Debugging with Visual Studio Code

Kamran Ayub

Helping developers get started with JavaScript

www.linkedin.com/in/kamranayub

Debugging in JavaScript Applications

Browser

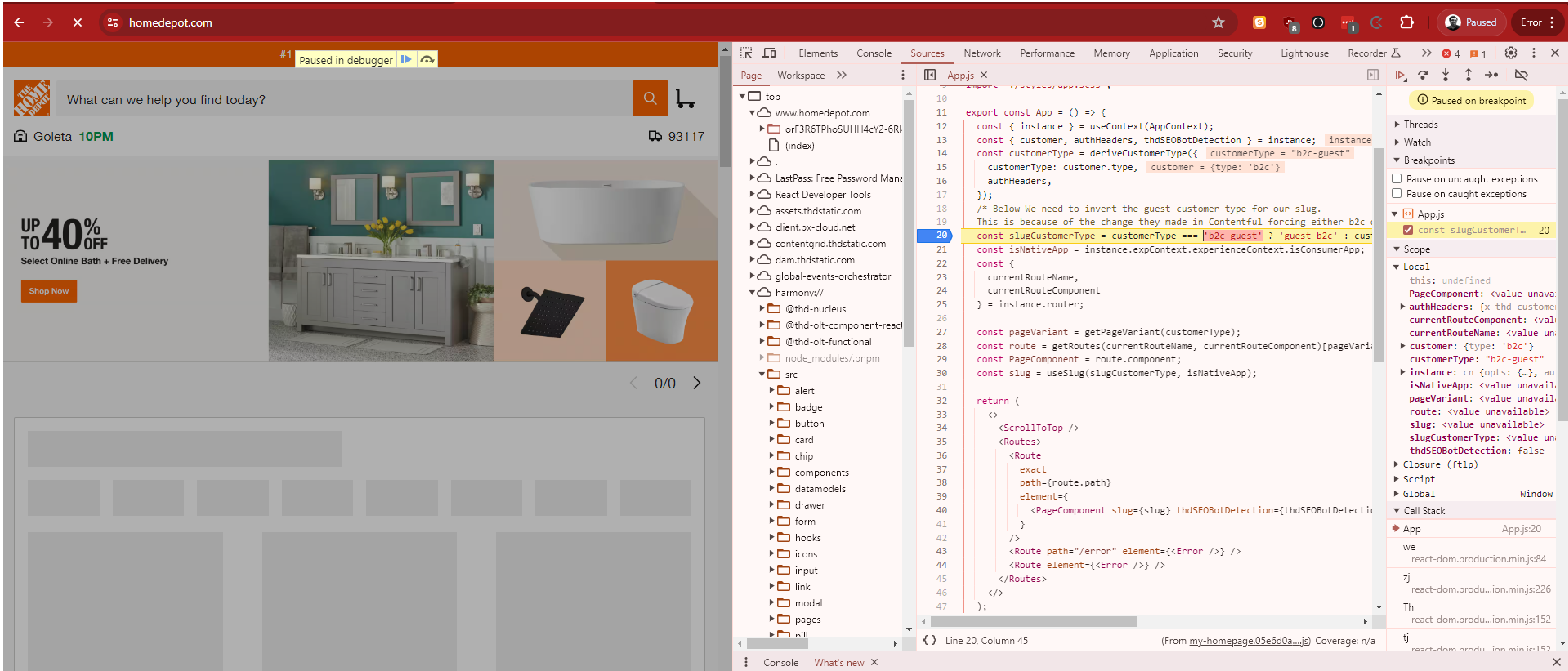
Client-side debugging

Node.js

Server-side debugging

```
console.log(someVar);
```

Demo: Browser-based Debugging



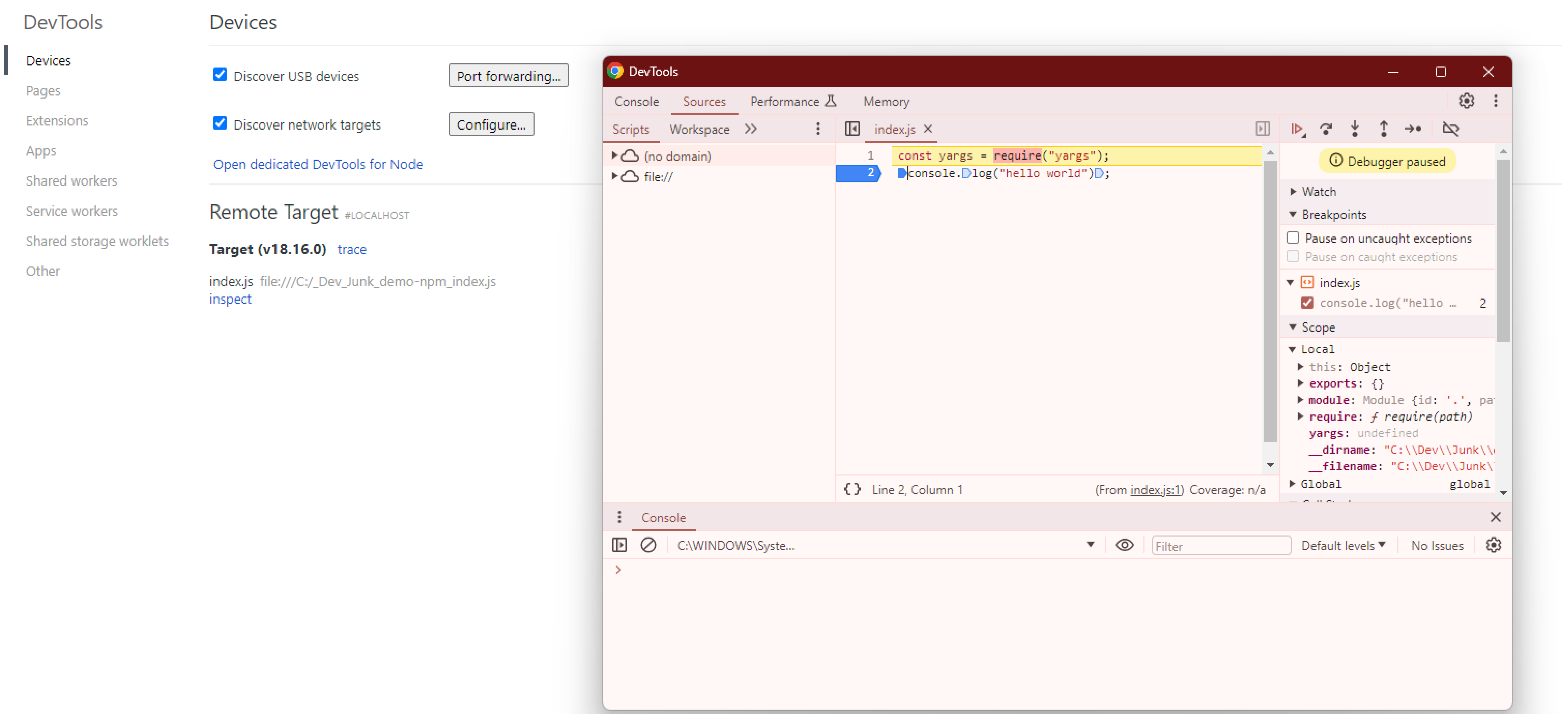
```
top
  www.homedepot.com
    orF3R6TPhoSUHH4cY2-6Rl
    (index)
  .
  LastPass: Free Password Mana
  React Developer Tools
  assets.thdstatic.com
  client.px-cloud.net
  contentgrid.thdstatic.com
  dam.thdstatic.com
  global-events-orchestrator
  harmony://
    @thd-nucleus
    @thd-olt-component-react
    @thd-olt-functional
    node_modules/.pnpm
    src
      alert
      badge
      button
      card
      chip
      components
      datamodels
      drawer
      form
      hooks
      icons
      input
      link
      modal
      pages
```

```
export const App = () => {
  const { instance } = useContext(AppContext);
  const { customer, authHeaders, thdSEOBotDetection } = instance;  instance
  const customerType = deriveCustomerType({  customerType = "b2c-guest"
    customerType: customer.type,  customer = {type: 'b2c'}
    authHeaders,
  });
  /* Below We need to invert the guest customer type for our slug.
  This is because of the change they made in Contentful forcing either b2c
  const slugCustomerType = customerType === 'b2c-guest' ? 'guest-b2c' : cus
  const isNativeApp = instance.expContext.experienceContext.isConsumerApp;
  const {
    currentRouteName,
    currentRouteComponent
  } = instance.router;

  const pageVariant = getPageVariant(customerType);
  const route = getRoutes(currentRouteName, currentRouteComponent)[pageVari
  const PageComponent = route.component;
  const slug = useSlug(slugCustomerType, isNativeApp);

  return (
    <>
      <ScrollToTop />
      <Routes>
        <Route
          exact
          path={route.path}
          element={
            <PageComponent slug={slug} thdSEOBotDetection={thdSEOBotDetecti
          }
        />
        <Route path="/error" element={<Error />} />
        <Route element={<Error />} />
      </Routes>
    </>
  );
```

Line 20, Column 45 (From my-homepage.05e6d0a....js) Coverage: n/a

```
Paused on breakpoint
Threads
Watch
Breakpoints
  Pause on uncaught exceptions
  Pause on caught exceptions
App.js
  ✓ const slugCustomerT…   20
Scope
  Local
    this: undefined
    PageComponent: <value unava
    authHeaders: {x-thd-custome
    currentRouteComponent: <val
    currentRouteName: <value un
    customer: {type: 'b2c'}
    customerType: "b2c-guest"
    instance: cn {opts: {…}, au
    isNativeApp: <value unavail
    pageVariant: <value unavail
    route: <value unavailable>
    slug: <value unavailable>
    slugCustomerType: <value un
    thdSEOBotDetection: false
  Closure (ftlp)
  Script
  Global                Window
Call Stack
  App                   App.js:20
  we
    react-dom.production.min.js:84
  zj
    react-dom.produ...ion.min.js:226
  Th
    react-dom.produ...ion.min.js:152
  tj
```

