

BUILDING SCALABLE AND DEPENDABLE

JAMUND FERGUSON

NODE.JS APPLICATIONS



The Mystery of the Missing Stack Trace

THE MYSTERY OF THE MISSING STACK TRACE

Error: Can't set headers after they are sent.

```
at ServerResponse.OutgoingMessage.setHeader (_http_outgoing.js:356:11)
at ServerResponse.header (/web/mynodeapp/node_modules/express/lib/response.js:767:10)
at ServerResponse.send (/web/mynodeapp/node_modules/express/lib/response.js:170:12)
at ServerResponse.res.send (/web/mynodeapp/node_modules/pplogger/index.js:225:18)
at done (/web/mynodeapp/node_modules/express/lib/response.js:1004:10)
at Stub.callback (/web/mynodeapp/node_modules/adaro/lib/engine.js:137:22)
at Stub.flush (/web/mynodeapp/pp/node_modules/dustjs-linkedin/lib/dust.js:513:10)
at Chunk.end (/web/mynodeapp/node_modules/dustjs-linkedin/lib/dust.js:612:15)
at /web/mynodeapp/node_modules/adaro/lib/patch/index.js:89:53
at /web/mynodeapp/node_modules/adaro/lib/reader/js.js:39:13
at /web/mynodeapp/node_modules/engine-munger/lib/munger.js:85:13
at /web/mynodeapp/node_modules/engine-munger/lib/cache.js:65:13
at /web/mynodeapp/node_modules/graceful-fs/graceful-fs.js:78:16
at /web/mynodeapp/node_modules/async-listener/glue.js:188:31
at FSReqWrap.readFileAfterClose [as oncomplete] (fs.js:380:3)
```

crashes • Paypal

Search

Home Organize Tools

New Email New Items Delete Archive Reply Reply All Forward Meeting Move Junk Rules Read/Unread Categorize Follow Up Filter Email Address Book Send & Receive Store

Favorites

Inbox 8997

Paypal

Inbox 8997

crashes 59567

- feedback
- github 15334
- hiring
- important
- lists 1278
- projects

Sent

Archive 273

Trash 16894

Drafts 229

Junk 4

Clutter 18406

Conversation History

p2p

Subscribed Public Fol...

Smart Folders

Items: 642110 Unread: 59567

Last synced at 3/3/18, 8:16 PM. Connecting to: Paypal

authnodeweb just restarted on dcg12authnodeweb3620:1 for time 6

website@paypal.com
Monday, January 1, 2018 at 12:55 AM
To: DL-PP-NodeJS-LiveMonitoring; DL-PP-AuthChallenge-Tech

W

walletexpnodedweb just restarted on d... 1/1/18
Mon Dec 25 2017 04:33:11 GMT-0800 (PST) 'ser...

website@paypal.com
bizwalletnodeweb just restarted on dcg13... 1/1/18
18-1-1:01:01:01 UNCAUGHTEXCEPTION TypeErro...

website@paypal.com
reportingnodeweb just restarted on slcre... 1/1/18
at trim_prefix (/dependencies/reportingnodeweb/...

website@paypal.com
walletexpnodedweb just restarted on d... 1/1/18
Thu Dec 28 2017 19:20:02 GMT-0800 (PST) 'serv...

website@paypal.com
walletexpnodedweb just restarted on dcg0... 1/1/18
at Function.process_params

website@paypal.com
authnodeweb just restarted on dcg12aut... 1/1/18
debug_id: '9be667f98721' } Sun Dec 31 2017 23:...

website@paypal.com
bizprofilenodeweb just restarted on dcg1... 1/1/18
2017-12-25T04:26:00-08:00:ERROR - An error o...

website@paypal.com
walletexpnodedweb just restarted on d... 1/1/18
errorCode: undefined, statusCode: 500, authFlow...

Older



*We had no idea why or where our apps
were failing*

WHAT CAN WE DO?

- ▶ Compare a **diff** between last known working code
- ▶ Look through other logs for more information (**nginx logs, access logs, etc**)
- ▶ Look at **system metrics** (is there a memory leak or CPU spike?)
- ▶ Add **console.log** statements somewhere???
- ▶ Advanced debugging techniques (**post-mortem debugging, heapdumps, etc.**)

PayPal: Send & Request

PayPal, Inc. [US] | https://www.paypal.com/myaccount/transfer/request

Summary Activity Send & Request Wallet Benefits Help LOG OUT

Request money

You can request money from anyone with an email address or phone number, even if they don't have a PayPal account. It's free for friends, or covered by **PayPal Seller Protection** for eligible sales.

Emails, Mobile Numbers or Names

Next

AB	RT	MH	AB	MK
N. Nursalim	R. Thomas	M. Harrison	A. Becker	I. Iaron
VH	MW	RT	JB	MV
V. Harrison	M. Weltner	S. Green	J. Berres	M. Kelly

