

How to debug TypeScript with VS Code



Philipp Kief

Mar 30, 2018 · 4 min read

Even if you are developing only a small application or have little experience with TypeScript altogether, debugging with VS Code will be inevitable. Not to mention larger projects. In this article I'll show you how to setup your development environment with VS Code to debug your code.



Create new project

At first we create a small TypeScript project. Please make sure that you have VS Code and Node.js pre-installed on your computer. You can also skip the next steps and simply clone this Github repository.

Create source files

Create a new project folder called “vscode-typescript-debugging”. After that create a “src” folder and add an “app.ts” file with the following content:

```
1  import { hello } from './hello';  
2  
3  class App {
```

```
4    /** Entry point of our app */
5    public static start() {
6        console.log(hello('world'));
7    }
8 }
9
10 App.start();
```

app.ts hosted with ❤ by GitHub

[view raw](#)

Add another file called “hello.ts” to the “src” folder:

```
1  /** Say hello */
2  export const hello = (name: string) => {
3      const greeting = `Hello ${name}!`;
4      return greeting;
5  };
```

hello.ts hosted with ❤ by GitHub

[view raw](#)

That’s the whole code! This is sufficient for this tutorial. We don’t make things more complicated than absolutely necessary.

TypeScript compiler

After that we should compile the TypeScript code into plain JavaScript. Create a “tsconfig.json” file in the project folder and add the following content:

```
1  {
2      "compilerOptions": {
3          "outDir": "./out",
4          "rootDir": "./src",
5          "sourceMap": true,
6          "moduleResolution": "node",
7          "target": "es5"
8      }
9  }
```

tsconfig.json hosted with ❤ by GitHub

[view raw](#)

These are very basic options for the TypeScript compiler. If you need more information about the compiler options, then take a short look into the official handbook. It is important to set the `sourceMap` -property to **true**. Sourcemap files are required to map the TypeScript code to the JavaScript code in the debugger later.

Note: If you have installed TypeScript globally with the command `npm i -g typescript` you can call the compiler in your terminal by just typing `tsc`. It compiles the TypeScript code according to the options in “tsconfig.json” and outputs the compiled JS-files into the “out” folder.

NPM-Scripts

What we also need is the well-known “package.json” file. Simply run the following commands in your terminal to create it and to add the required dependencies:

- `npm init --yes`
- `npm install typescript --save-dev`

Next we add the required scripts to the “package.json” file, which finally looks similar to this:

```
1  {
2    "name": "vscode-typescript-debugging",
3    "version": "1.0.0",
4    "devDependencies": {
5      "typescript": "^2.7.2"
6    },
7    "scripts": {
8      "start": "node out/app.js",
9      "prestart": "npm run build",
10     "build": "tsc"
11   }
12 }
```

package.json hosted with ❤ by GitHub

[view raw](#)

Short explanation of the scripts:

start — run the compiled app with node

prestart — is called automatically before the start script

build — runs the TypeScript compiler

Run the app

Open your terminal and run the following command:

```
npm start
```

Eventually you should see the “Hello world!” in your terminal and the compiled JavaScript-files in the “out” folder. Fine! But what about the debugging? One step at a time!

Debugging

Create a “.vscode” folder in the project directory and add a file called “launch.json” with the following configurations:

```
1  {
2      "version": "0.2.0",
3      "configurations": [
4          {
5              "type": "node",
6              "request": "launch",
7              "name": "Build Project",
8              "program": "${workspaceFolder}\\src\\app.ts",
9              "preLaunchTask": "npm: build",
10             "sourceMaps": true,
11             "smartStep": true,
12             "internalConsoleOptions": "openOnSessionStart",
13             "outFiles": [
14                 "${workspaceFolder}/out/**/*.js"
15             ]
16         }
17     ]
18 }
```

launch.json hosted with ❤ by GitHub

[view raw](#)

Short explanation of the most *relevant* configs:

