# Debugging Like a Boss

A DEVELOPER'S GUIDE TO GREAT DEBUGGING EXPERIENCE

GIFT EGWUENU

lauragift21

**Imagine a world exists where your code works without errors and there's never anything as a bug**

- Likely a fairy tale

**We encounter bugs and errors daily as developers, how we fix these problems matters**

- Real life scenario

lauragift21

## Isn’t every developer go to method for debugging console.log() ?

It'll be great to discover other ways of debugging with JavaScript

lauragift21

lauragift21

## Procedure for Debugging

lauragift21

Identify the bug

## Procedure for Debugging

## Procedure for Debugging

Identify the bug

Reproduce the bug

lauragift21

## Procedure for Debugging

Identify the bug

Reproduce the bug

Find Possible Fix

lauragift21

## Procedure for Debugging

Identify the bug

Reproduce the bug

Find Possible Fix

Test and Document your solution.

## Step Up Your Debugging Skills

**Console Statements**

**Browser Developer Tools**

**IDE**

lauragift21

## Console Statements

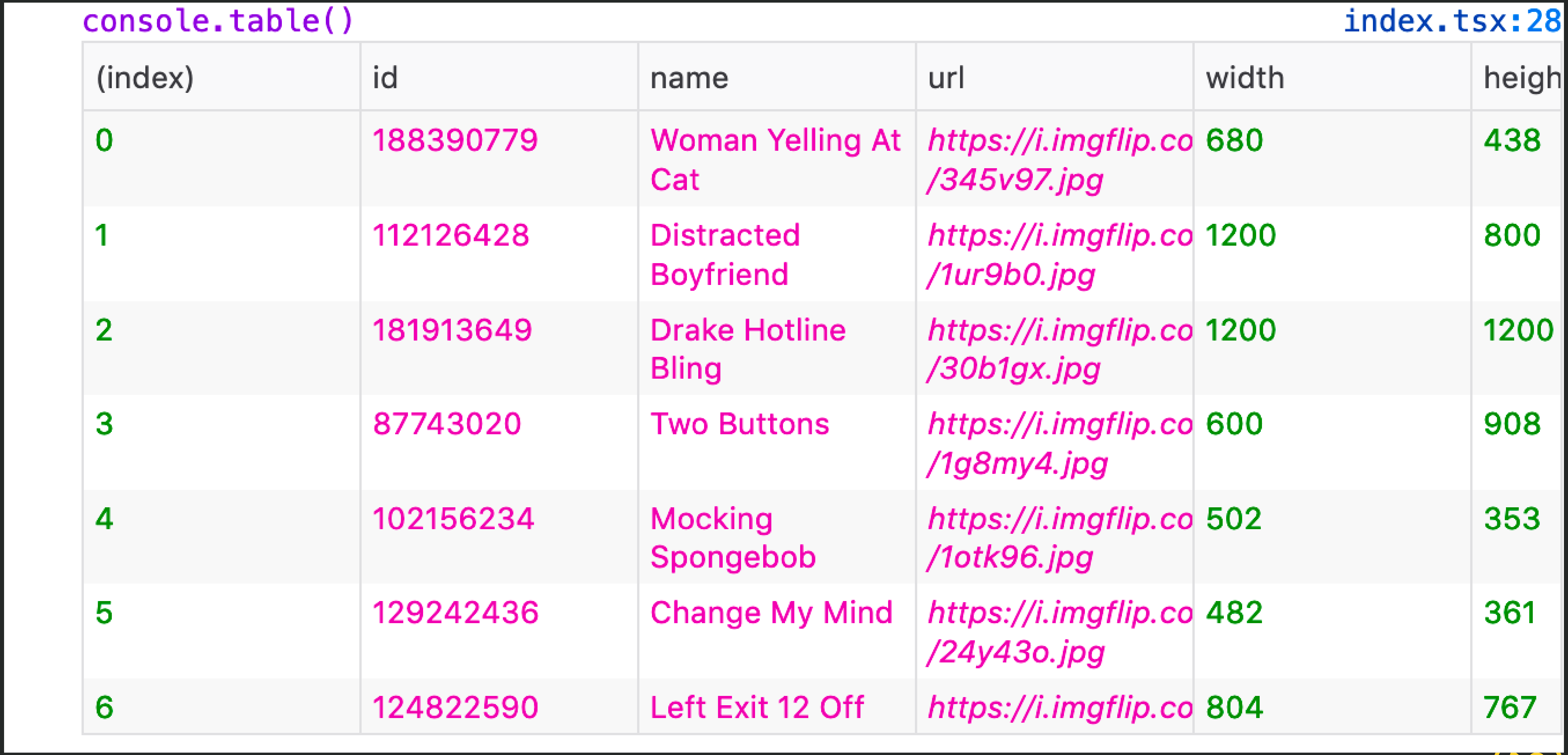
**Console.table( )**

**Console.time( ) &**
**Console.timeEnd( )**

lauragift21