Demo: Node.js-based Debugging



Devices

☑ Discover USB devices

☑ Discover network targets

Open dedicated DevTools for Node

Remote Target #LOCALHOST

Target (v18.16.0) trace

index.js file:///C:/_Dev_Junk_demo-npm_index.js
inspect

(no domain)
file://

```
const yargs = require("yargs");
console.log("hello world");
```

Line 2, Column 1 (From index.js:1) Coverage: n/a

Debugger paused

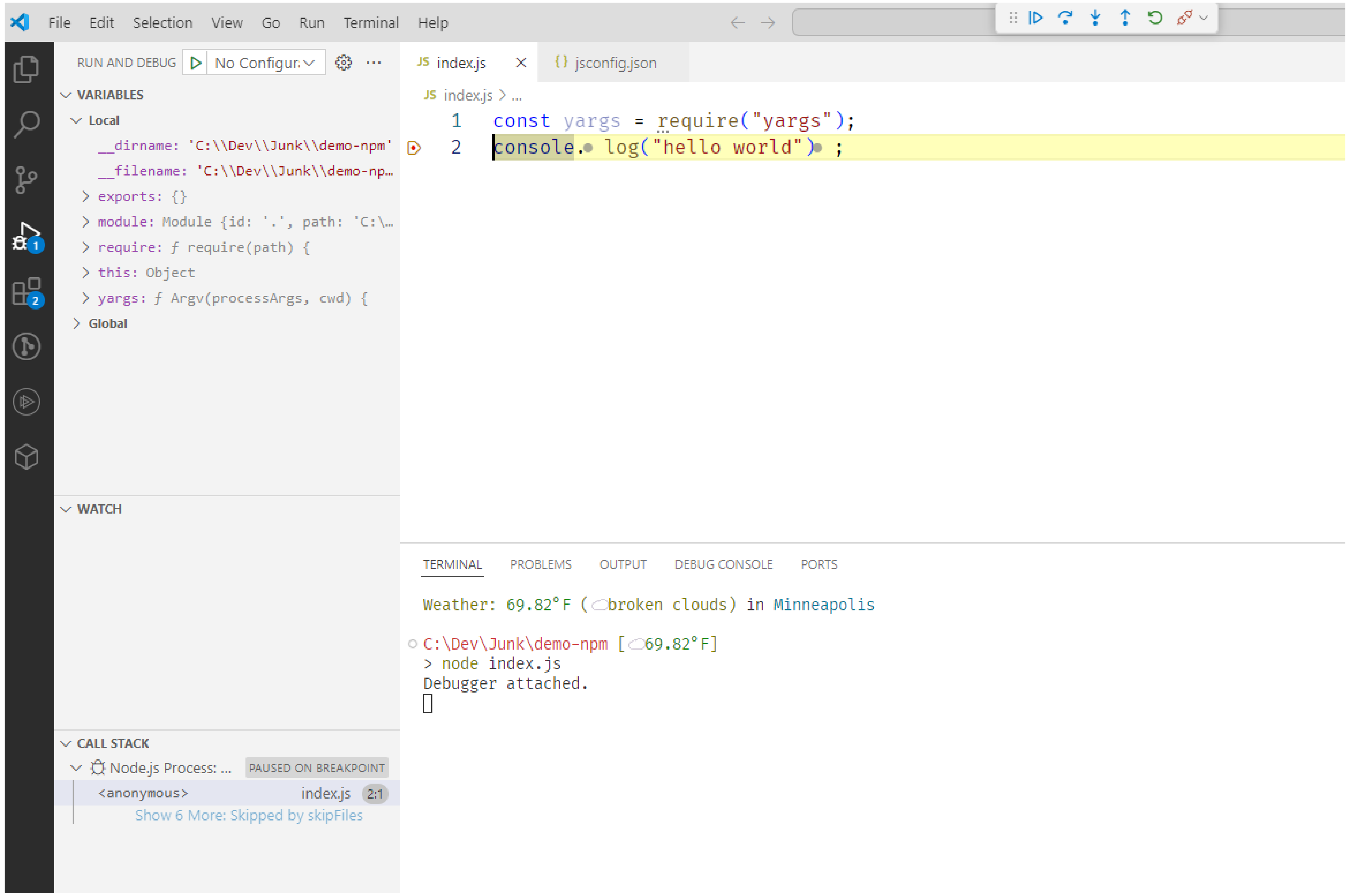
☐ Pause on uncaught exceptions
☐ Pause on caught exceptions

index.js
☑ console.log("hello … 2

```
Local
  this: Object
  exports: {}
  module: Module {id: '.', pa
  require: ƒ require(path)
  yargs: undefined
  __dirname: "C:\\Dev\\Junk\\
  __filename: "C:\\Dev\\Junk\
Global                      global
```

C:\WINDOWS\Syste...

Demo: Debugging JavaScript in IDEs



```
__dirname: 'C:\\Dev\\Junk\\demo-npm'
__filename: 'C:\\Dev\\Junk\\demo-np…
> exports: {}
> module: Module {id: '.', path: 'C:\…
> require: ƒ require(path) {
> this: Object
> yargs: ƒ Argv(processArgs, cwd) {
```

```
<anonymous>    index.js  2:1
Show 6 More: Skipped by skipFiles
```

```javascript
const yargs = require("yargs");
console.log("hello world");
```

```
Weather: 69.82°F (broken clouds) in Minneapolis

C:\Dev\Junk\demo-npm [69.82°F]
> node index.js
Debugger attached.
```

