert that its operand is non-null and non-undefined in contexts where the type checker is unable to conclude that fact. Specifically, the operation x! produces a value of the type of x with null and undefined excluded. Similar to type assertions of the forms x and x as T, the ! non-null assertion operator is simply removed in the emitted JavaScript code.

```
// Compiled with --strictNullChecks
function validateEntity(e?: Entity) {
```

```
function processEntity(e?: Entity) {
    validateEntity(e);
    let s = e!.name;  // Assert that e is non-null and access name
}
```

Edit

There's an issue for documenting this feature: Document non-null assertion operator (!)

edited Jun 9 '18 at 13:53
Eduard
2,772 1 14 33

answered Aug 10 '16 at 13:42
Nitzan Tomer
84k 25 204 212

1 wow. ive probably written `e && e.name` a couple thousand times. thanks – Tope Feb 5 at 4:31

15 @Tope do not confuse this with a 'null safety operator' of other languages (a?.b?.c?.d?). This simply tells typescript compiler that a variable is not null, which may be false and it will crash at runtime – Daniel San Feb 5 at 18:07