Edit Contacts

This screenshot shows the 'Send & Request' section of the PayPal website. At the top, there are tabs for Summary, Activity, Send & Request (which is selected), Wallet, Benefits, and Help, along with a Log Out button. Below the tabs, the page title is 'Request money'. A descriptive text explains that users can request money from anyone with an email address or phone number, even if they don't have a PayPal account. It mentions that it's free for friends and is covered by 'PayPal Seller Protection' for eligible sales. There is a search bar labeled 'Emails, Mobile Numbers or Names' and a 'Next' button. Below this, there are two rows of contact cards. Each card contains initials and a name. The first row includes AB (N. Nursalim), RT (R. Thomas), MH (M. Harrison), AB (A. Becker), and MK (I. Iaron). The second row includes VH (V. Harrison), MW (M. Weltner), RT (S. Green), JB (J. Berres), and MV (M. Kelly). At the bottom, there is a 'Edit Contacts' link.

```
function payRequest(req, res) {  
    // all the important pay request logic...  
  
    // send back the request id  
    res.send(requestId)  
  
    // if sending to a known sender...  
    if (senderId) {  
        await optInToExperiment(senderId)  
    }  
}
```

```
async function optInToExperiment(senderId) {  
  let encryptedId = decrypt(senderId)  
  expService.addUser(encryptedId, function(err, res) {  
    if (err) return logger.log(res)  
    // success  
  })  
}
```

DECRYPT RETURNS A PROMISE

ADDUSER EXPECTS A STRING

LOG EXPECTS A SMALL OBJECT OR A STRING



```
on: [Function: readingMore],
_paused: false,
read: [Function: decoder],
_encoding: null },
_readable: true,
domain: null,
_events:
{ end: [Function: drain],
finish: [Function: onSocketFinish],
_socketEnd: [Function: onSocketEnd],
drain: [Function: timeout],
timeout: [Function: socketOnTimeout],
data: [Function: bound socketOnData],
error: [Function: socketOnError],
close: [Function: close],
resume: [Function: onSocketResume],
pause: [Function: onSocketPause] },
_eventsCount: 10,
_maxListeners: undefined,
_writableState:
WritableState {
objectMode: false,
highWaterMark: 16384,
finalCalled: false,
needDrain: false,
ending: false,
ended: false,
finished: false,
destroyed: false,
decodeStrings: false,
defaultEncoding: 'utf8',
length: 0,
writing: false,
corked: 0,
sync: true,
bufferProcessing: false,
onwrite: [Function: bound onwrite],
writecb: null,
writelen: 0,
bufferedRequest: null,
lastBufferedRequest: null,
pendingcb: 0,
prefinished: false,
errorEmitted: false,
bufferedRequestCount: 0,
corkedRequestsFree: [Object] },
writable: true,
allowHalfOpen: true,
_bytesDispatched: 0,
_sockname: null,
_pendingData: null,
_pendingEncoding: '',
server:
Server {
domain: null,
_events: [Object],
_eventsCount: 2,
_maxListeners: undefined,
_connections: 2,
_handle: [Object],
_usingSlaves: false,
_slaves: []},
```

```
_idleNext:  
Socket {  
  connecting: false,  
  _hadError: false,  
  _handle: [Object],  
  _parent: null,  
  _host: null  
  _readableState: [Object],  
  readable: true,  
  domain: null,  
  _events: [Object],  
  _eventsCount: 10,  
  _maxListeners: undefined,  
  _writableState: [Object],  
  writable: true,  
  allowHalfOpen: true,  
  _bytesDispatched: 0,  
  _sockname: null,  
  _pendingData: null,  
  _pendingEncoding: '',  
  server: [Object],  
  _server: [Object],  
  _idleTimeout: 120000,  
  _idleNext: [Object],  
  _idlePrev: [Circular],  
  _idleStart: 257473,  
  _destroyed: false,  
  parser: [Object],  
  on: [Function: socketOnWrap]  
  _paused: false,  
  read: [Function],  
  _consuming: true,  
  [Symbol(asyncId)]: 14,  
  [Symbol(bytesRead)]: 0,  
  [Symbol(asyncId)]: 15,  
  [Symbol(triggerAsyncId)]: 6  
_idlePrev:  
TimersList {  
  _idleNext: [Circular],  
  _idlePrev: [Object],  
  _timer: [Object],  
  _unrefed: true,  
  msecs: 120000,  
  nextTick: false },  
_idleStart: 257482,  
_destroyed: false,  
parser:  
HTTPParser {  
  '0': [Function: parserOnHead],  
  '1': [Function: parserOnHead],  
  '2': [Function: parserOnBody],  
  '3': [Function: parserOnMessage],  
  '4': [Function: bound onParse],  
  _headers: [],  
  _url: '',  
  _consumed: true,  
  socket: [Circular],  
  incoming: [Object],  
  outgoing: null,  
  maxHeaderPairs: 2000,
```

```
onwrite: [Function: bound onwrite],
writecb: null,
writelen: 0,
bufferedRequest: null,
lastBufferedRequest: null,
pendingcb: 0,
prefinished: false,
errorEmitted: false,
bufferedRequestCount: 0,
corkedRequestsFree: [Object]
writable: true,
allowHalfOpen: true,
_bytesDispatched: 0,
_sockname: null,
_pendingData: null,
_pendingEncoding: '',
server:
Server {
  domain: null,
  _events: [Object],
  _eventsCount: 2,
  _maxListeners: undefined,
  _connections: 2,
  _handle: [Object],
  _usingSlaves: false
}
_server:
Server {
  domain: null,
  _events: [Object],
  _eventsCount: 2,
  _maxListeners: undefined,
  _connections: 2,
  _handle: [Object],
  _usingSlaves: false
}
  _unref: false,
  allowHalfOpen: true,
  pauseOnConnect: false,
  httpAllowHalfOpen: false,
  timeout: 120000,
  keepAliveTimeout: 5000,
  _pendingResponseData: 0,
  maxHeadersCount: null,
  _connectionKey: '6::::3000',
  [Symbol(asyncId)]: 6 },
server:
Server {
  domain: null,
  _events: [Object],
  _eventsCount: 2,
  _maxListeners: undefined,
  _connections: 2,
  _handle: [Object],
  _usingSlaves: false
}