Launching the JavaScript Debugger

Demo

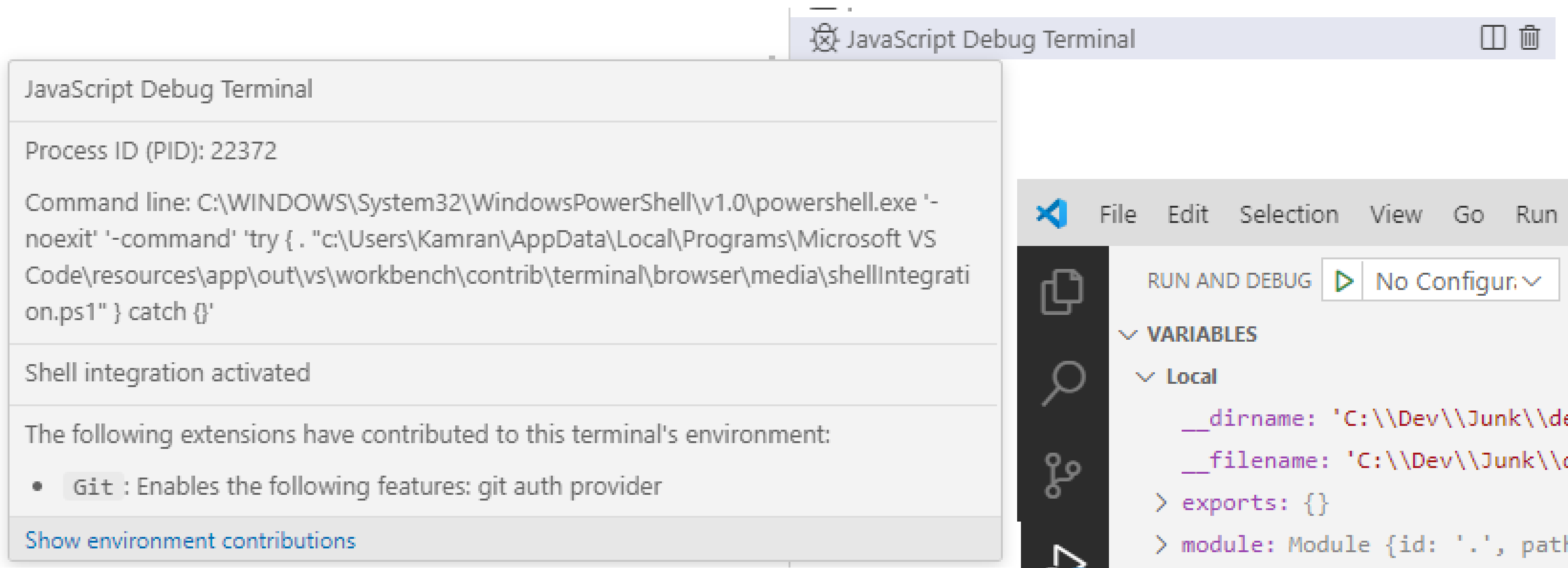
Setting breakpoints

Debug a Node.js app

Debug a browser app

Create a debug configuration

Demo: JavaScript Debug Terminal



JavaScript Debug Terminal

Process ID (PID): 22372

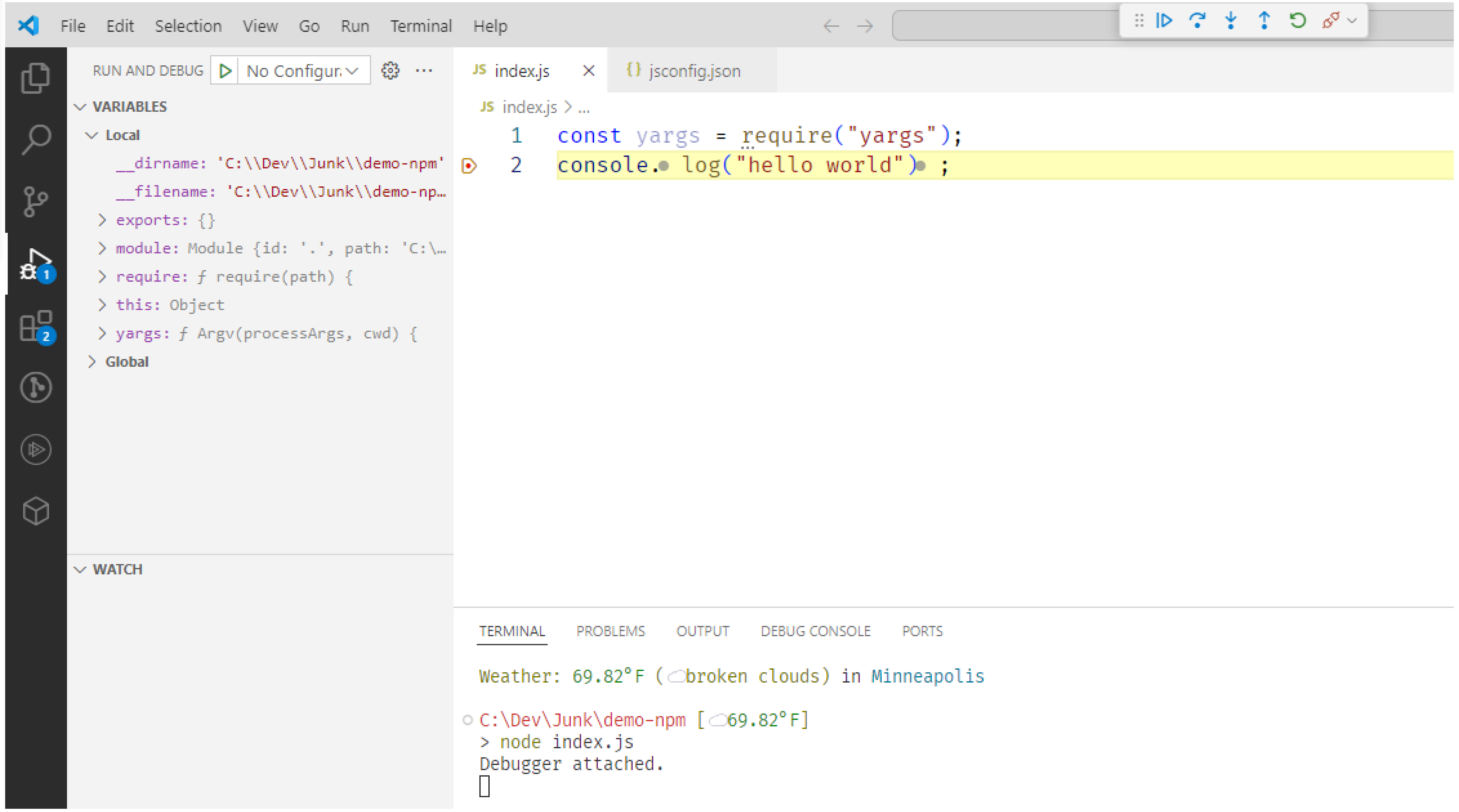
Command line: C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe '-noexit' '-command' 'try { . "c:\Users\Kamran\AppData\Local\Programs\Microsoft VS Code\resources\app\out\vs\workbench\contrib\terminal\browser\media\shellIntegration.ps1" } catch {}'

Shell integration activated

The following extensions have contributed to this terminal's environment:

- `Git` : Enables the following features: git auth provider

Show environment contributions



VARIABLES

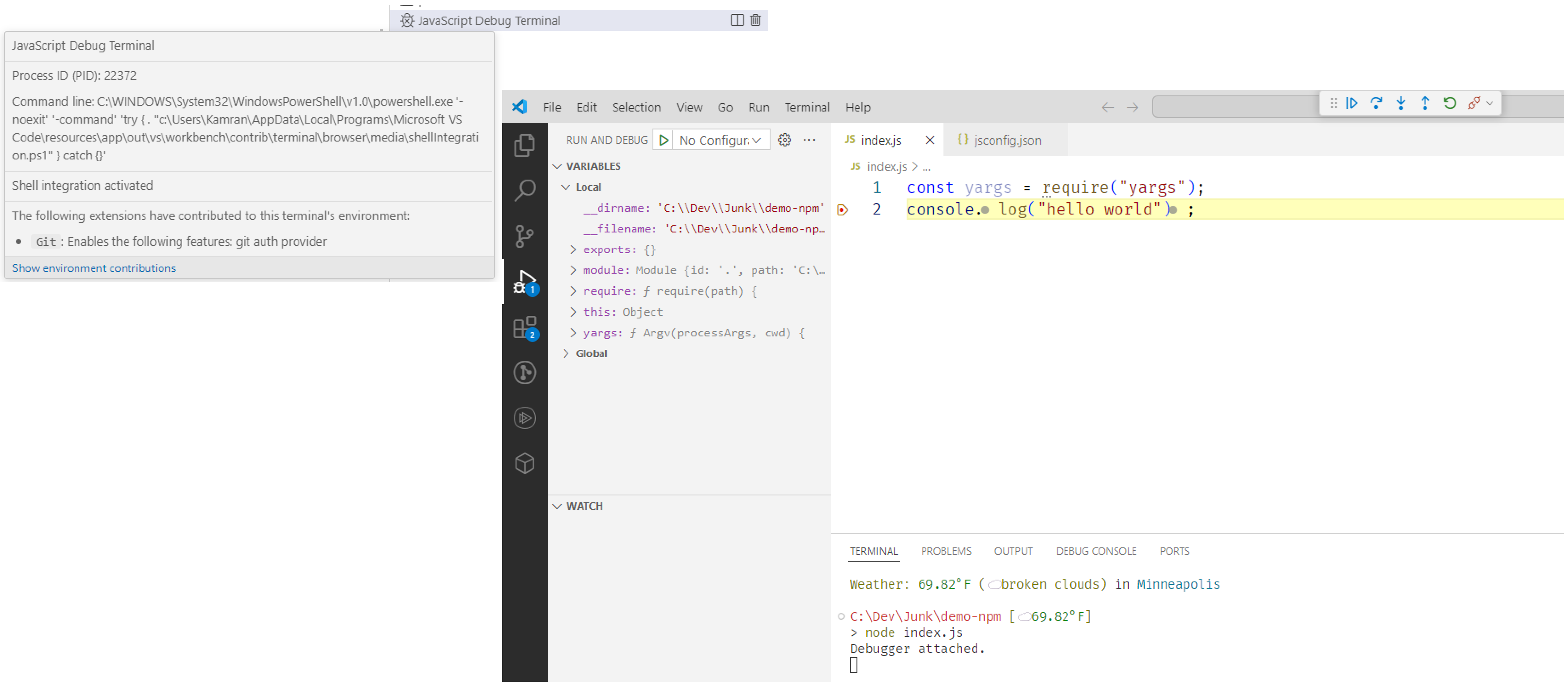
- Local
  - `__dirname: 'C:\\Dev\\Junk\\demo-npm'`
  - `__filename: 'C:\\Dev\\Junk\\demo-np…`
  - `exports: {}`
  - `module: Module {id: '.', path: 'C:\…`
  - `require: ƒ require(path) {`
  - `this: Object`
  - `yargs: ƒ Argv(processArgs, cwd) {`
- Global

WATCH

```javascript
const yargs = require("yargs");
console.log("hello world");
```

```
Weather: 69.82°F (☁broken clouds) in Minneapolis

C:\Dev\Junk\demo-npm [☁69.82°F]
> node index.js
Debugger attached.
```

Demo: Setting a Breakpoint



JavaScript Debug Terminal

Process ID (PID): 22372

Command line: C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe '-noexit' '-command' 'try { . "c:\Users\Kamran\AppData\Local\Programs\Microsoft VS Code\resources\app\out\vs\workbench\contrib\terminal\browser\media\shellIntegration.ps1" } catch {}'

Shell integration activated

The following extensions have contributed to this terminal's environment:

- `Git` : Enables the following features: git auth provider

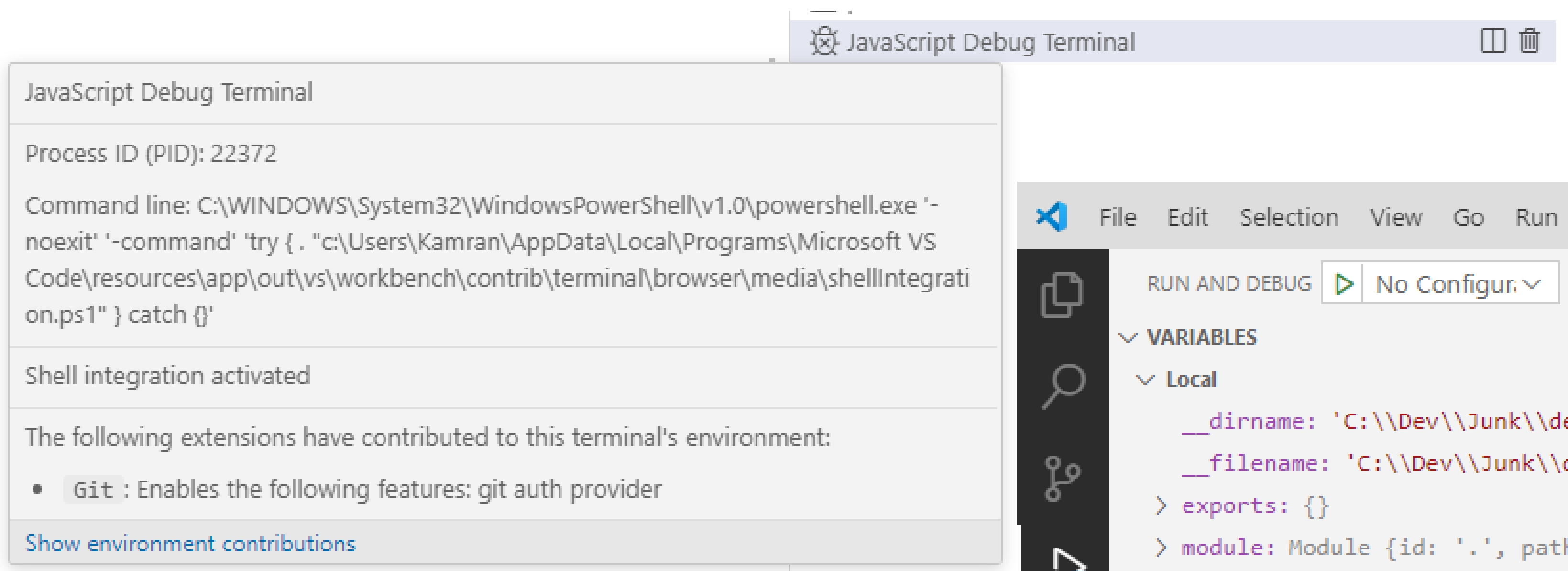
Show environment contributions

```
__dirname: 'C:\\Dev\\Junk\\demo-npm'
__filename: 'C:\\Dev\\Junk\\demo-np…
> exports: {}
> module: Module {id: '.', path: 'C:\…
> require: ƒ require(path) {
> this: Object
> yargs: ƒ Argv(processArgs, cwd) {
```

```
const yargs = require("yargs");
console.log("hello world");
```

```
Weather: 69.82°F (broken clouds) in Minneapolis

C:\Dev\Junk\demo-npm [69.82°F]
> node index.js
Debugger attached.
```