# Console.table()

```
const { memes } = await imgData.data
console.table(memes);
```

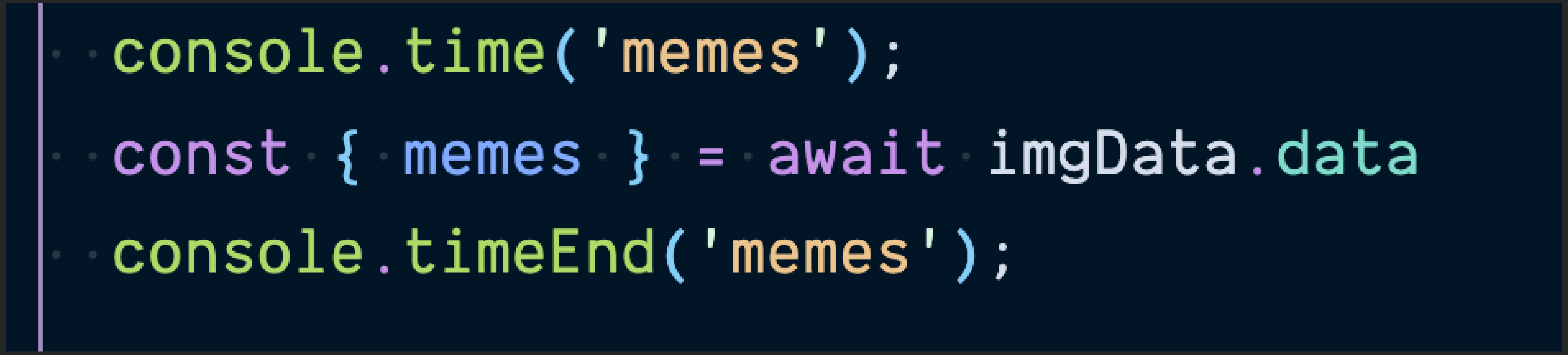


console.table()　　index.tsx:28

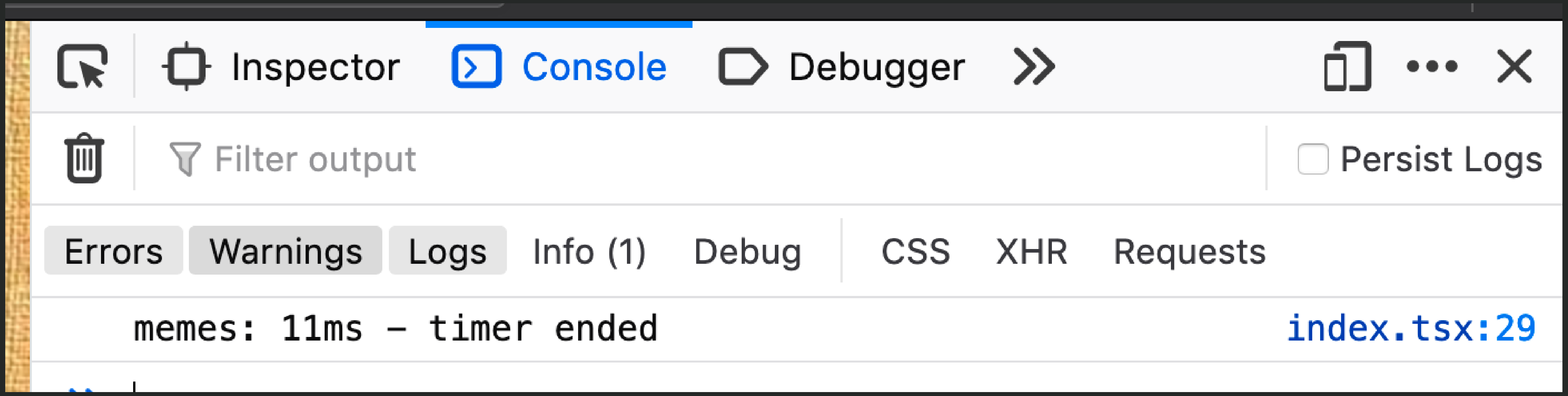
| (index) | id | name | url | width | heigh |
|---|---|---|---|---|---|
| 0 | 188390779 | Woman Yelling At Cat | https://i.imgflip.co /345v97.jpg | 680 | 438 |
| 1 | 112126428 | Distracted Boyfriend | https://i.imgflip.co /1ur9b0.jpg | 1200 | 800 |
| 2 | 181913649 | Drake Hotline Bling | https://i.imgflip.co /30b1gx.jpg | 1200 | 1200 |
| 3 | 87743020 | Two Buttons | https://i.imgflip.co /1g8my4.jpg | 600 | 908 |
| 4 | 102156234 | Mocking Spongebob | https://i.imgflip.co /1otk96.jpg | 502 | 353 |
| 5 | 129242436 | Change My Mind | https://i.imgflip.co /24y43o.jpg | 482 | 361 |
| 6 | 124822590 | Left Exit 12 Off | https://i.imgflip.co | 804 | 767 |

lauragift21

## Console.time() & Console.timeEnd



```
console.time('memes');
const { memes } = await imgData.data
console.timeEnd('memes');
```



```
memes: 11ms - timer ended                                    index.tsx:29
```

