js 'types' can only be used in a .ts file - Visual Studio Code using @ts-check

Asked 1 year, 5 months ago Active 8 months ago Viewed 33k times

60

11

I am starting to use TypeScript in a Node project I am working on in Visual Studio Code. I wanted to follow the "opt-in" strategy, similar to Flow. Therefore I put `// @ts-check` at the top of my `.js` file in hope to enable TS for that file. Ultimately I want the same experience of "linting" as Flow, therefore I installed the plugin TSLint so I could see Intellisense warnings/errors.

But with my file looking like:

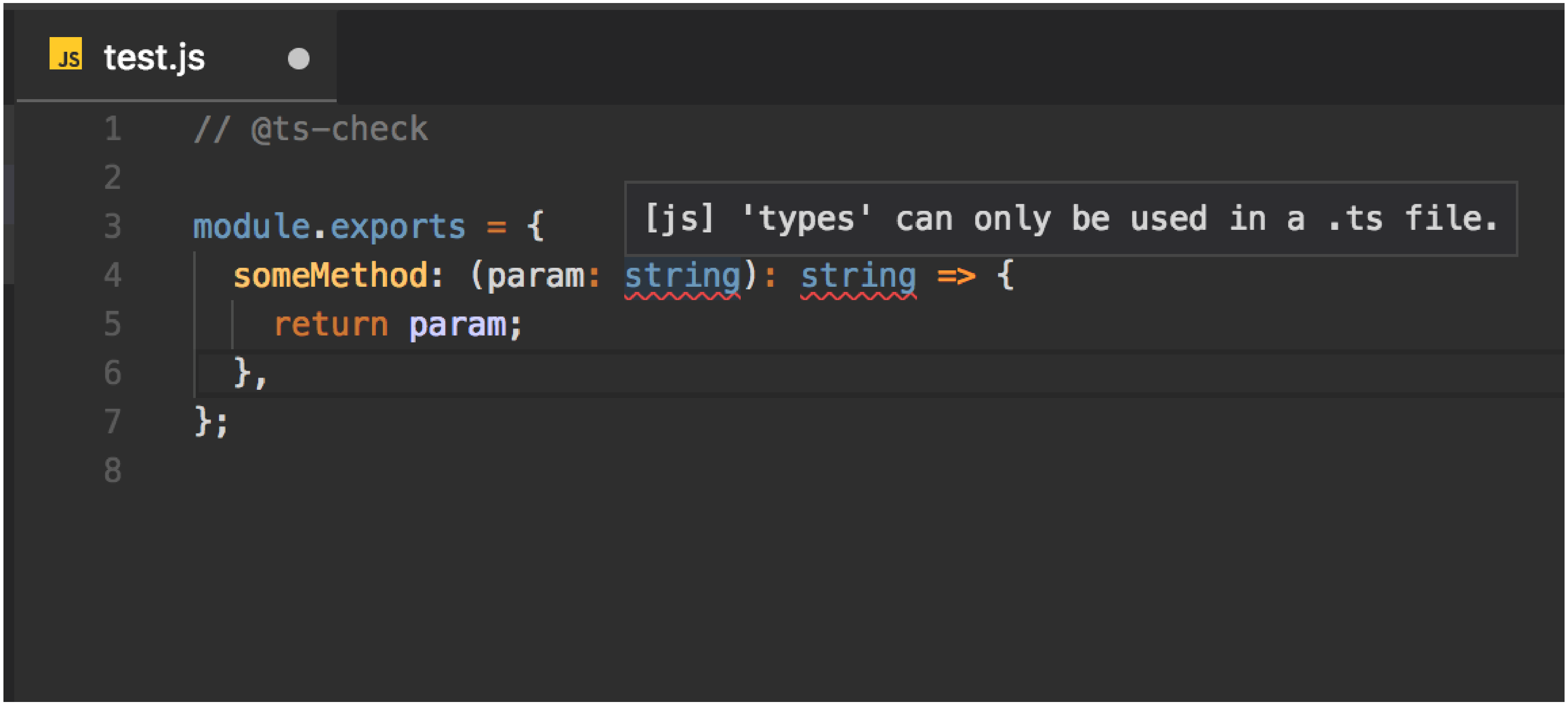
```
// @ts-check

module.exports = {
  someMethod: (param: string): string => {
    return param;
  },
};
```

and my `tsconfig.json` file looking like...

```
{
  "compilerOptions": {
      "target": "es2016",
      "module": "commonjs",
      "allowJs": true
  }
}
```

I get this error: `[js] 'types' can only be used in a .ts file.` as shown below in the image.



```
// @ts-check

module.exports = {
  someMethod: (param: string): string => {
    return param;
  },
};
```

[js] 'types' can only be used in a .ts file.

I saw this question which recommended disabling javascript validation in vscode but then that doesn't show me **any** TypeScript Intellisense info.

I tried setting `tslint.jsEnable` to `true` in my vscode settings as mentioned in the TSLint extension docs but no luck there.