Demo: VS Code Debugger



JavaScript Debug Terminal

Process ID (PID): 22372

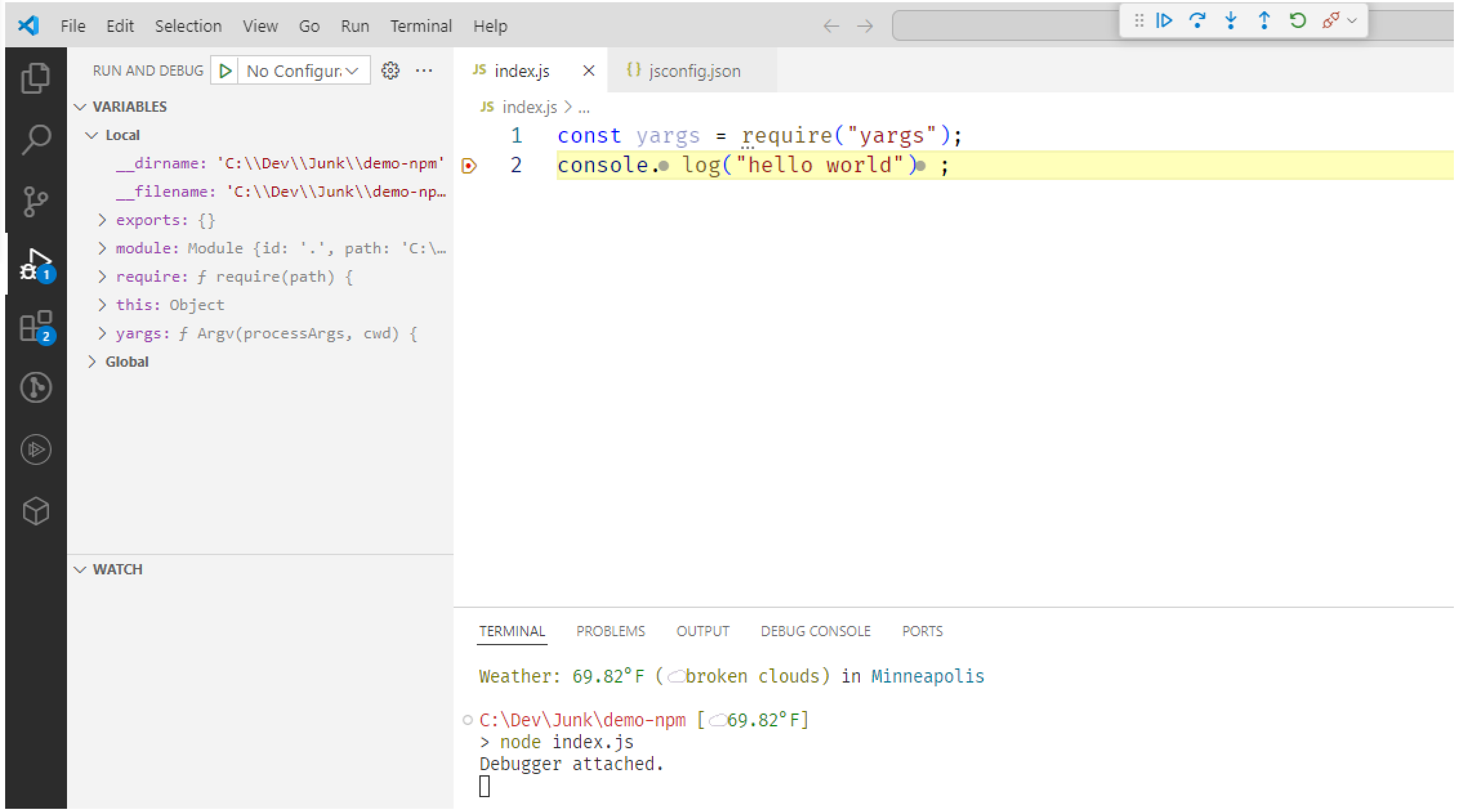
Command line: C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe '-noexit' '-command' 'try { . "c:\Users\Kamran\AppData\Local\Programs\Microsoft VS Code\resources\app\out\vs\workbench\contrib\terminal\browser\media\shellIntegration.ps1" } catch {}'

Shell integration activated

The following extensions have contributed to this terminal's environment:

- `Git` : Enables the following features: git auth provider

Show environment contributions

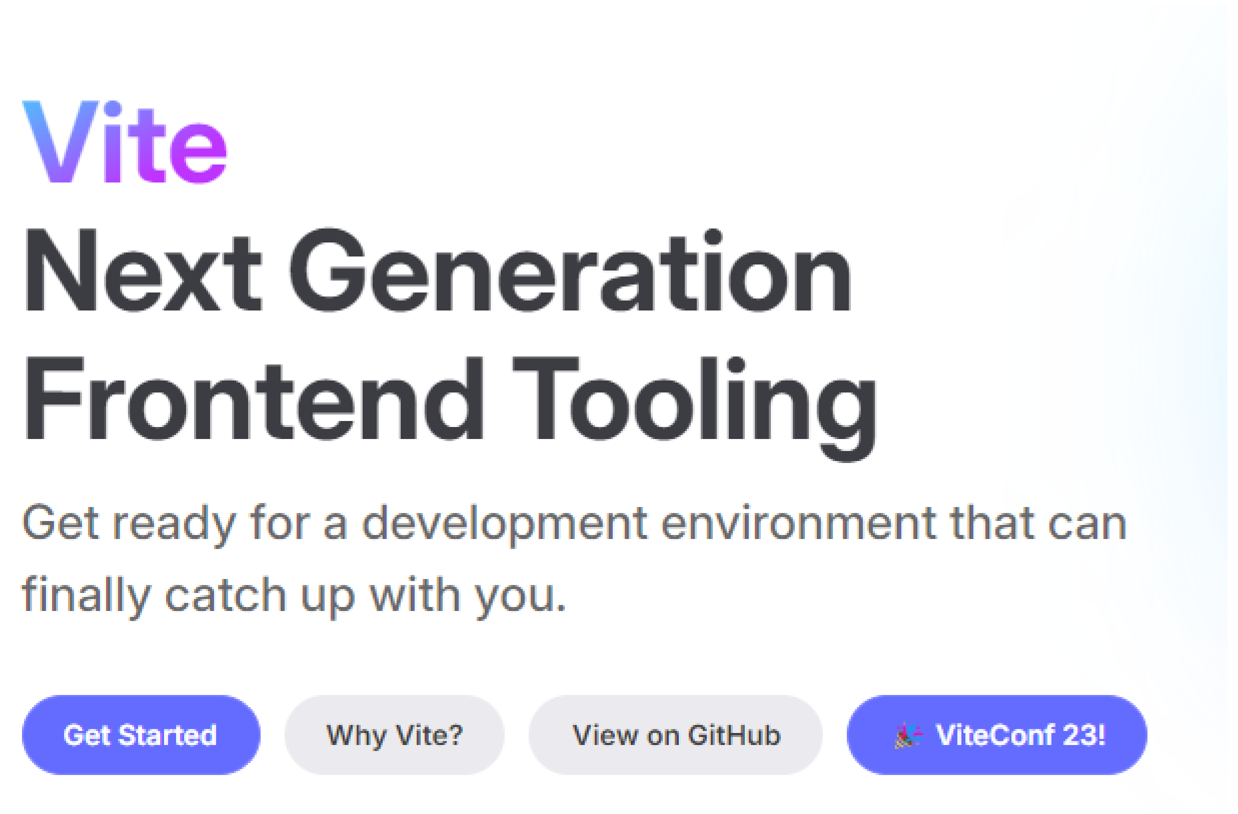


```
Local
  __dirname: 'C:\\Dev\\Junk\\demo-npm'
  __filename: 'C:\\Dev\\Junk\\demo-np…
> exports: {}
> module: Module {id: '.', path: 'C:\…
> require: ƒ require(path) {
> this: Object
> yargs: ƒ Argv(processArgs, cwd) {
Global
```

```js
const yargs = require("yargs");
console.log("hello world");
```

```
Weather: 69.82°F (broken clouds) in Minneapolis

C:\Dev\Junk\demo-npm [69.82°F]
> node index.js
Debugger attached.
```