```

server: [Object],
_server: [Object],
_idleTimeout: 120000,
_idleNext: [Object],
_idlePrev: [Circular],
_idleStart: 257473,
destroyed: false
}
```


```

```
decoder: null,
encoding: null },
domain: null,
_events: [Object],
_eventsCount: 10,
_maxListeners: undefined,
_writableState: [Object],
writable: true,
allowHalfOpen: true,
_bytesDispatched: 0,
_sockname: null,
_pendingData: null,
_pendingEncoding: '',
server: [Object],
_server: [Object],
_idleTimeout: 120000,
_idleNext: [Object],
_idlePrev: [Object],
_idleStart: 257482,
_destroyed: false,
parser: [Object],
on: [Function: socketOnWrap],
_paused: false,
read: [Function],
_consumming: true,
_httpMessage: [Circular],
[Symbol(asyncId)]: 8,
[Symbol(bytesRead)]: 0,
[Symbol(asyncId)]: 10,
[Symbol(triggerAsyncId)]: 6 },
httpVersionMajor: 1,
httpVersionMinor: 1,
httpVersion: '1.1',
complete: true,
headers: { host: 'localhost:3000',
connection: 'keep-alive',
'cache-control': 'max-age=0',
'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36',
'upgrade-insecure-requests': '1',
accept: 'text/html,application/xhtml+xml,application/xml,application/xml;q=0.9,image/webp,image/*/*;q=0.8,application/signed-exchange;v=b3',
'accept-encoding': 'gzip, deflate, br',
'accept-language': 'en-US,en;q=0.9',
cookie: '44907=; connect.sid=s%3At4Lwt9VNee: [Object]'},
'if-none-match': 'W/"4-7JiQZgN440hPd04Ung',
rawHeaders: [
'Host',
'localhost:3000',
'Connection',
'keep-alive',
'Cache-Control',
'max-age=0',
'User-Agent',
'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36',
'Upgrade-Insecure-Requests',
'1',
'Accept',
'text/html,application/xhtml+xml,application/xml,application/xml;q=0.9,image/webp,image/*/*;q=0.8,application/signed-exchange;v=b3',
'Accept-Encoding',
'gzip, deflate, br',
'Accept-Language',
'en-US,en;q=0.9',
'Cookie']
```

```
"ServerResponse {
  domain: null,
  _events: { finish: [Function: bound resOnFinish] },
  _eventsCount: 1,
  _maxListeners: undefined,
  output: [],
  outputEncodings: [],
  outputCallbacks: [],
  outputSize: 0,
  writable: true,
  _last: false,
  upgrading: false,
  chunkedEncoding: false,
  shouldKeepAlive: true,
  useChunkedEncodingByDefault: true,
  sendDate: true,
  _removedConnection: false,
  _removedContLen: false,
  _removedTE: false,
  _contentLength: null,
  _hasBody: true,
  _trailer: '',
  finished: false,
  _headerSent: false,
  socket:
    Socket {
      connecting: false,
      _hadError: false,
      _handle:
        TCP {
          reading: true,
          owner: [Circular],
          onread: [Function: onread],
          onconnection: null,
          writeQueueSize: 0,
          _consumed: true },
      _parent: null,
      _host: null,
      _readableState:
        ReadableState {
          objectMode: false,
          highWaterMark: 16384,
          buffer: [Object],
          length: 0,
          pipes: null,
          pipesCount: 0,
          flowing: true,
          ended: false,
          endEmitted: false,
          reading: true,
          sync: false,
          needReadable: true,
          emittedReadable: false,
          readableListening: false,
          resumeScheduled: false,
          destroyed: false,
          defaultEncoding: 'utf8',
          awaitDrain: 0,
          readingMore: false,
          decoder: null
        }
    }
},
```

LESSONS LEARNED

- ▶ We need better **static analysis**
- ▶ We need better **debugging tools**
- ▶ We need a **consistent way to handle errors**
- ▶ We need to **better understand our logging & monitoring**



STATIC ANALYSIS WITH

FLOWTYPE & ESLINT

TYPE CHECKING COULD HAVE CAUGHT
THAT BUG WITH 2-LINES OF CODE

PREVENTING BUGS WITH TYPES

```
async function optInToExperiment(senderId) {  
  let encrtypedId: string = decrypt(senderId)  
  expService.addUser(encrtypedId, function(err, res) {  
    if (err) return logger.log(res)  
    // success  
  })  
}
```

PREVENTING BUGS WITH TYPES

```
function decrypt(encryptedId: string): Promise<string> {  
    return Service.decryptValue(encryptedId)  
}
```

FLOW WON'T LET THAT SLIDE

```
Error ----- test.js:7:28

Cannot assign decrypt(...) to encryptedId because Promise [1] is incompatible with string [2].

[1] 2| function decrypt(encryptedId): Promise<string> {
[2] 3|   return Service.decryptValue(encryptedId).then(id => id)
[4] 4|
[5] 5|
[6] 6| async function optInToExperiment(senderId) {
[7] 7|   let encryptedId: string = decrypt(senderId)
[8] 8|
[9] 9| }
```

WHY ADD TYPES TO YOUR JS?