lauragift21

## Browser Developer Tools

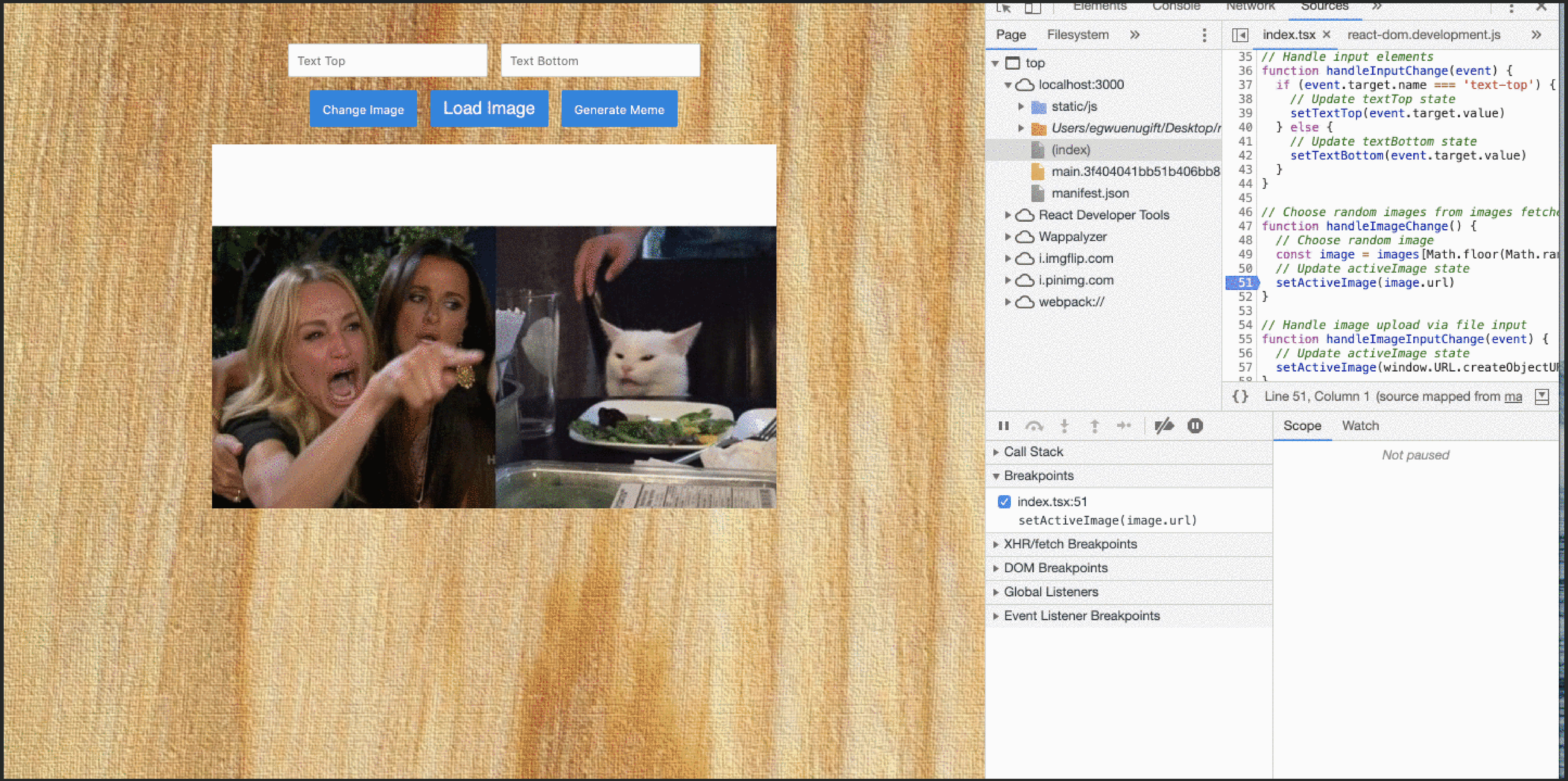
**Debugger**

**Breakpoints**

**Black Boxing**

lauragift21

# Debugger



```
35 // Handle input elements
36 function handleInputChange(event) {
37   if (event.target.name === 'text-top') {
38     // Update textTop state
39     setTextTop(event.target.value)
40   } else {
41     // Update textBottom state
42     setTextBottom(event.target.value)
43   }
44 }
45
46 // Choose random images from images fetche
47 function handleImageChange() {
48   // Choose random image
49   const image = images[Math.floor(Math.ra
50   // Update activeImage state
51   setActiveImage(image.url)
52 }
53
54 // Handle image upload via file input
55 function handleImageInputChange(event) {
56   // Update activeImage state
57   setActiveImage(window.URL.createObjectUR
```