Demo: Debugging Browser-Based Projects


Vite

Next Generation Frontend Tooling

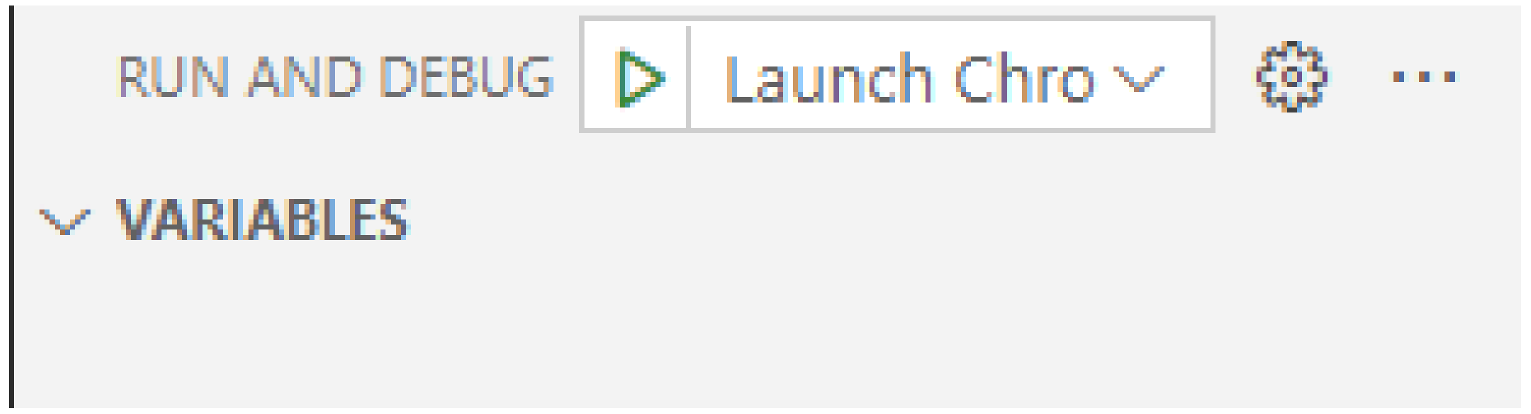
Get ready for a development environment that can finally catch up with you.

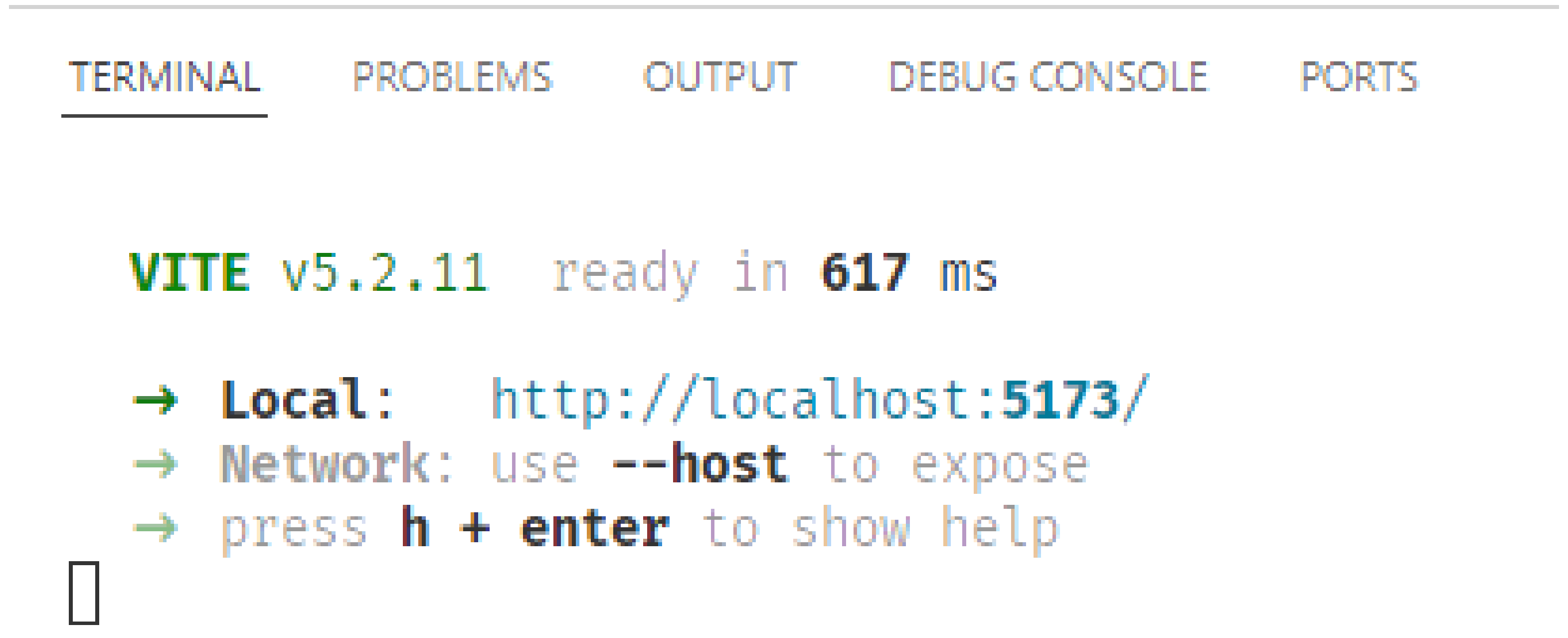
Demo: Creating a launch.json

Use Chrome configuration for browser-based projects

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "chrome",
      "request": "launch",
      "name": "Launch Chrome against localhost",
      "url": "http://localhost:5173",
      "webRoot": "${workspaceFolder}"
    }
  ]
}
```

Demo: Debugging Browser-Based Projects





```
  VITE v5.2.11  ready in 617 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

Debugging Resources

Debugging in Visual Studio Code
https://code.visualstudio.com/docs/editor/debugging

Configuring JavaScript Debugger in Web Storm
https://www.jetbrains.com/help/webstorm/configuring-javascript-debugger.html

Configuring JavaScript Debugger in IntelliJ IDEA
https://www.jetbrains.com/help/idea/configuring-javascript-debugger.html

