How to instantiate an object in TypeScript by specifying each property and its value?

Asked 2 years, 5 months ago Active 2 years, 1 month ago Viewed 15k times

9

4

Here's a snippet in which I instantiate a new `content` object in my service:

```
const newContent = new Content(
     result.obj.name
     result.obj.user.firstName,
     result.obj._id,
     result.obj.user._id,
);
```

The problem is that this way of object instantiation relies on the order of properties in my `content` model. I was wondering if there's a way to do it by mapping every property to the value I want to set it to, for example:

```
const newContent = new Content(
    name: result.obj.name,
    user: result.obj.user.
    content_id: result.obj._id,
    user_id: result.obj.user._id,
);
```

typescript syntax instance instantiation

asked Apr 10 '17 at 4:04

YSA

175 1 2 20

2 Answers

16

```
    name: result.obj.name,
    user: result.obj.user.
    content_id: result.obj._id,
    user_id: result.obj.user._id,
});
```

Here you can instantiate an object and use type assertion or casting to the Content type. For more information on type assertion: https://www.typescriptlang.org/docs/handbook/basic-types.html#type-assertions

edited Jul 28 '17 at 2:21

answered Jul 27 '17 at 2:07

Geoff Lentsch
694 9 13

It is clear what you have changed, but it would be useful to say why. Teaching moments everywhere. – Anthony Horne Jul 27 '17 at 6:35

6 This is actually very very wrong! You're not instantiating an instance of `Content` here, you are creating a simple object which has the same members. If for example `Content` would have a method `fn`, then in your example this will fail: `newContent.fn()`. – Nitzan Tomer Sep 2 '18 at 14:07

1 this answer will work if you extend the prototype of the object like so: `const newContent = <Content>(_.extend({...}, Content.prototype));` – logeyg Oct 11 '18 at 20:11

7

You can pass an object to the constructor which wraps all of those variables:

```
type ContentData = {
    name: string;
    user: string;
    content_id: string;
    user_id: string;
}

class Content {
    constructor(data: ContentData) {
        ...
    }
}
```

And then:

```
        user: result.obj.user.
        content_id: result.obj._id,
        user_id: result.obj.user._id,
    });
```

answered Apr 10 '17 at 4:07

Nitzan Tomer

82.3k 25 197 210