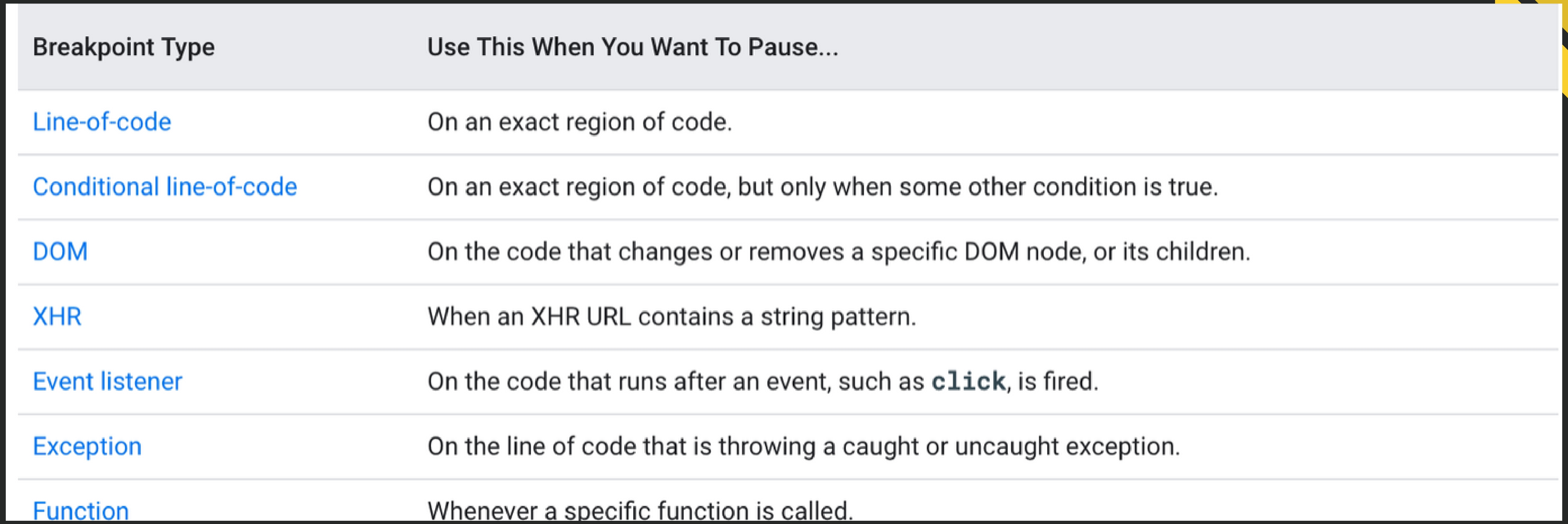
Line 51, Column 1 (source mapped from ma