program — entry file of our app

preLaunchTask — calls the “build” script of package.json

sourceMaps — use the sourcemaps from the “out” folder

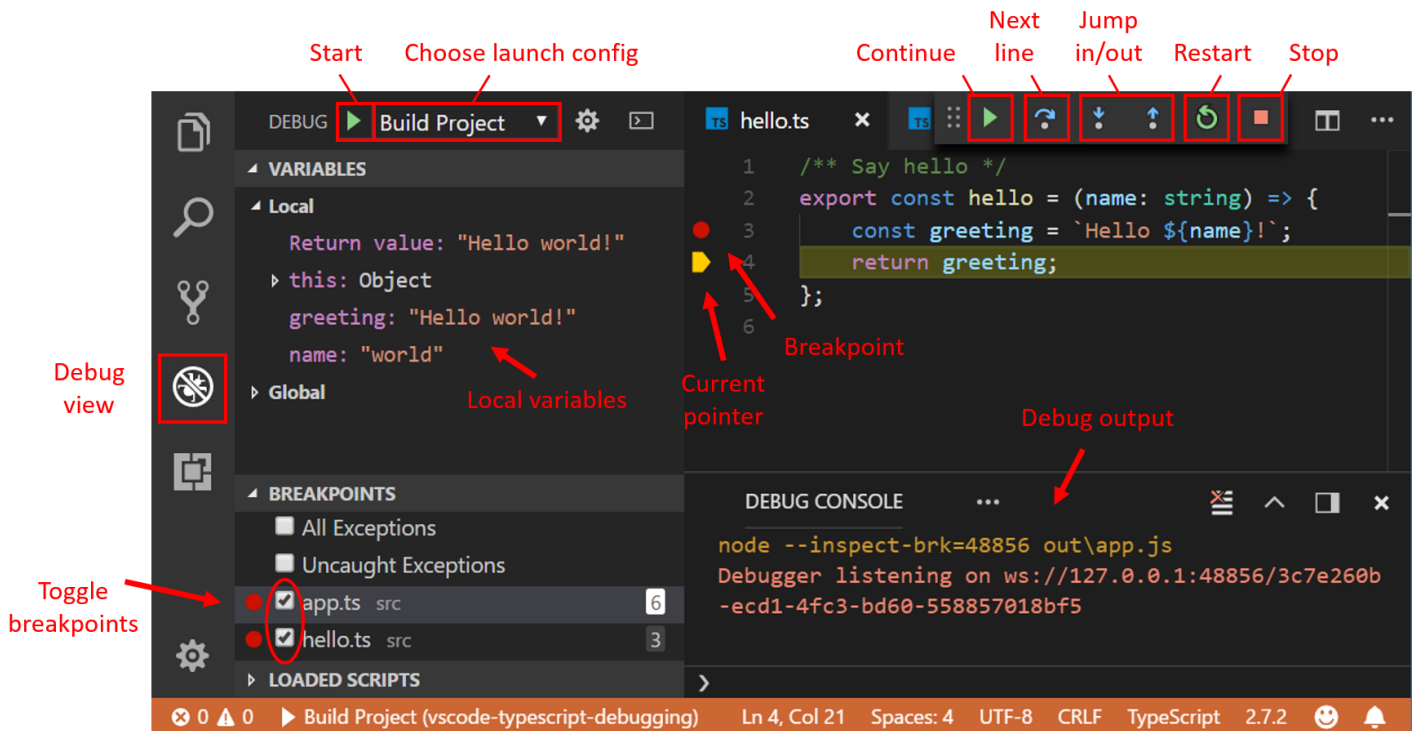
smartStep — skip “uninteresting” code in the debugger (e.g. compiled JS-files)

internalConsoleOptions — open the debug console during a debugging session

outFiles — place where the debugger looks for the sourceMap files

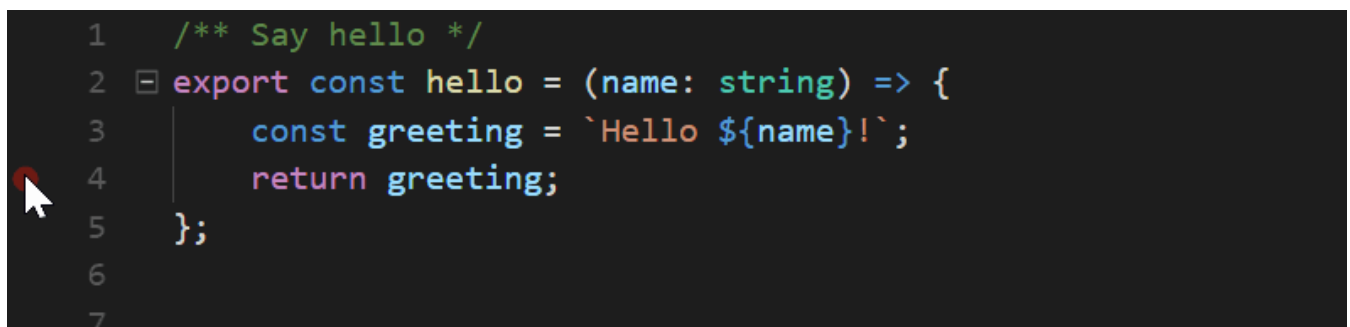
Now you can open the debug view (Ctrl-Shift-D) of VS Code. By clicking to the left of a line number you can add a new breakpoint. Press the green debug button with the selected launch configuration (“Build project”) or simply press the **F5**-key on your keyboard to start debugging.

Short overview:



Conditional breakpoints

Assuming you have a for-loop and do not want to break execution at each iteration, you can add conditional breakpoints. Here you can decide between an “Expression” or “Hit Count” condition.



This breakpoint only stops execution if "name" has the value "world".

Expression: If the expression is true, the breakpoint stops execution.

Hit Count: Number of hits until the breakpoint stops execution.

More information can be found in the VS Code documentation.

Conclusion

VS Code offers a very comfortable development environment for TypeScript, no matter how big the project itself is. Using the built-in debugger is much more useful and time-saving than using the well-known “console.log” for this purpose.

Hopefully you enjoyed this article. You can find the link to the related source code of the demo project below.

. . .

Related source code

PKief/vscode-typescript-debugging

vscode-typescript-debugging - How to debug TypeScript with VS Code

github.com

[Typescript](#) [Vscode](#) [Debugging](#) [Visual Studio Code](#)

[About](#) [Help](#) [Legal](#)