What is the correct setup in order to use `.js` files with TypeScript and get Intellisense so I know what the errors in my code are before I run any TS commands?

javascript typescript visual-studio-code

asked Feb 19 '18 at 3:28

james
1,198 3 21 40

Do you have any error if you change the extension to ts? – israel.zinc Feb 19 '18 at 3:34

@israel.zinc changing the extension to `.ts` show me the TS errors/warnings as expected. I guess this could work, but I was hoping for more of the opt-in method using `@ts-check` while keeping all my extensions as `.js` – james Feb 19 '18 at 3:49

Possible duplicate of [Visual Studio Code:[js] types can only be used in a .ts file] – BuZZ-dEE Jun 29 at 23:36

@BuZZ-dEE it might be, but the answers there did not solve the issue. I mentioned this in my original question. – james Jul 2 at 14:17

3 Answers

123

✓

I'm using [@flow] with vscode but had the same problem.

> I solved it with this steps:

1. install the extension [Flow Language Support]
2. disable the built-in TypeScript extension:

> How to disable built-in TypeScript:

1. go to *extensions* tab
2. search for **@builtin TypeScript and JavaScript Language Features**
3. click on *Disable*

answered May 17 '18 at 6:59

Idan Dagan
2,719 2 16 28

4 The accepted answer is not always correct. There's a bug in VSCode with React Native and Flow that gives the same message in .js files, and in that case YOU SHOULD NOT change to .ts, but continue with .js and fix the problem as suggested here!! – pashute May 17 '18 at 13:24

43 Just an FYI: Disabling this extension basically nullifies all of the nice things about JS development with VSCode. Instead, you should just add the following into your settings json file: `"javascript.validate.enable": false` – heez May 22 '18 at 21:52

as @heez has mentioned this will affect, finding reference to javascript files, autocompletion etc don't disable Javascript Language Features. – Ajitsen Dec 29 '18 at 12:35

This disables things like going to function definitions. – GEMI Jan 16 at 20:13

The valid answer was said by @heez – Asim Olmez Jul 20 at 14:41

44

Use `"javascript.validate.enable": false` in your VS Code settings, It doesn't disable ESLINT. I use both ESLINT & Flow. Simply follow the instructions [Flow For Vs Code Setup](https://github.com/flowtype/flow-for-vscode#setup)

Adding this line in settings.json. Helps `"javascript.validate.enable": false`

edited Oct 23 '18 at 9:10
chengsam
4,841 3 20 32

answered Jun 6 '18 at 17:42
Adeel Imran
3,983 1 25 47

3 With this You don't have to disable typescript. Both can exists. Best solution! – Nirus Oct 18 '18 at 5:48

1 If you are using ESLint to validate your Javascript, then this answer is the best solution. More info: code.visualstudio.com/docs/languages/… and also github.com/flowtype/flow-for-vscode#setup – Beau Smith Dec 6 '18 at 18:59

0

You must use a `.ts` file - e.g. `test.ts` to get Typescript ~~validation, intellisense~~ `typing` of vars, return types, as well as "typed" error checking (e.g. passing a `string` to a method that expects an `number` param will error out).

It will be transpiled into (standard) `.js` via `tsc`.

Update (11/2018):

Clarification needed based on down-votes, very helpful comments and other answers.

`types`

- Yes, you can do `type` checking in VS Code in `.js` files with `@ts-check` - as shown in the animation
- What I originally was referring to for **Typescript** `types` is something like this in `.ts` which isn't quite the same thing:

  `hello-world.ts`

  ```
  function hello(str: string): string {
    return 1;
  }
  ```

```
function foo(str:string):void{
   console.log(str);
}
```

This will not compile. `Error: Type "1" is not assignable to String`

- if you tried this syntax in a **Javascript** `hello-world.js` file:

```
//@ts-check

function hello(str: string): string {
  return 1;
}

function foo(str:string):void{
   console.log(str);
}
```

The error message referenced by OP is shown: `[js] 'types' can only be used in a .ts file`

If there's something I missed that covers this as well as the OP's context, please add. Let's all learn.

edited Nov 17 '18 at 17:34

answered Feb 19 '18 at 3:34
EdSF
8,576 3 31 63

I have found a few articles that suggest you can use all the features of TS within a JS file. Here's one from Smashing Magazine, a GitHub issue for vscode that describes the setup, and some release notes for vscode – james Feb 19 '18 at 3:53

@jamez14 I haven't tried/had the need to *not* use `.ts`. IINM, the samples show *error checking* and *intellisense*. The issue you have is (static) `type` ing your variables, and return `types` as well - e.g. `foo:string`, `bar:number`, `fubar:MyClass`, `myMethod(f:Array):void`, `anotherMethod():string`. Hth... – EdSF Feb 19 '18 at 5:31 ✎

yep, looks like you are right with your assessment. After looking more at the examples, I'm seeing it is just JS error checking, not TS, when using the `@ts-check` attribute. Very misleading... – james Feb 19 '18 at 13:39