☑ index.tsx:51
`setActiveImage(image.url)`

lauragift21

# Breakpoints

**A breakpoint is set wherever you want to pause a debugger execution**

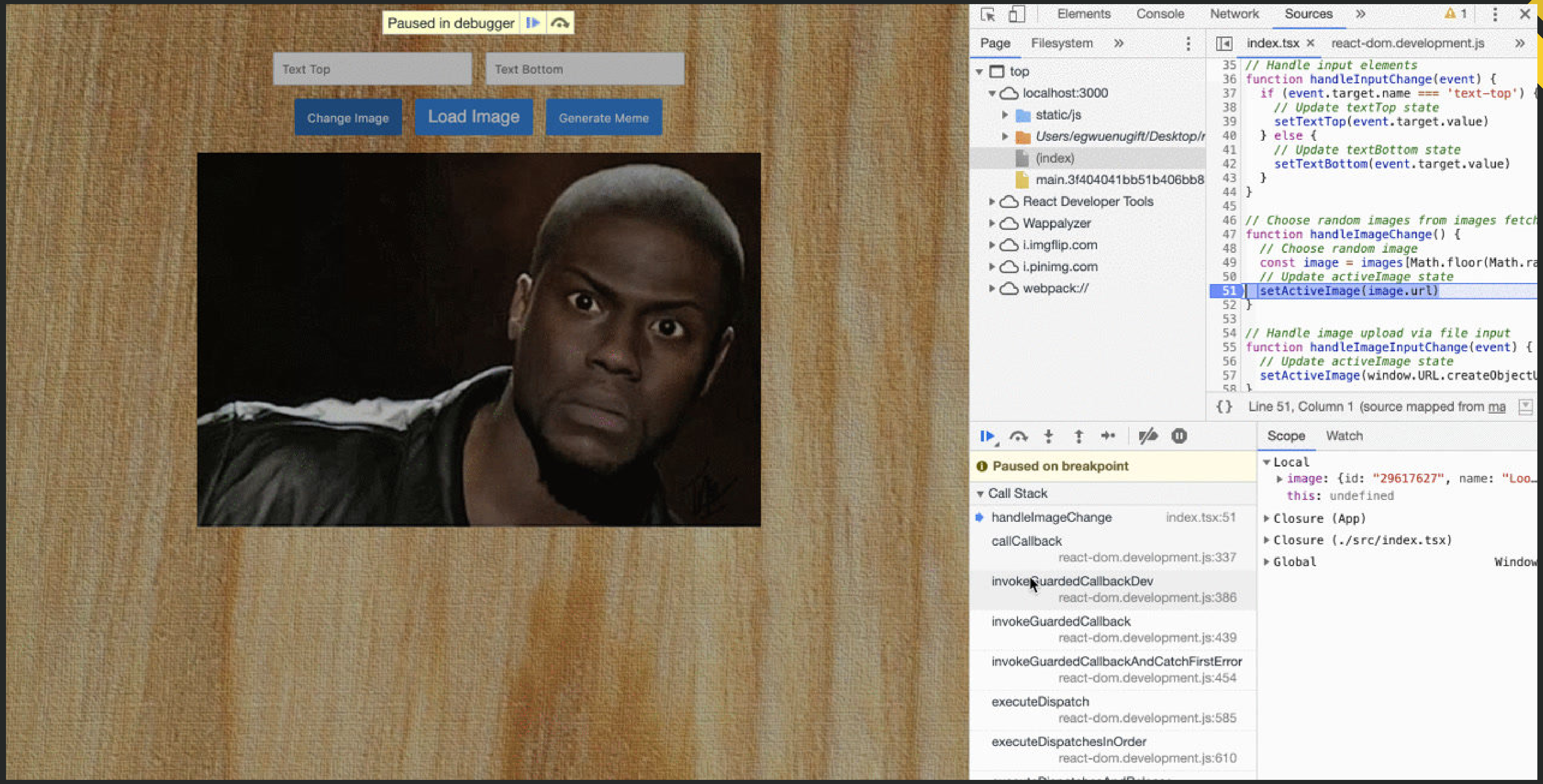


| Breakpoint Type | Use This When You Want To Pause... |
|---|---|
| Line-of-code | On an exact region of code. |
| Conditional line-of-code | On an exact region of code, but only when some other condition is true. |
| DOM | On the code that changes or removes a specific DOM node, or its children. |
| XHR | When an XHR URL contains a string pattern. |
| Event listener | On the code that runs after an event, such as `click`, is fired. |
| Exception | On the line of code that is throwing a caught or uncaught exception. |
| Function | Whenever a specific function is called. |

# Black Boxing

**A cool way to ignore library internals when debugging**

Blackboxing gives you a first-class way to denote library (or other abstraction) code so that the debugger can route around it



```
35 // Handle input elements
36 function handleInputChange(event) {
37   if (event.target.name === 'text-top') {
38     // Update textTop state
39     setTextTop(event.target.value)
40   } else {
41     // Update textBottom state
42     setTextBottom(event.target.value)
43   }
44 }
45
46 // Choose random images from images fetch
47 function handleImageChange() {
48   // Choose random image
49   const image = images[Math.floor(Math.ra
50   // Update activeImage state
51   setActiveImage(image.url)
52 }
53
54 // Handle image upload via file input
55 function handleImageInputChange(event) {
56   // Update activeImage state
57   setActiveImage(window.URL.createObjectU
```

Line 51, Column 1 (source mapped from ma

Paused on breakpoint

Call Stack

```
handleImageChange                      index.tsx:51
callCallback
                     react-dom.development.js:337
invokeGuardedCallbackDev
                     react-dom.development.js:386
invokeGuardedCallback
                     react-dom.development.js:439
invokeGuardedCallbackAndCatchFirstError
                     react-dom.development.js:454
executeDispatch
                     react-dom.development.js:585
executeDispatchesInOrder
                     react-dom.development.js:610
```

```
▼ Local
  ▶ image: {id: "29617627", name: "Loo…
    this: undefined
▶ Closure (App)
▶ Closure (./src/index.tsx)
▶ Global                              Window
```

