

Debugging Advantages of TypeScript

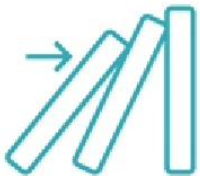


Type errors stopped at compile time



Additional tooltips, code hints prevent errors

One of TypeScript's main advantages of JS is easier debugging in many cases.



Common pitfalls (such as switch statements lacking a *break*), are disabled



Which Errors Cannot Be Prevented by TypeScript?

If TypeScript's built-in type-checking prevents most categories of error from ever occurring, what errors *can* still occur?



Incorrectly written functions and miscalculations



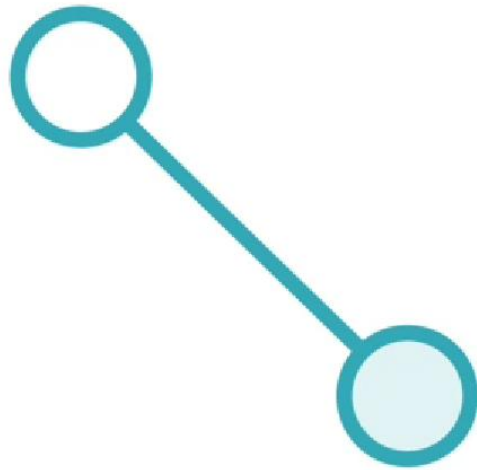
Errors arising from corner cases and user input



Unanticipated values from 3rd party APIs



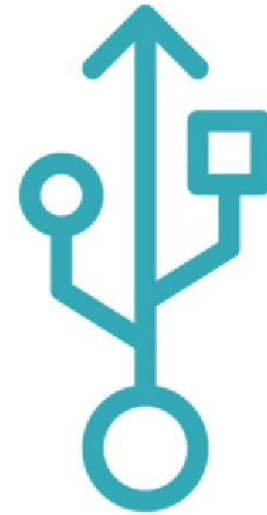
Source Maps



**Couples generated
code with source
code**



**Browser will show
source file, not
generated file, while
debugging**



**Can be embedded
entirely within
generated file**



Enabling Source Maps

tsconfig.json

```
{  
  compilerOptions : {  
    sourceMap : true  
  }  
}
```



Demo



Start on Git Branch: 2-browser
*[https://github.com/danielstern/
configuring-typescript/tree/
2-browser](https://github.com/danielstern/configuring-typescript/tree/2-browser)*

**Update tsconfig.json and webpack config
to output source maps**

- Examine generated sourcemap
- Investigate troubleshooting with Chrome using source maps



FileEditSelectionViewGoRunTerminalHelp

tsconfig.json - typescript-scaffold-demo-2 - Visual Studio Code

EXPLORER

TYPESCRIPT-SCAFFOLD-DEMO-2

> dist

> node_modules

> public

> src

> app

TS index.ts

.gitignore

{ } package-lock.json

{ } package.json

README.md

TS tsconfig.json M

webpack.config.js

OUTLINE

TIMELINE

tsconfig.json M

tsconfig.json > { } compilerOptions > sourceMap

```
1 {
2   "compilerOptions": {
3     "outDir": "./dist/js",
4     "sourceMap": true
5   },
6   "include": ["src/index.ts"],
7   "exclude": ["node_modules"]
8 }
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

PS C:\Projects\typescript-scaffold-demo-2>

node

powershell

2-browser* 0 0 Ln 4, Col 26 Spaces: 4 UTF-8 CRLF JSON with Comments

FileEditSelectionViewGoRunTerminalHelpwebpack.config.js - typescript-scaffold-demo-2 - Visual Studio Co...

EXPLORER

TYPESCRIPT-SCAFFOLD-DEMO-2

- > dist
- > node_modules
- > public
- > src
 - > app

TS index.ts

.gitignore

{ } package-lock.json

{ } package.json

i README.md

TS tsconfig.jsonM

webpack.config.jsM

OUTLINE

TIMELINE

tsconfig.json M

webpack.config.js M

webpack.config.js > [?] <unknown> > 🔑 devtool

```
17  },
18  output: {
19    filename: 'bundle.js',
20    path: path.resolve(__dirname, 'bin'),
21    sourceMapFilename: "bundle.js.map"
22  },
23  devtool: "eval-source-map",
24  devServer: {
25    contentBase: [path.join(__dirname, 'public'), path.join(__dirname, 'bin')],
26    port: 7777,
27  },
28  };
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

dditional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\Code Whisperer\AppData\Roaming\npm-cache_logs\2021-07-07T09_53_59_486Z-debug.log
PS C:\Projects\typescript-scaffold-demo-2> npm run dev

node

powershell

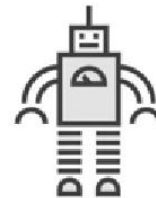
2-browser* 0 0 Ln 23, Col 7 Spaces: 2 UTF-8 CRLF JavaScript

Debugging TypeScript with VSCode and Chrome



TypeScript, VSCode, and Chrome

Chrome and VSCode can work together to create a sophisticated TypeScript debugging flow.