- ▶ Prevents large % of bugs
- ▶ Helps surface architectural problems
- ▶ Both **Flow** and **TypeScript** are well maintained, high quality tools
- ▶ Both syntaxes are light-weight and easy to use
- ▶ Both allow for gradual adoption

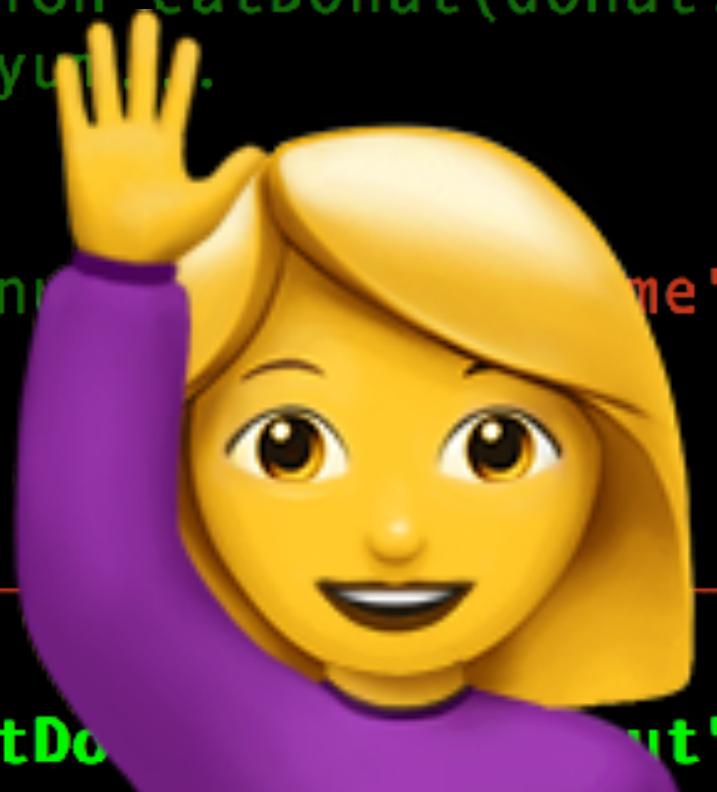
```
// @flow
type Donut = "maple" | "chocolate" | "glazed"
class Person {
  name: string;
  constructor(name) {
    this.name = name;
  }
  function eatDonut(donut: Donut, person: Person) {
    // yum...
  }
}
eatDonut(new Person("me"), "donut")
```

Error

donut.js:13:10

Cannot call **eatDonut** with **new Person(...)** bound to **donut** because **Person** [1] is incompatible with **string enum** [2].

```
[2] 9| function eatDonut(donut: Donut, person: Person) {  
10|   // yum...  
11| }  
[1] 13| eatDonut(new Person("me"), "donut")  
14|
```



Error

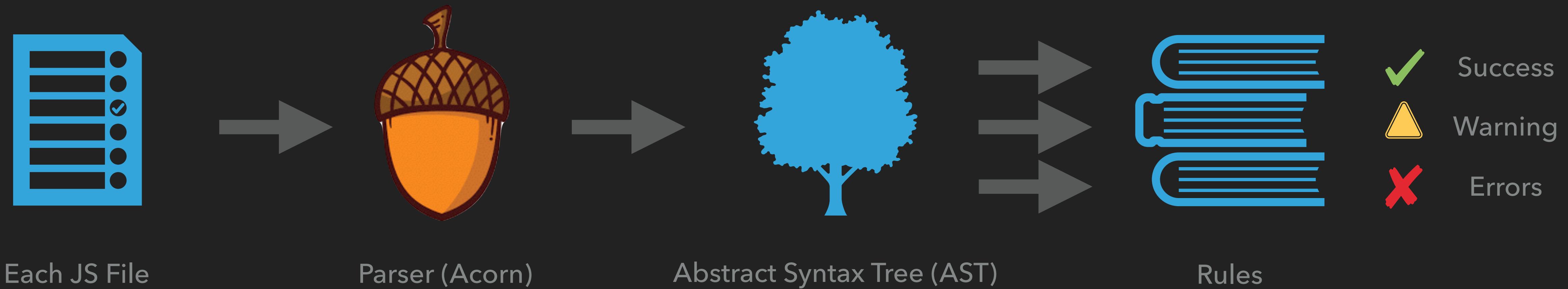
donut.js:13:28

Cannot call **eatDonut** with **"donut"** bound to **person** because **string** [1] is incompatible with **Person** [2].

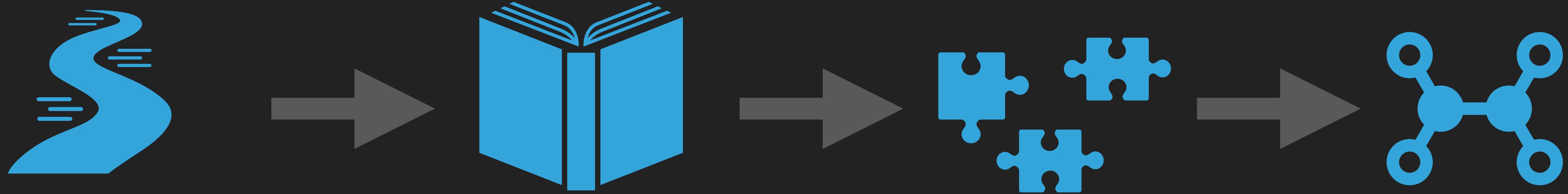
```
[2] 9| function eatDonut(donut: Donut, person: Person) {  
10|   // yum...  
11| }  
[1] 13| eatDonut(new Person("me"), "donut")  
14|
```

Type systems have a lot in common with linters

Architecture of JavaScript Linter (ESLint)



Architecture of JavaScript Type System (Flow)