## IDE

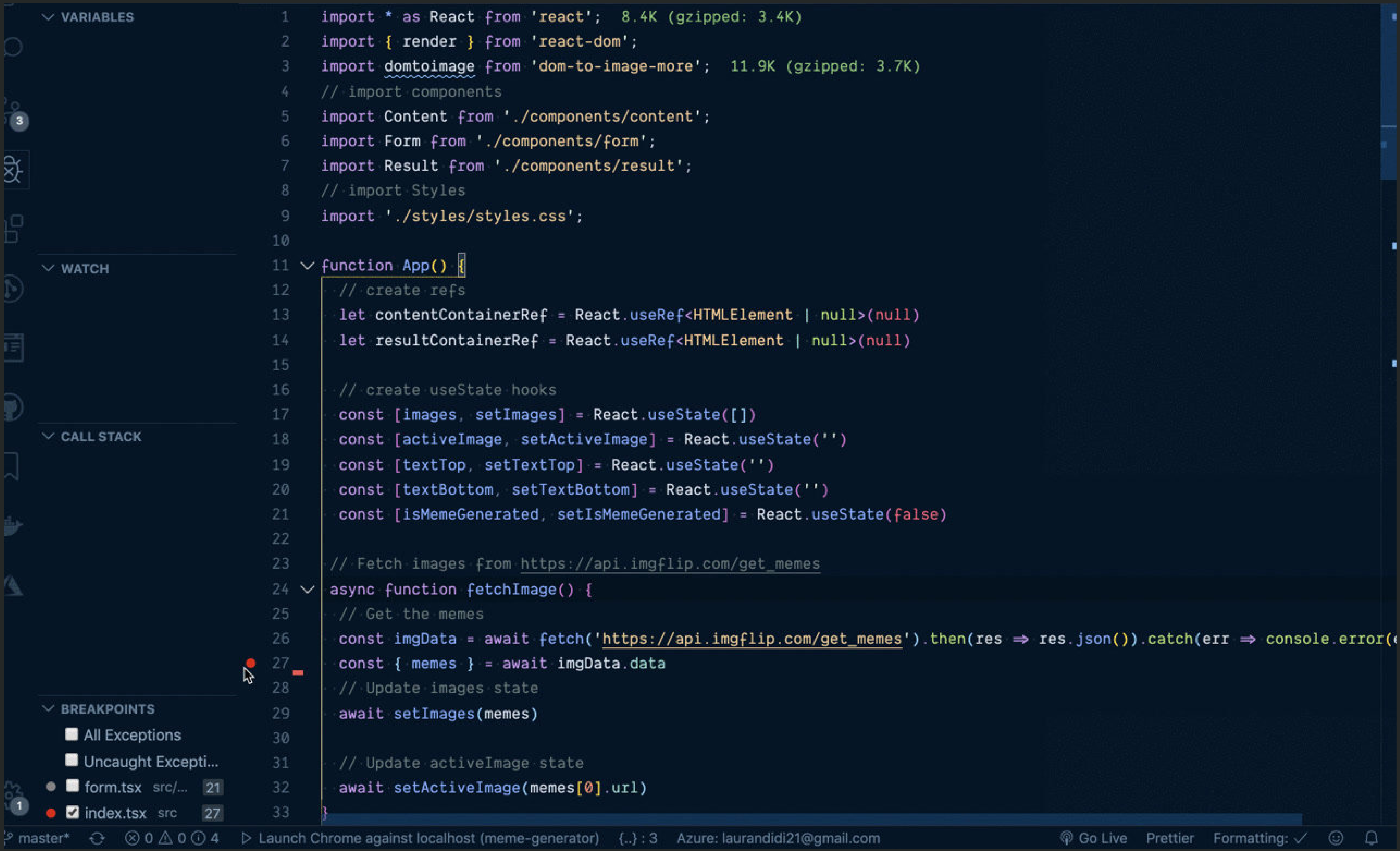
VS Code, Web Storm, Atom

## Debugger for Chrome Extension

Debug your JavaScript code running in Google Chrome from VS Code.

lauragift21

# Debugger for Chrome Extension with VS Code



VARIABLES

WATCH

CALL STACK

BREAKPOINTS

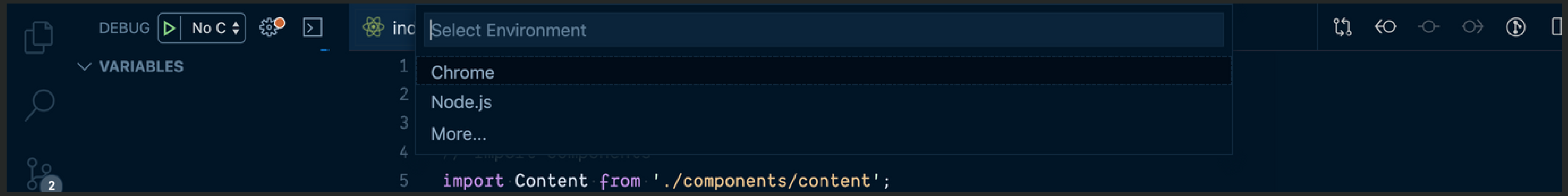
- [ ] All Exceptions
- [ ] Uncaught Excepti...
- [ ] form.tsx src/... 21
- [x] index.tsx src 27

```
import * as React from 'react';  8.4K (gzipped: 3.4K)
import { render } from 'react-dom';
import domtoimage from 'dom-to-image-more';  11.9K (gzipped: 3.7K)
// import components
import Content from './components/content';
import Form from './components/form';
import Result from './components/result';
// import Styles
import './styles/styles.css';

function App() {
  // create refs
  let contentContainerRef = React.useRef<HTMLElement | null>(null)
  let resultContainerRef = React.useRef<HTMLElement | null>(null)

  // create useState hooks
  const [images, setImages] = React.useState([])
  const [activeImage, setActiveImage] = React.useState('')
  const [textTop, setTextTop] = React.useState('')
  const [textBottom, setTextBottom] = React.useState('')
  const [isMemeGenerated, setIsMemeGenerated] = React.useState(false)

 // Fetch images from https://api.imgflip.com/get_memes
 async function fetchImage() {
  // Get the memes
  const imgData = await fetch('https://api.imgflip.com/get_memes').then(res => res.json()).catch(err => console.error(
  const { memes } = await imgData.data
  // Update images state
  await setImages(memes)

  // Update activeImage state
  await setActiveImage(memes[0].url)
}
```