VSCode automatically opens and closes connected Chrome window



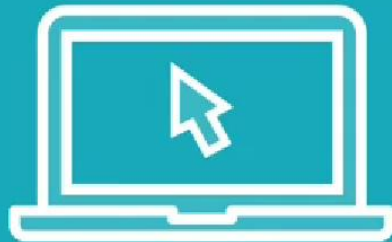
Pausing on a breakpoint brings up original breakpoint in VSCode



Extensions required, principle can be applied to most browsers and IDEs



Demo



Start on Git Branch: 3-model-view
`https://github.com/danielstern/
configuring-typescript/tree/
3-model-view`

Install VSCode debugging extension

**Use Chrome and VSCode to create a
debugging workflow**



Demo: Debugging TypeScript with VSCode and Chrome



FileEditSelectionViewGoRunTerminalHelp

typescript-scaffold-demo-2 - Visual Studio Code

EXTENSIONS: ...

Debugger for Chrome

★

Debugger for Chrome 4.12.12

Debug your JavaScript code in t...

Microsoft

Debugger for Java 0.34.0

A lightweight Java debugger fo...

Microsoft

Install

★

Debugger for Firefox 2.9.4

Debug your web application or ...

Firefox DevTools

Install

ElixirLS: Elixir suppor... 0.7.0

Elixir support with debugger, a...

ElixirLS

Install

Apex Interactive Deb... 52.3.0

Provides debugging support fo...

Salesforce

Install

Maven for Java 0.31.0

Manage Maven projects, execut...

Microsoft

Install

Project Manager for J... 0.18.5

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

powershell + ^ X

PS C:\Projects\typescript-scaffold-demo-2>

3-model-view-b 0 0

FileEditSelectionViewGoRunTerminalHelp

launch.json - typescript-scaffold-demo-2 - Visual Studio Code

Open chrome

launch.json

VARIABLES

WATCH

CALL STACK

.vscode > launch.json > Launch Targets > Open chrome debugger

```
1  {
2      "configurations": [{
3
4          "type": "chrome",
5          "request": "launch",
6          "name" : "Open chrome debugger",
7          "url" : "http://localhost:7777",
8          "webRoot": "${workspaceFolder}",
9          "sourceMaps": true,
10         "trace": "verbose"
11     }]
12 }
```

Add Configuration...

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

node

modules by path ./src/ 1.7 KiB

./src/index.ts 379 bytes [built] [code generated]

./src/app/Main.ts 955 bytes [built] [code generated]

./src/defaultState.ts 409 bytes [built] [code generated]

webpack 5.42.1 compiled successfully in 3306 ms

i [wdm]: Compiled successfully.

3-model-view-b* 0 0 Open chrome debugger (typescript-scaffold-demo-2) Spaces: 4 UTF-8 CRLF JSON with Comments

Demo: Using Breakpoints to Debug TypeScript



VS

FileEditSelectionViewGoRunTerminalHelp

index.ts - typescript-scaffold-demo-2 - Visual Studio Code

—□×

📁

🔍

🔗 4

🔧

📦

👤

⚙️

🟢 Start Debugging

Open chrome ▾

TS index.ts M ×

webpack.config.js M

launch.json

TS tsconfig.json M

🔄

📄

⋮

▽ VARIABLES

▽ WATCH

▽ CALL STACK

src > TS index.ts > [🔗] renderApp

1 import { Main } from './app/Main';

2 import { defaultState } from './defaultState';

3

4 const renderApp = ()=>{

5

6 const rendered : string = new Main(defaultState).render();

7 document.getElementById("App").innerHTML = rendered;

8

9 }

10

11

12 renderApp();

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

+ ▾ ^ ×

./src/index.ts 311 bytes [built] [code generated]

webpack 5.42.1 compiled successfully in 72 ms

i 「wdm」: Compiled successfully.

🏠

📄 node

📄 powershell

▶

🔗 debugging-configured*

🔄

⊗ 0 ⚠ 0

🔗 Open chrome debugger (typescript-scaffold-demo-2)

Spaces: 4

UTF-8

CRLF

TypeScript

4.3.2

🔗

🔔

VS Code interface showing a TypeScript project. The editor displays `index.ts` with the following code:

```
1 import { Main } from './app/Main';
2 import { defaultState } from './defaultStat
3
4 const renderApp = ()=>{
5
6     const rendered : string = new Main(defa
7     document.getElementById("App").innerHTM
8
9 }
10
11
12 renderApp();
```

The left sidebar shows the **VARIABLES** panel with a local variable `renderApp` and the **CALL STACK** panel showing the function call.

The bottom status bar indicates the project is **debugging-configured*** and the **Open chrome debugger (typescript-scaffold-** is active.

localhost:7777

Name	Remaining
The Chatty Chellist	50
The Zany Xylophone	50
The Humble Double Bass	30