**COULD A LINTER HAVE HELPED
US WITH OUR MYSTERY BUG?**

STATIC ANALYSIS

```
function payRequest(req, res) {  
    // all the important pay request logic...  
  
    // send back the request id  
    res.send(requestId)  
  
    // ... sending to a known sender...  
    if (senderId) {  
        await startInExperiment(senderId)  
    }  
}
```

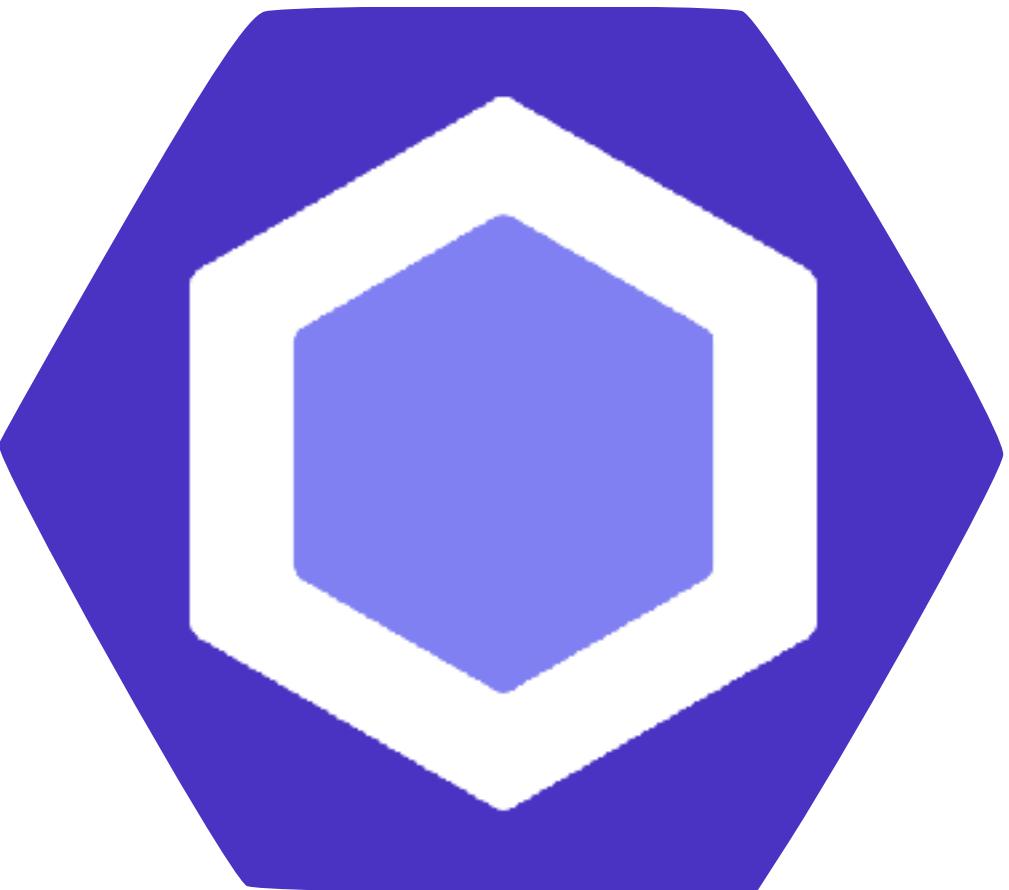
STATIC ANALYSIS

```
async function optInToExperiment(senderId) {  
  let encrtypedId = decrypt(senderId)  
  expService.addUser(encrtypedId, function(err, res) {  
    if (err) return logger.log((err))  
    // success  
  })  
}
```

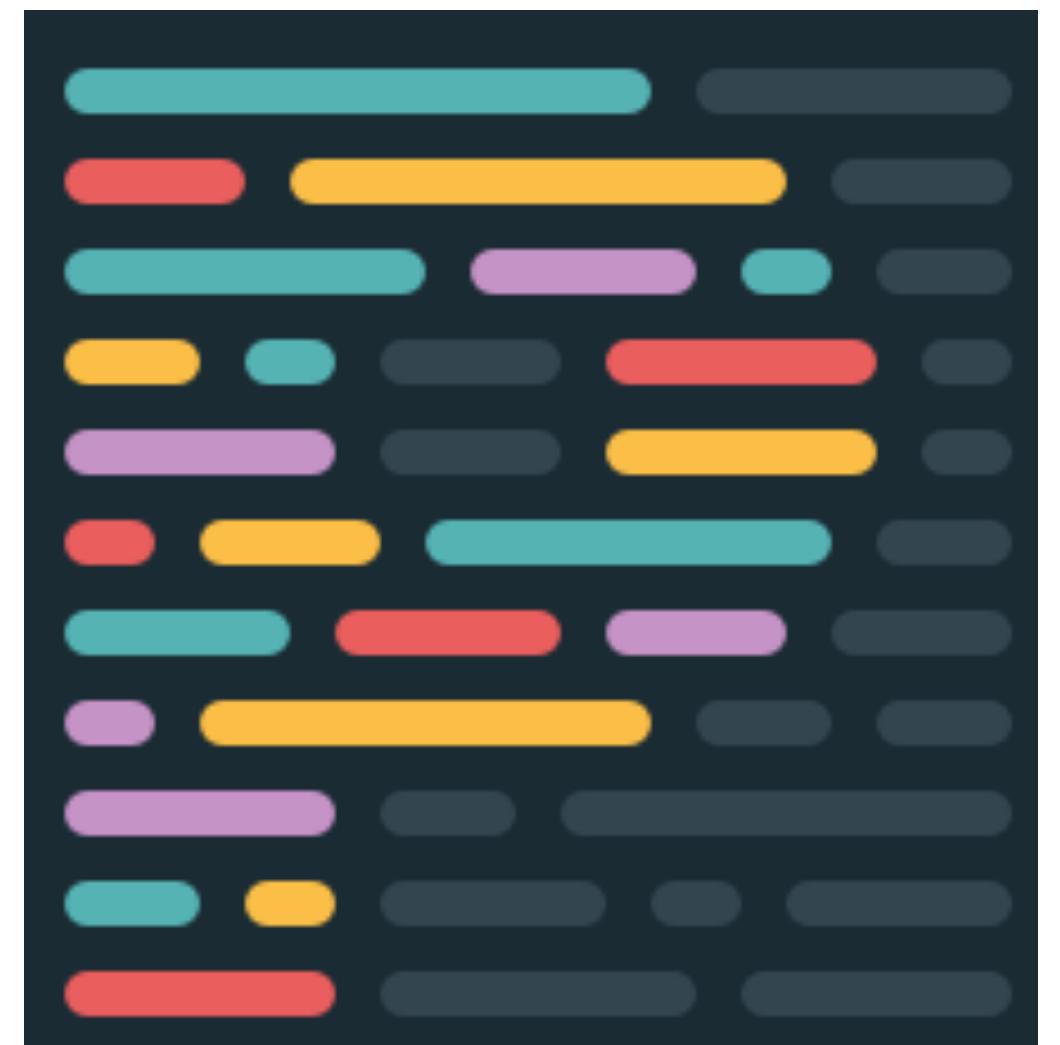
STATIC ANALYZERS AND FORMATTERS ARE PRETTY COOL