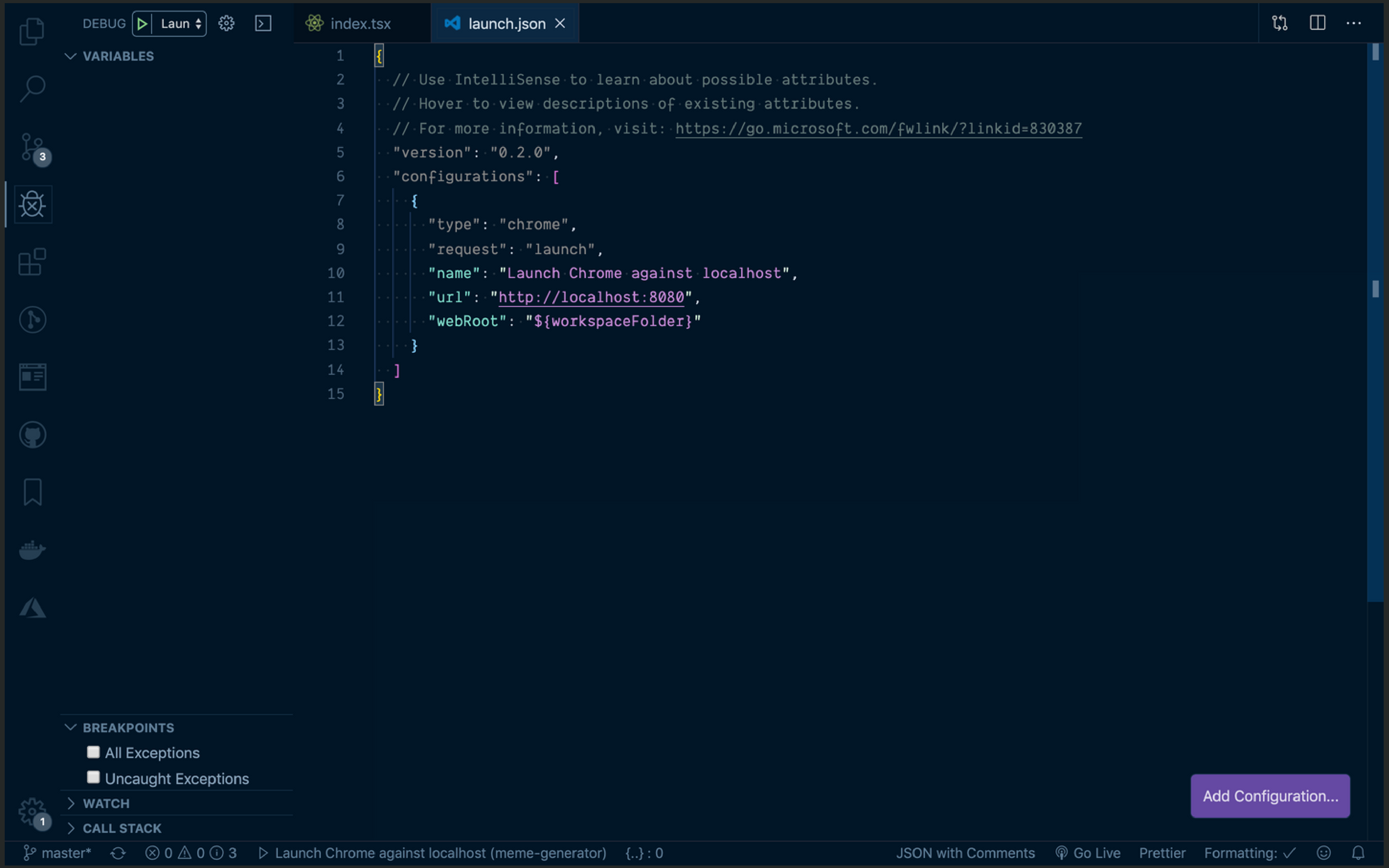
master* ⊗ 0 ⚠ 0 ⓘ 4 ▷ Launch Chrome against localhost (meme-generator) {..}: 3 Azure: laurandidi21@gmail.com Go Live Prettier Formatting: ✓

lauragift21

# Chrome Developer Tools Extension with VS Code



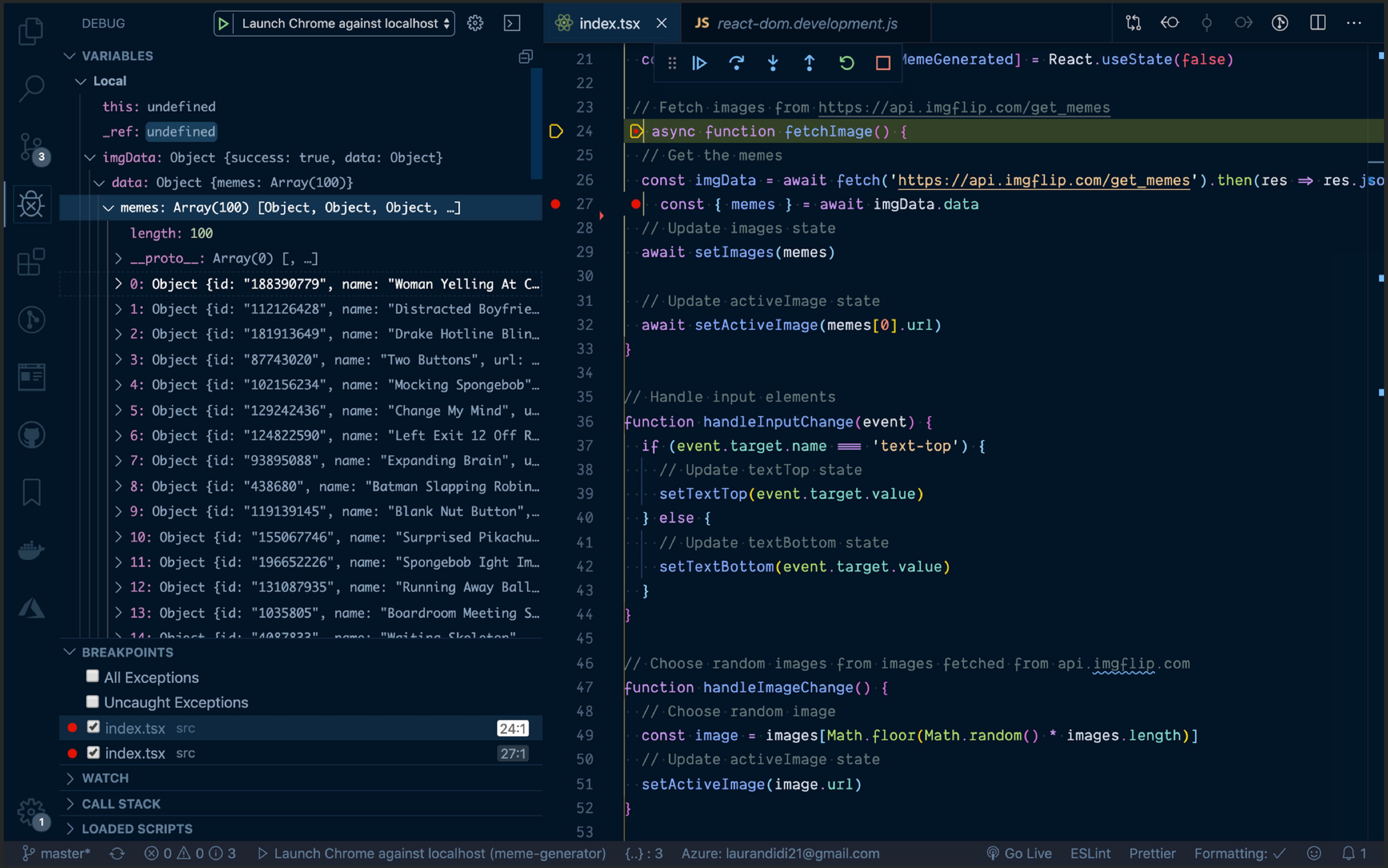
```
import Content from './components/content';
```



```
{
  // Use IntelliSense to learn about possible attributes.
  // Hover to view descriptions of existing attributes.
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
  "configurations": [
    {
      "type": "chrome",
      "request": "launch",
      "name": "Launch Chrome against localhost",
      "url": "http://localhost:8080",
      "webRoot": "${workspaceFolder}"
    }
  ]
}
```