Controlling the Debugger

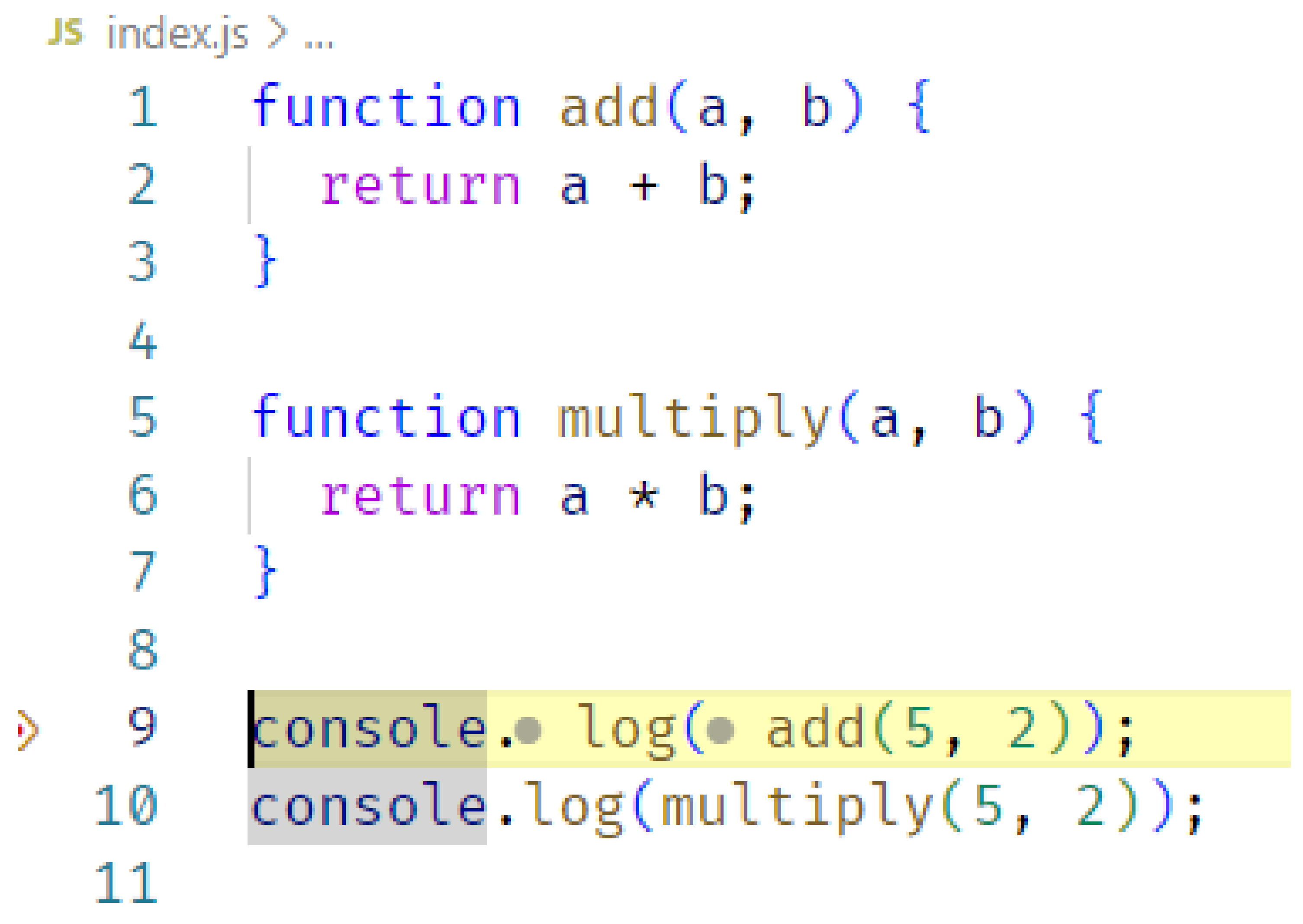
Demo

Step over, into, and out of code

Understand the call stack

Setting multiple breakpoints

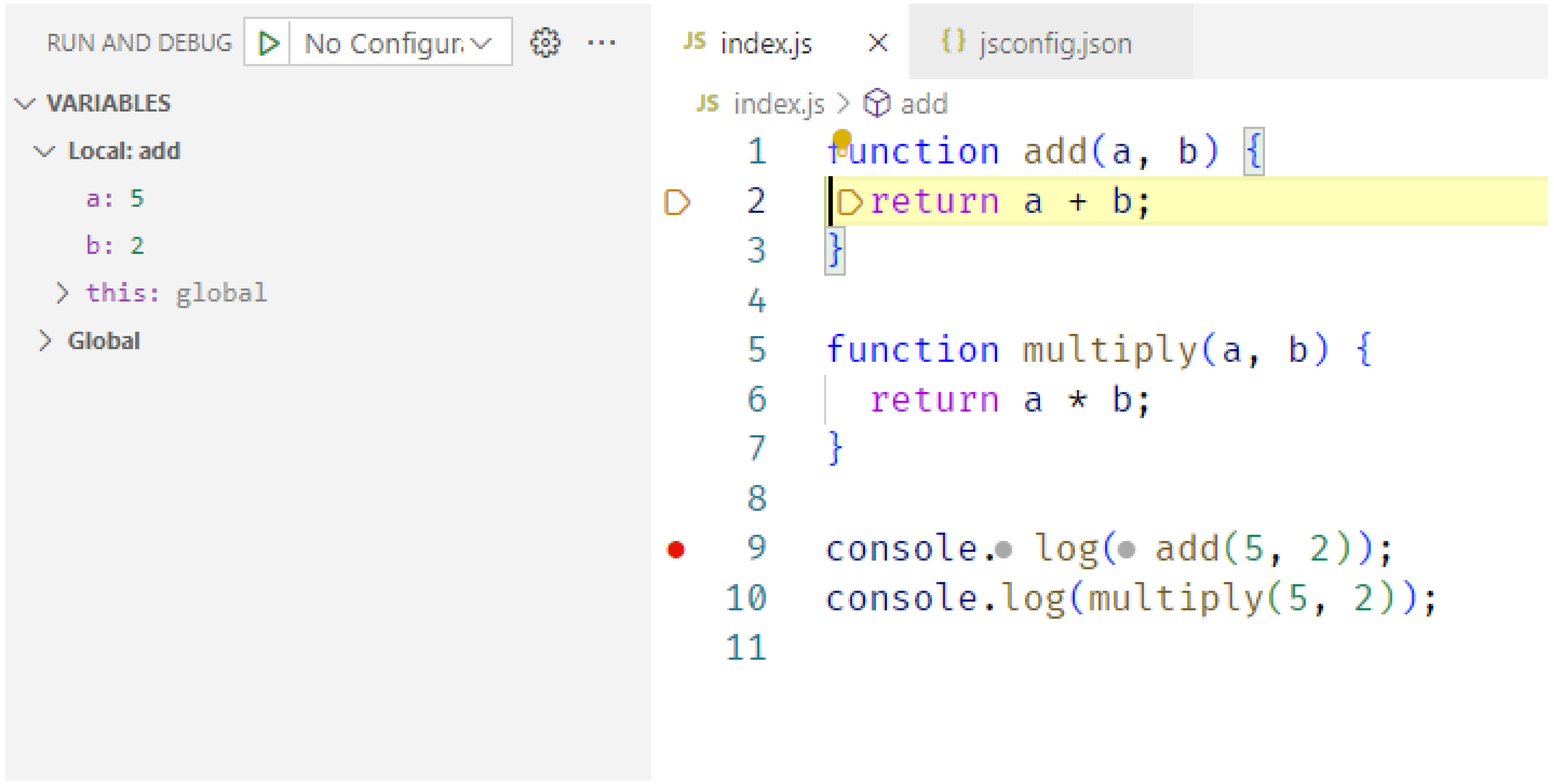
Demo: Step over code



index.js > ...

```js
function add(a, b) {
  return a + b;
}

function multiply(a, b) {
  return a * b;
}

console.log(add(5, 2));
console.log(multiply(5, 2));

```

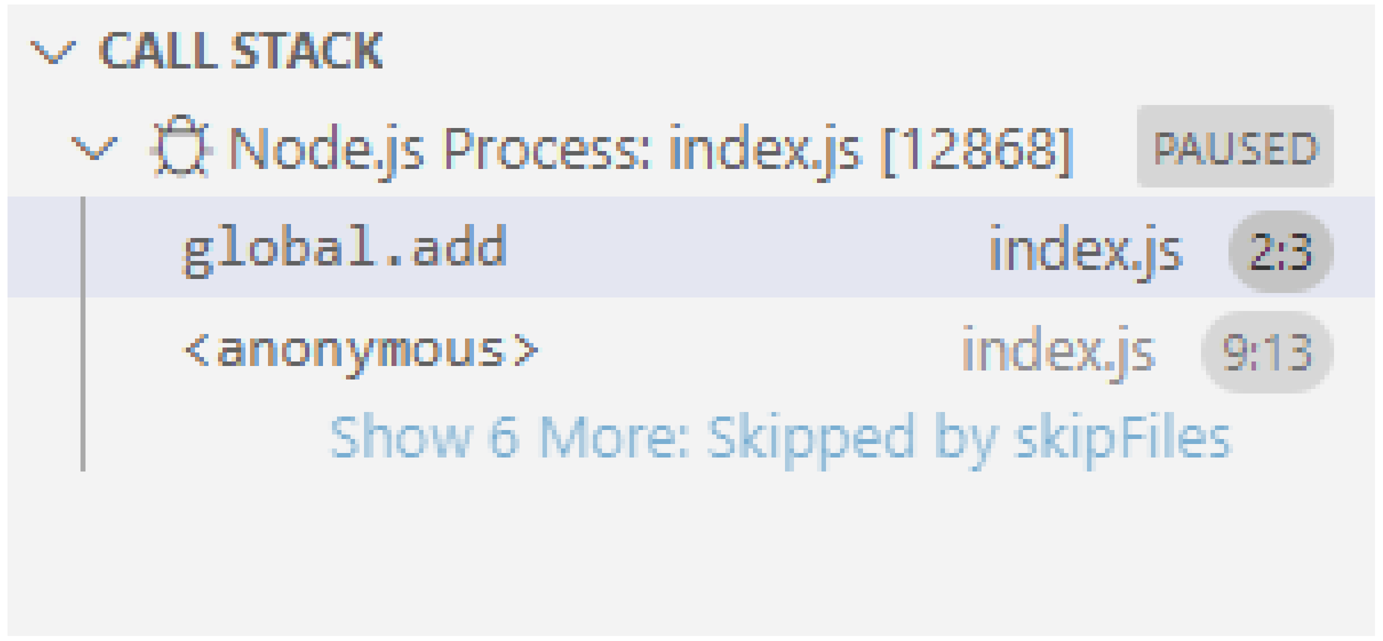
Demo: Step into



```
a: 5
b: 2
this: global
```

```js
function add(a, b) {
  return a + b;
}

function multiply(a, b) {
  return a * b;
}

console.log(add(5, 2));
console.log(multiply(5, 2));
```