FlowType



ESLint

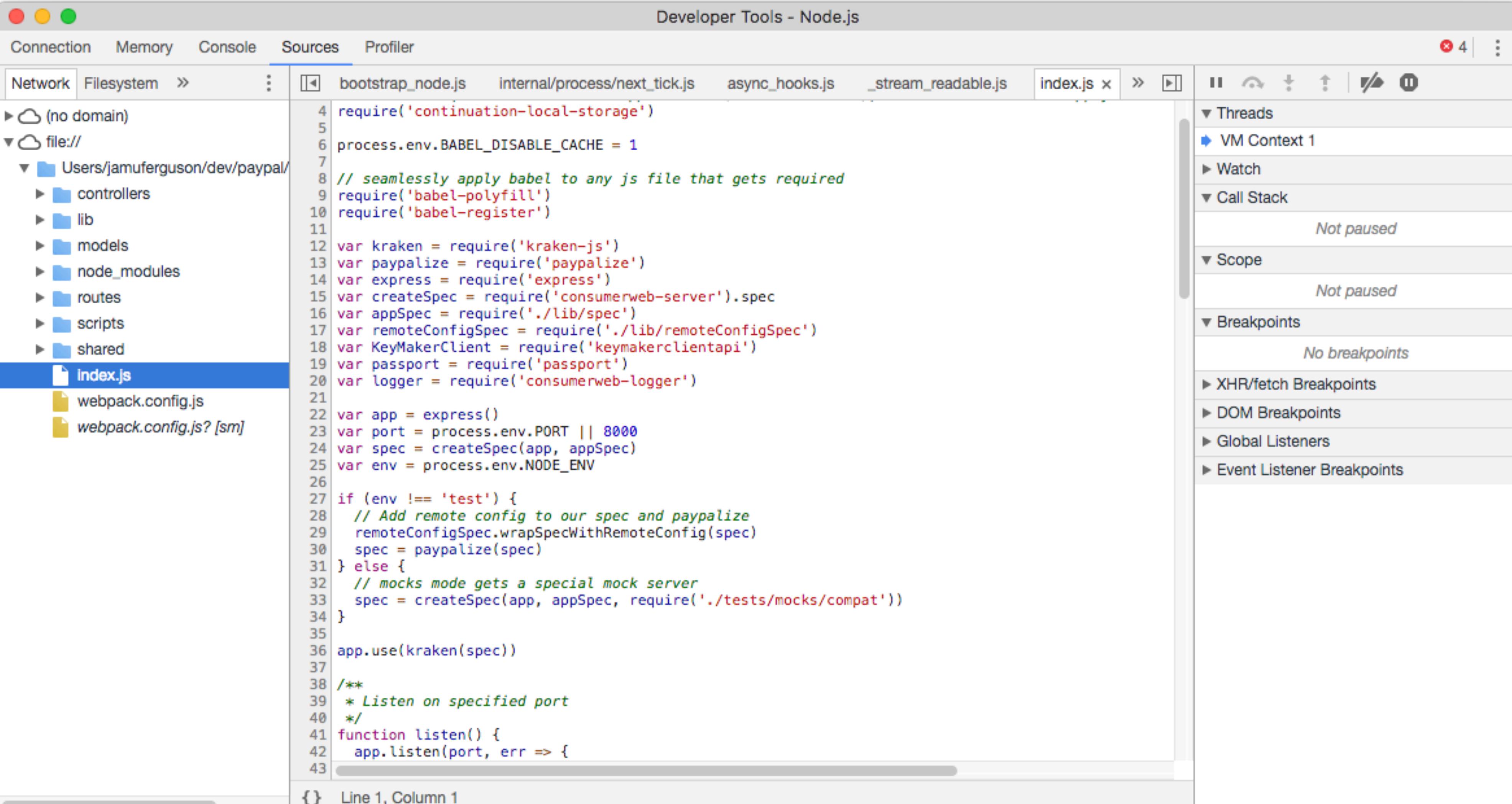


Prettier

DEBUGGING USING THE

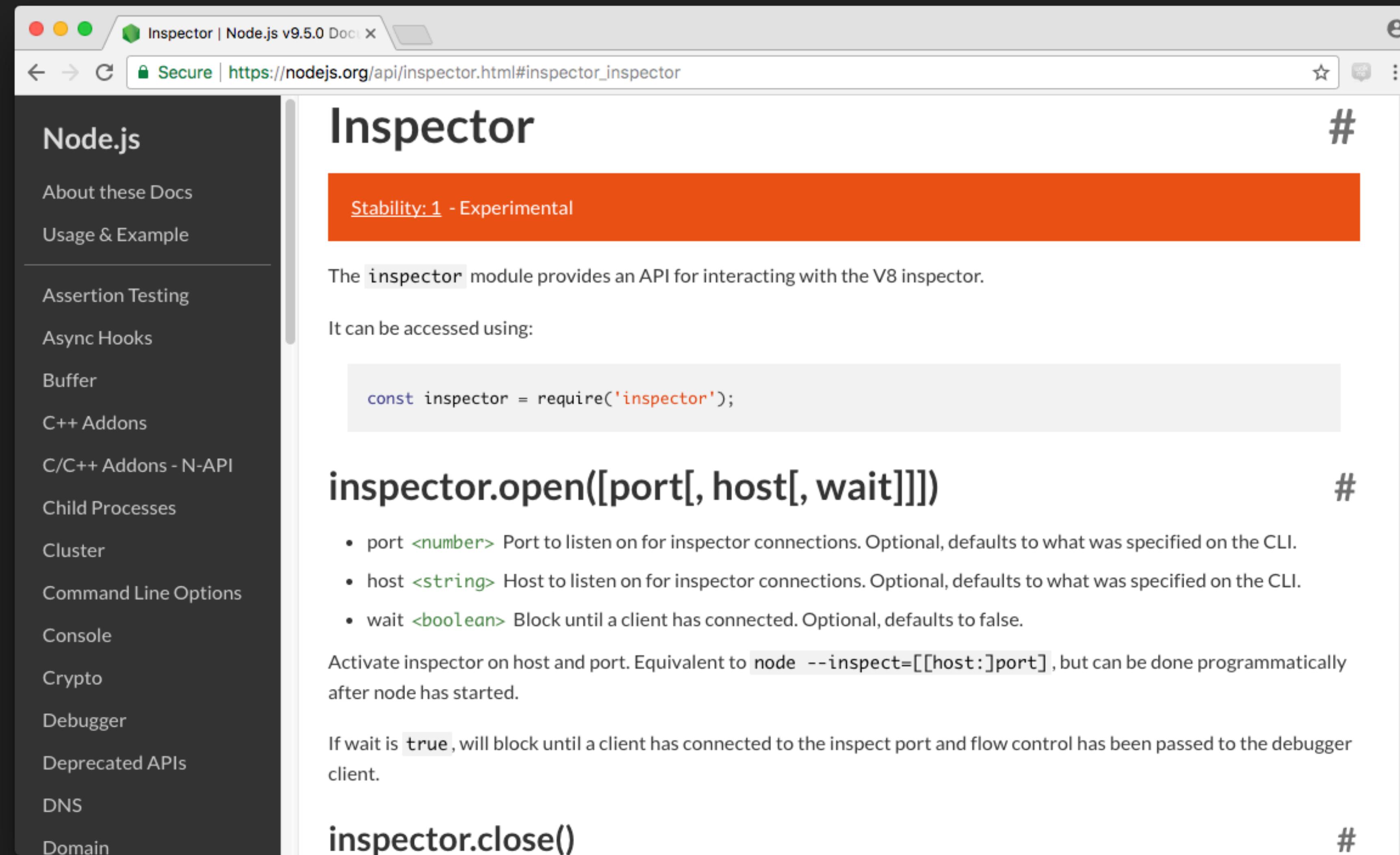
INSPECTOR MODULE

THE BUILT-IN INSPECTOR MODULE



The screenshot shows the Node.js Developer Tools interface with the "Sources" tab selected. The left sidebar displays a file tree under "file://". The "Index.js" file is selected and highlighted in blue. The main pane shows the source code for "index.js". The right sidebar contains various developer tools like "Threads", "VM Context 1", "Watch", "Call Stack", "Scope", "Breakpoints", "XHR/fetch Breakpoints", "DOM Breakpoints", "Global Listeners", and "Event Listener Breakpoints".

```
4 require('continuation-local-storage')
5
6 process.env.BABEL_DISABLE_CACHE = 1
7
8 // seamlessly apply babel to any js file that gets required
9 require('babel-polyfill')
10 require('babel-register')
11
12 var kraken = require('kraken-js')
13 var paypalize = require('paypalize')
14 var express = require('express')
15 var createSpec = require('consumerweb-server').spec
16 var appSpec = require('./lib/spec')
17 var remoteConfigSpec = require('./lib/remoteConfigSpec')
18 var KeyMakerClient = require('keymakerclientapi')
19 var passport = require('passport')
20 var logger = require('consumerweb-logger')
21
22 var app = express()
23 var port = process.env.PORT || 8000
24 var spec = createSpec(app, appSpec)
25 var env = process.env.NODE_ENV
26
27 if (env !== 'test') {
28   // Add remote config to our spec and paypalize
29   remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
30   spec = paypalize(spec)
31 } else {
32   // mocks mode gets a special mock server
33   spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
34 }
35
36 app.use(kraken(spec))
37
38 /**
39 * Listen on specified port
40 */
41 function listen() {
42   app.listen(port, err => {
43     {} Line 1, Column 1
44   })
45 }
```



The screenshot shows a web browser displaying the Node.js v9.5.0 Documentation for the `Inspector` module. The URL is https://nodejs.org/api/inspector.html#inspector_inspector. The page has a dark theme with orange highlights. On the left is a sidebar with links to various Node.js modules. The main content area has a large title `Inspector` and a stability banner indicating it is experimental. It describes the `inspector` module and provides a code snippet to access it. Below this is a section for the `inspector.open()` method, which takes optional parameters for port, host, and wait. It also describes the `inspector.close()` method.