lauragift21

# Chrome Developer Tools Extension with VS Code



```
VARIABLES
  Local
    this: undefined
    _ref: undefined
    imgData: Object {success: true, data: Object}
      data: Object {memes: Array(100)}
        memes: Array(100) [Object, Object, Object, …]
          length: 100
          > __proto__: Array(0) [, …]
          > 0: Object {id: "188390779", name: "Woman Yelling At C…
          > 1: Object {id: "112126428", name: "Distracted Boyfrie…
          > 2: Object {id: "181913649", name: "Drake Hotline Blin…
          > 3: Object {id: "87743020", name: "Two Buttons", url: …
          > 4: Object {id: "102156234", name: "Mocking Spongebob"…
          > 5: Object {id: "129242436", name: "Change My Mind", u…
          > 6: Object {id: "124822590", name: "Left Exit 12 Off R…
          > 7: Object {id: "93895088", name: "Expanding Brain", u…
          > 8: Object {id: "438680", name: "Batman Slapping Robin…
          > 9: Object {id: "119139145", name: "Blank Nut Button",…
          > 10: Object {id: "155067746", name: "Surprised Pikachu…
          > 11: Object {id: "196652226", name: "Spongebob Ight Im…
          > 12: Object {id: "131087935", name: "Running Away Ball…
          > 13: Object {id: "1035805", name: "Boardroom Meeting S…
          > 14: Object {id: "4087833", name: "Waiting Skeleton"…
BREAKPOINTS
  ☐ All Exceptions
  ☐ Uncaught Exceptions
  ● ☑ index.tsx src    24:1
  ● ☑ index.tsx src    27:1
> WATCH
> CALL STACK
> LOADED SCRIPTS
```

```tsx
  c…emeGenerated] = React.useState(false)

// Fetch images from https://api.imgflip.com/get_memes
async function fetchImage() {
  // Get the memes
  const imgData = await fetch('https://api.imgflip.com/get_memes').then(res => res.jso
    const { memes } = await imgData.data
  // Update images state
  await setImages(memes)

  // Update activeImage state
  await setActiveImage(memes[0].url)
}

// Handle input elements
function handleInputChange(event) {
  if (event.target.name === 'text-top') {
    // Update textTop state
    setTextTop(event.target.value)
  } else {
    // Update textBottom state
    setTextBottom(event.target.value)
  }
}

// Choose random images from images fetched from api.imgflip.com
function handleImageChange() {
  // Choose random image
  const image = images[Math.floor(Math.random() * images.length)]
  // Update activeImage state
  setActiveImage(image.url)
}
```

```
master*  ⊗ 0 ⚠ 0 ⓘ 3  ▷ Launch Chrome against localhost (meme-generator)  {..}: 3  Azure: laurandidi21@gmail.com  Go Live  ESLint  Prettier  Formatting: ✓  🔔 1
```

lauragift21

# Round Up

## OVERVIEW OF DEBUGGING

Talk through what debugging is all about and explain the process of debugging

## DEBUGGING PROCEDURES

- IDENTITY THE BUG
- REPRODUCE THE BUG
- FIX THE BUG
- TEST AND DOCUMEMNT THE FIX

## METHODS FOR DEBUGGING

- CONSOLE STATEMENTS
- DEVELOPER TOOLS
- IDE

lauragift21

# Thank You!

Slides: bit.ly/debugging-talk