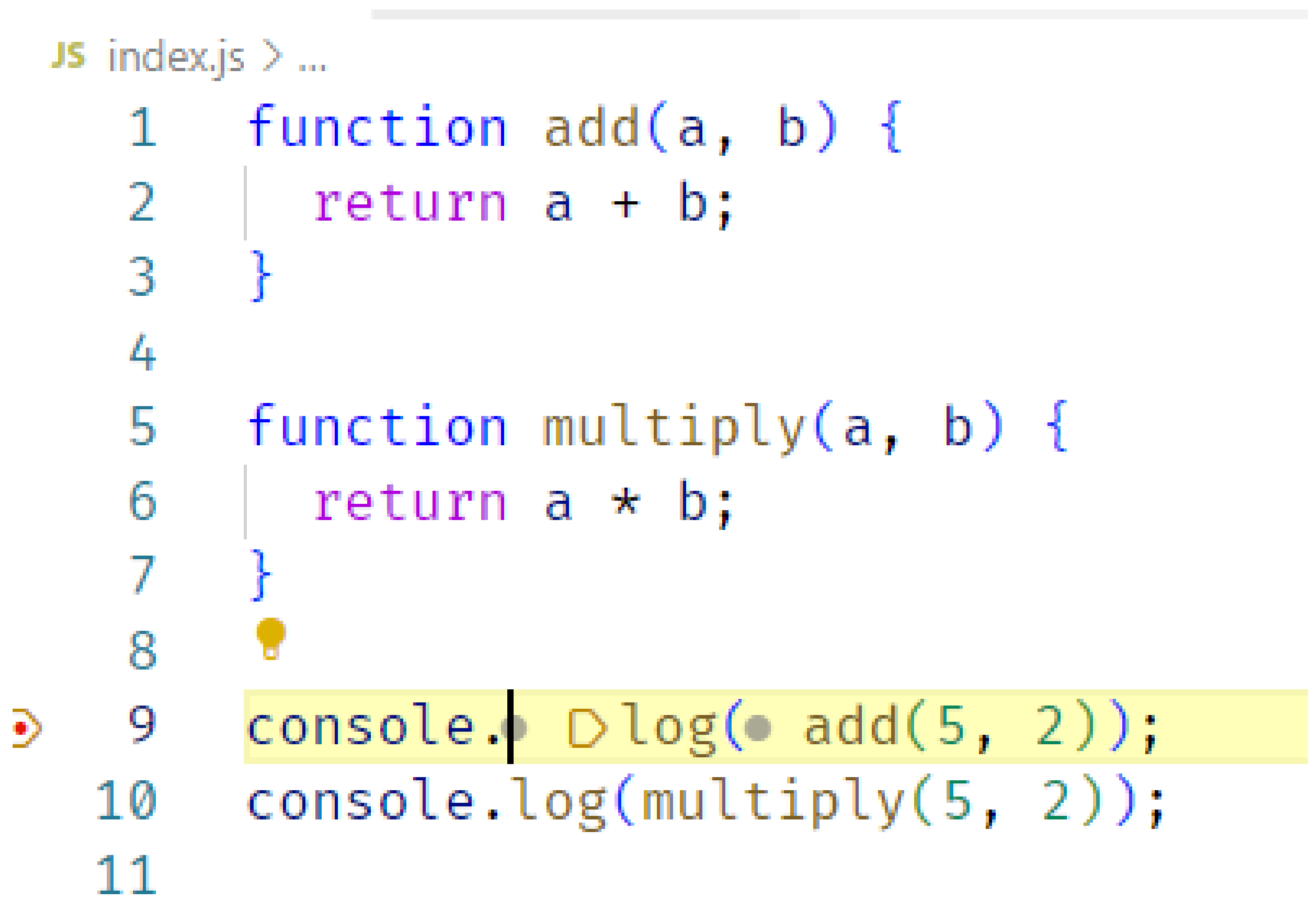
Demo: Call stack



CALL STACK
Node.js Process: index.js [12868] PAUSED

global.add	index.js	2:3
<anonymous>	index.js	9:13

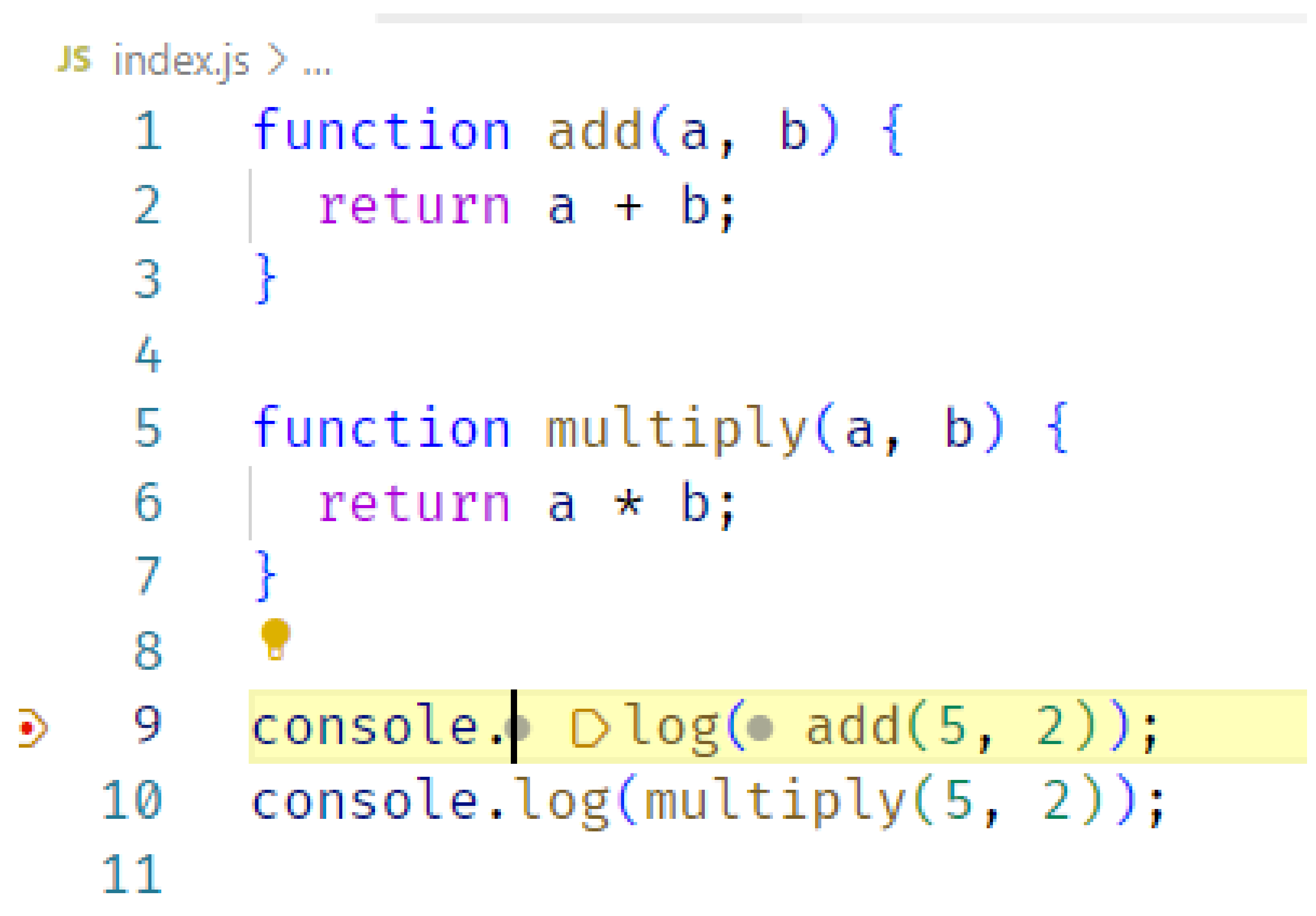
Show 6 More: Skipped by skipFiles



index.js > ...

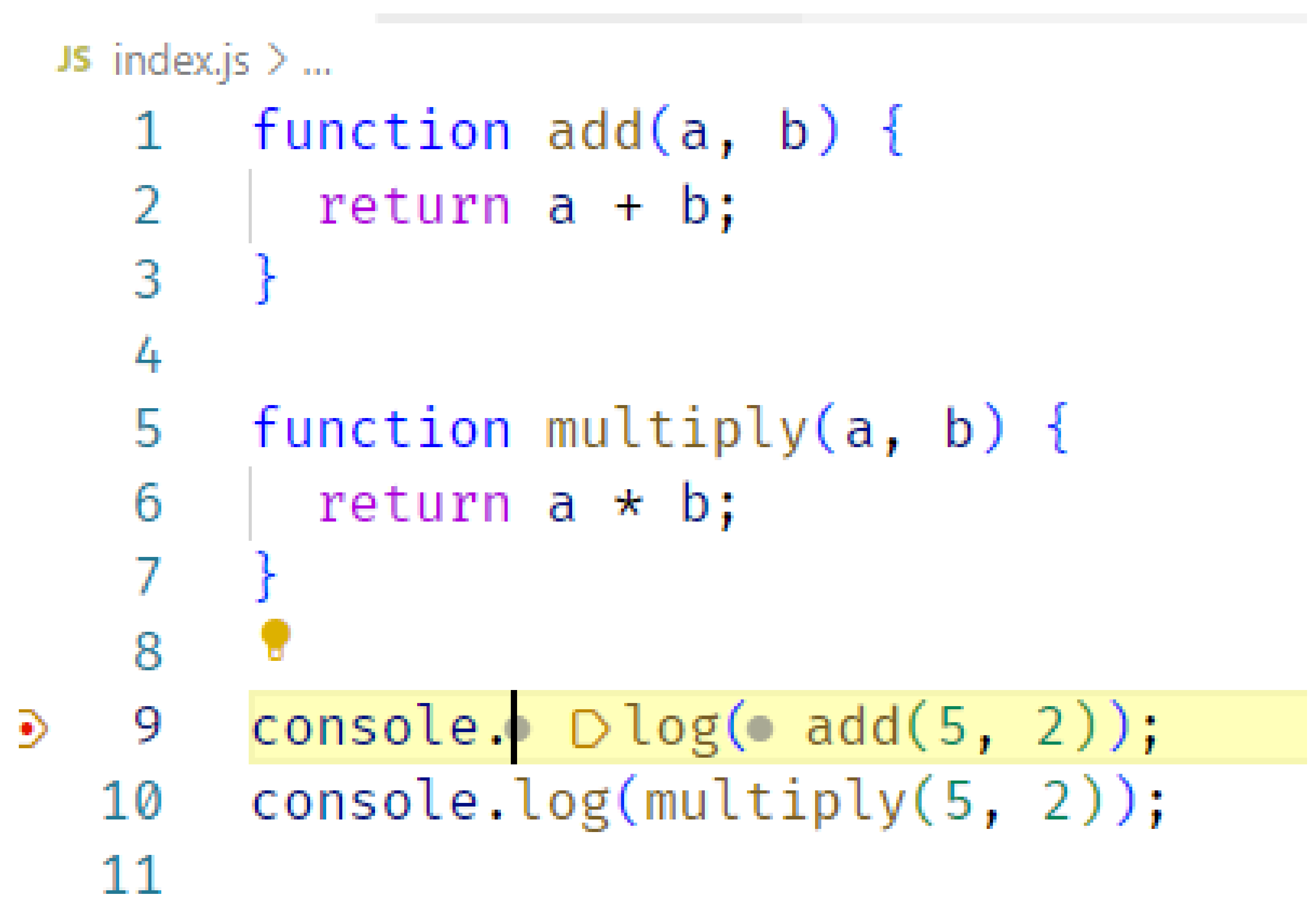
```
function add(a, b) {
  return a + b;
}

function multiply(a, b) {
  return a * b;
}

console.log(add(5, 2));
console.log(multiply(5, 2));
```

Demo: Step out


index.js > ...

```javascript
function add(a, b) {
  return a + b;
}

function multiply(a, b) {
  return a * b;
}

console.log(add(5, 2));
console.log(multiply(5, 2));

```

Demo: Continue


index.js > ...

```
function add(a, b) {
  return a + b;
}

function multiply(a, b) {
  return a * b;
}

console.log(add(5, 2));
console.log(multiply(5, 2));
```

