TypeScript Debugging

Daniel Stern
CODE WHISPERER

@danieljackstern

Advantages of TypeScript Debugging

TypeScript offers solid protection from compile-time errors. The Debugger allows developers to fix run-time errors as well.

```
const triple
  = a=>a*2
```

Find runtime errors

Some errors cannot be found at compile time

Use breakpoints

Breakpoints are an advanced debugging tool

Save time

Good debugging practices saves many hours with a large team

Debugging Node with VSCode

Debugging Node with VSCode

The Debug Panel (`Ctrl+Shift+D`) provides low-level control over applications

Debugger can attach to special Node processes

`launch.json` can launch application and debugger with one click

Variables can be observed and changed in real time

Demo

Create launch.json

- Add script to debug using Node
- Script to debug using Chrome will be added in upcoming clip

Launch server and attach debugger to it

- Note the effects of breakpoints on debugging process

Configuring ESLint

“Amazing, beauteous change!
A world created new!
My thoughts with transport range
The lovely scene to view;”

Philip Doddridge

Why Do We Need Linting in Addition to Type Checking?

Compile-time error checking cannot catch certain mistakes

Linting catches stylistic errors that type checking does not

Developers on a team will write code in a similar manner

Well-linted project is much easier (and cheaper) to maintain

Demo

Install ESLint

- Download from NPM
- Create a configuration file
- Integrate with TypeScript

Use ESLint to correct code style

Client-Side TypeScript
Debugging with Chrome Extensions

Server or Client Debugging?

Server-side applications are run and debugged with Node

Client applications are run by the browser of the client’s choice and debugged with Chrome

Demo

Install Chrome Debugger for VSCode

Update `launch.json` to include Chrome debugging script

Execute debugging procedure via Debug panel and note the output in Chrome

Completing the Application

Demo

Display existing items as HTML list

Add functionality for new questions

- Add server API to add questions
- Add client-side form to work with API

Finalize NPM scripts

TypeScript Debugging Summary

TypeScript Debugging is an essential task

- Replaces console log statements and guesswork
- Ensures functioning of app’s core features

Debugger is launched from Debug Panel in VSCode

- Can be configured for client or server
- Works with breakpoints in TypeScript

ESLint is used to standardize code style

- Prevents certain classes of error