TypeScript or JavaScript type casting

Asked 6 years, 10 months ago Active 1 year, 9 months ago Viewed 171k times

145

8

How does one handle type casting in TypeScript or Javascript?

Say I have the following TypeScript code:

```
module Symbology {

    export class SymbolFactory {

        createStyle( symbolInfo : SymbolInfo) : any {
            if (symbolInfo == null)
            {
                 return null;
            }

            if (symbolInfo.symbolShapeType === "marker") {

                // how to cast to MarkerSymbolInfo
                return this.createMarkerStyle((MarkerSymbolInfo) symbolInfo);
            }
        }

        createMarkerStyle(markerSymbol : MarkerSymbolInfo ): any {
            throw "createMarkerStyle not implemented";
        }

    }
}
```

where `SymbolInfo` is a base class. How do I handle typecasting from `SymbolInfo` to `MarkerSymbolInfo` in TypeScript or Javascript?

casting typescript

edited Jan 28 '14 at 10:56 asked Nov 3 '12 at 0:17

2 Answers

232 ✓

You can cast like this:

```
return this.createMarkerStyle(<MarkerSymbolInfo> symbolInfo);
```

Or like this if you want to be compatible with tsx mode:

```
return this.createMarkerStyle(symbolInfo as MarkerSymbolInfo);
```

Just remember that this is a compile-time cast, and not a runtime cast.

edited Nov 10 '17 at 16:44
Andy Skirrow
2,883 11 36

answered Nov 3 '12 at 5:53
blorkfish
12.3k 4 27 19

8 Now, I see that in in doc, referred to as Type Assertions in section 4.13. – Klaus Nji Nov 3 '12 at 12:27

2 It doesn't work in JSX mode. Very sad. – Brian Haak Jul 7 '16 at 6:06

This answer does no longer provide the full picture of type assertion in typescript, whereas Alex's answer gives a more complete picture, and should be the accepted answer. – Kristoffer Dorph Jan 5 '17 at 9:20

@KristofferDorph This answer is 4 years old. At the time of writing TypeScript was at version 0.8.1, and thus was the correct answer at the time. JSX support was only included 3 years later. – blorkfish Jan 5 '17 at 11:43

@blorkfish that is true, but it is good practice to follow the times, so people asking the same question today gets the current answer, and not as it where 4 years ago :-) – Kristoffer Dorph Jan 5 '17 at 14:04

130

This is called type assertion in TypeScript, and since TypeScript 1.6, there are two ways to express this:

```
// Original syntax
var markerSymbolInfo = <MarkerSymbolInfo> symbolInfo;
```

Both alternatives are **functionally identical**. The reason for introducing the `as` -syntax is that the original syntax conflicted with JSX, see the design discussion here.

If you are in a position to choose, just use the syntax that you feel more comfortable with. I personally prefer the `as` -syntax as it feels more fluent to read and write.

edited Jun 15 '17 at 18:21

answered Feb 12 '16 at 12:10

Alex
9,044 6 42 66

1 How do you indicate to typescript that you have converted an object to another type? For example a func that returns type2, inside it it http gets type 1, does logic to convert, and returns what was type1 but is now type2? – Tony Gutierrez Jul 5 '18 at 5:56

@TonyGutierrez How do you do the conversion? – Alex Jul 5 '18 at 7:49

Basically take one property and modify. The only way I have found to do this is to create a new var (type2) and copy in the props from the type1var and then return it. You can't modify the type1 and return, or you get a "Can't cast" error. – Tony Gutierrez Jul 5 '18 at 12:50