Inspecting and Watching Variables

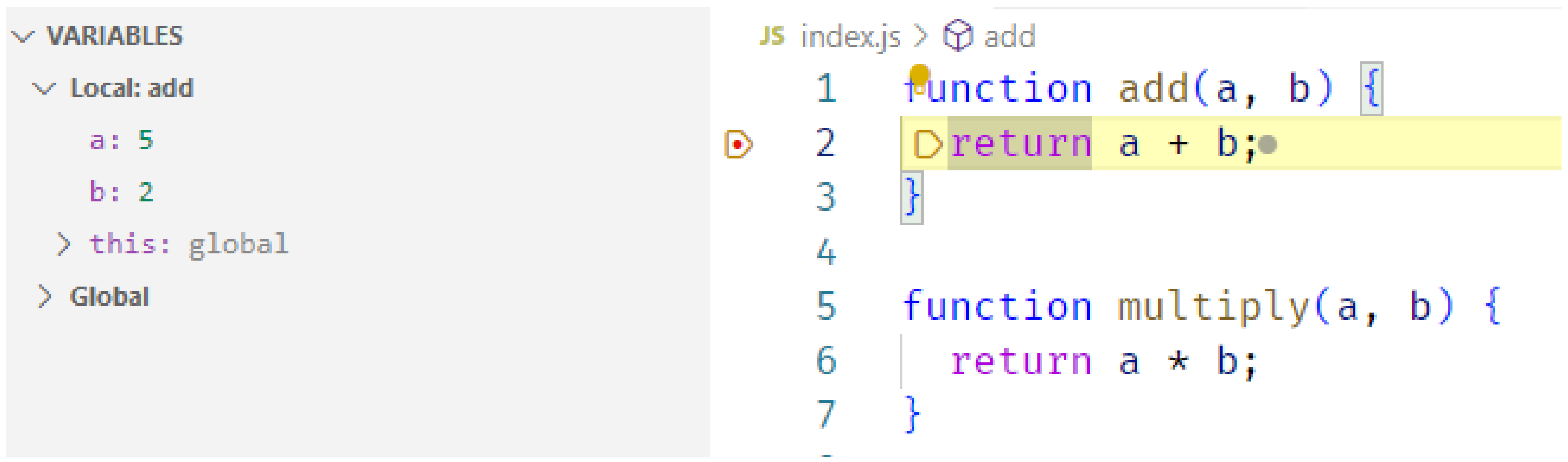
Demo

Inspecting variables

Watching variables

Executing ad-hoc code

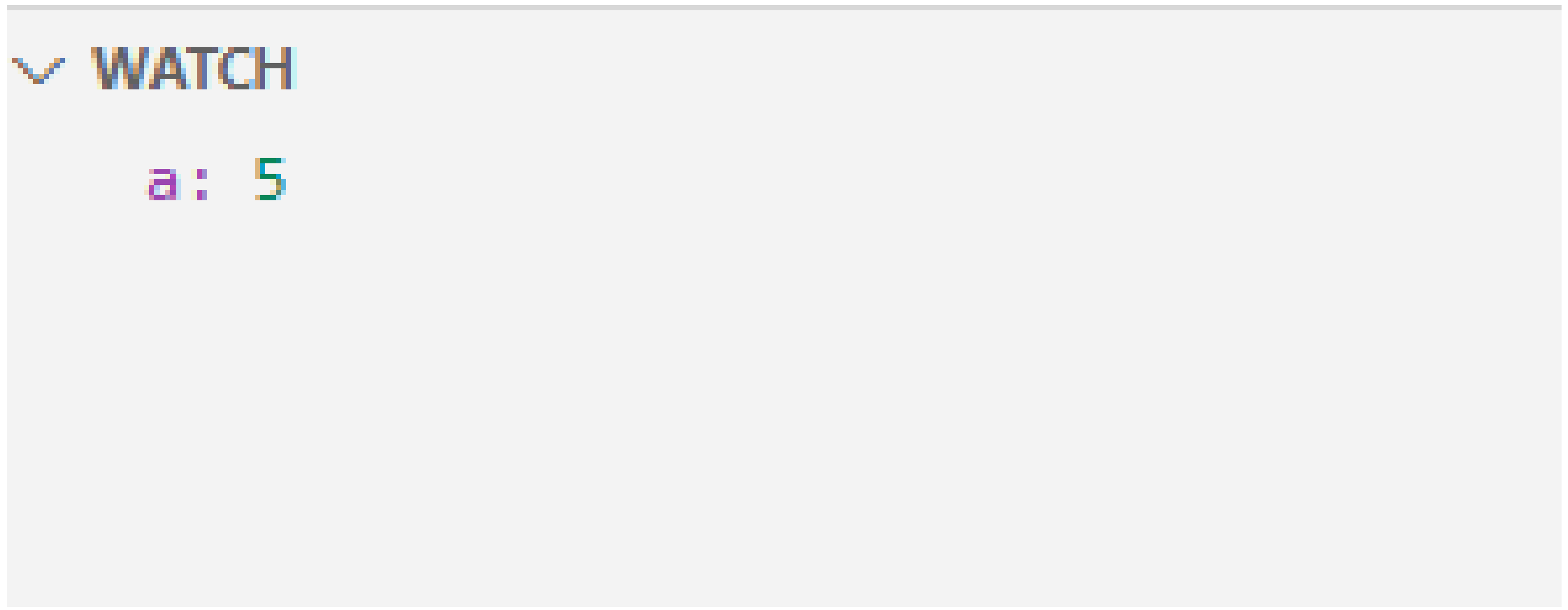
Demo: Variables list



```
VARIABLES
  Local: add
    a: 5
    b: 2
  > this: global
> Global
```

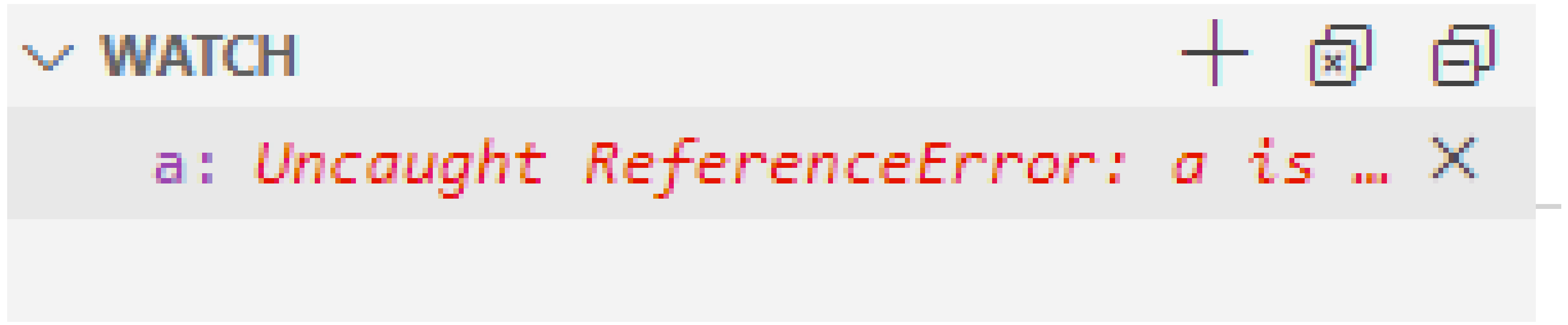
```
index.js > add
function add(a, b) {
  return a + b;
}

function multiply(a, b) {
  return a * b;
}
```

Demo: Watching variables



WATCH

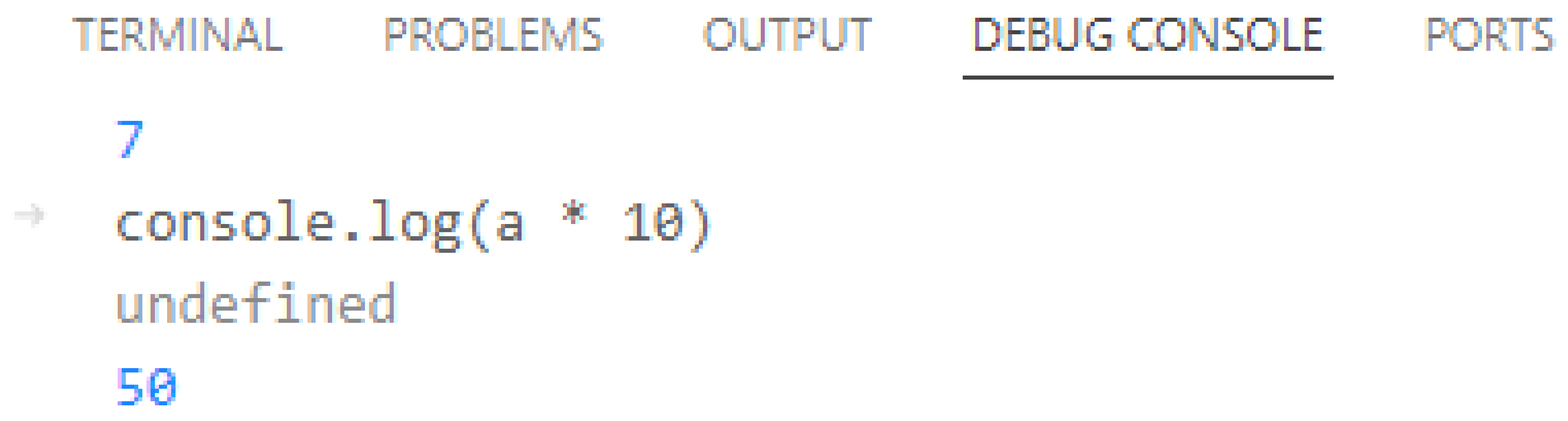
```
a: 5
```



WATCH

```
a: Uncaught ReferenceError: a is …
```

Demo: Immediate execution



```
7
console.log(a * 10)
undefined
50
```

Summary

IDEs provide powerful debugging capabilities to figure out issues quickly

You can debug both Node.js and browser apps with the same tools

Sharing debug configuration and tooling makes your whole team more effective

You're now equipped to be more productive on your JavaScript learning journey

Where to Go from Here

Course GitHub repository with files
https://bit.ly/PSJSDevEnvRepo

Like VS Code? Dive in deeper
https://code.visualstudio.com

JavaScript Learning Path on Pluralsight
https://app.pluralsight.com/paths/skill/javascript-2022

Thank You!

linkedin.com/in/kamranayub

kamran@kamranayub.com