Node.js

Inspector

Stability: 1 - Experimental

The `inspector` module provides an API for interacting with the V8 inspector.

It can be accessed using:

```
const inspector = require('inspector');
```

inspector.open([port[, host[, wait]])]

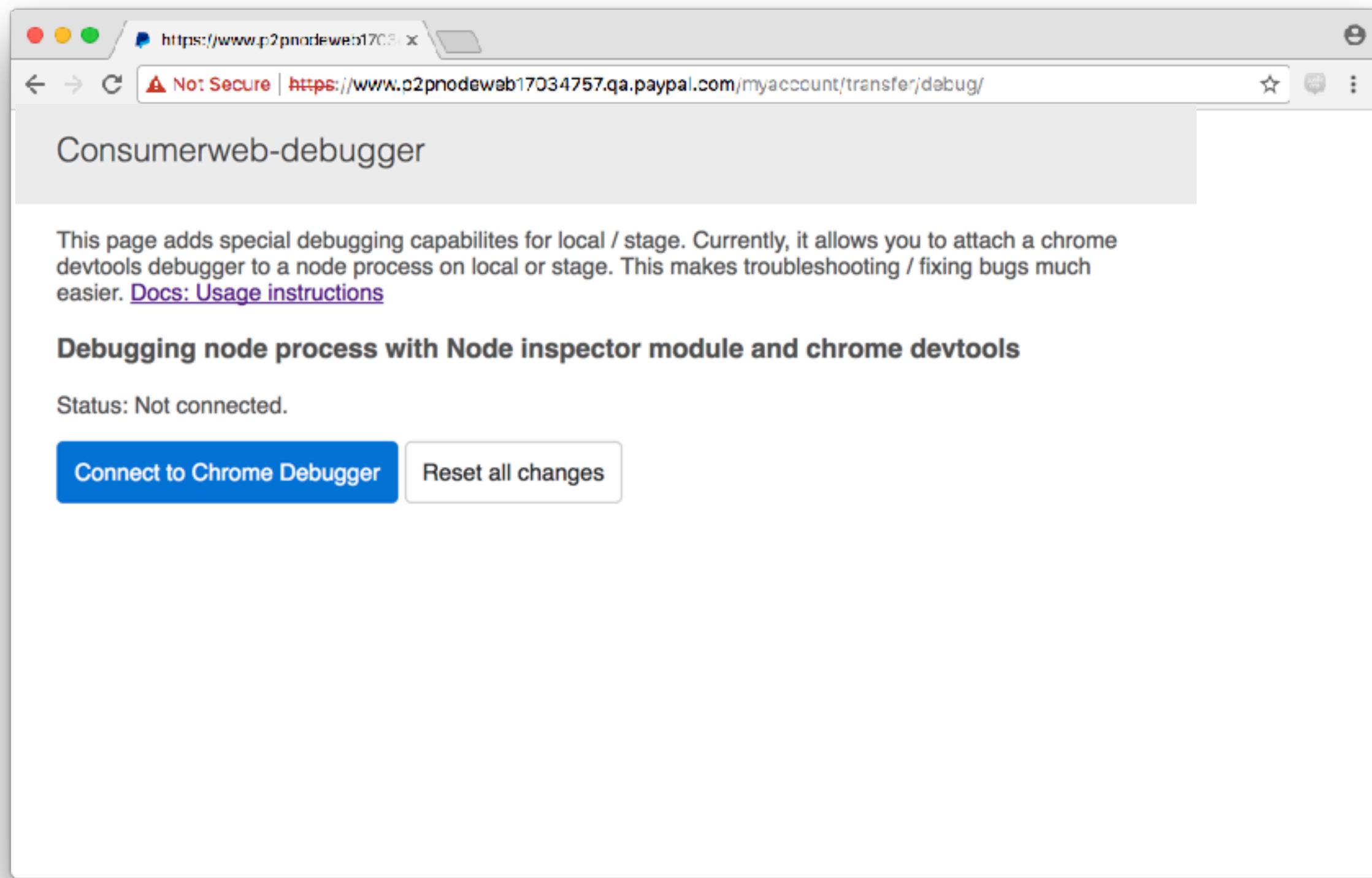
- `port <number>` Port to listen on for inspector connections. Optional, defaults to what was specified on the CLI.
- `host <string>` Host to listen on for inspector connections. Optional, defaults to what was specified on the CLI.
- `wait <boolean>` Block until a client has connected. Optional, defaults to false.

Activate inspector on host and port. Equivalent to `node --inspect=[[:port]]`, but can be done programmatically after node has started.

If `wait` is `true`, will block until a client has connected to the inspect port and flow control has been passed to the debugger client.

inspector.close()

SETTING UP A DEBUG MODE



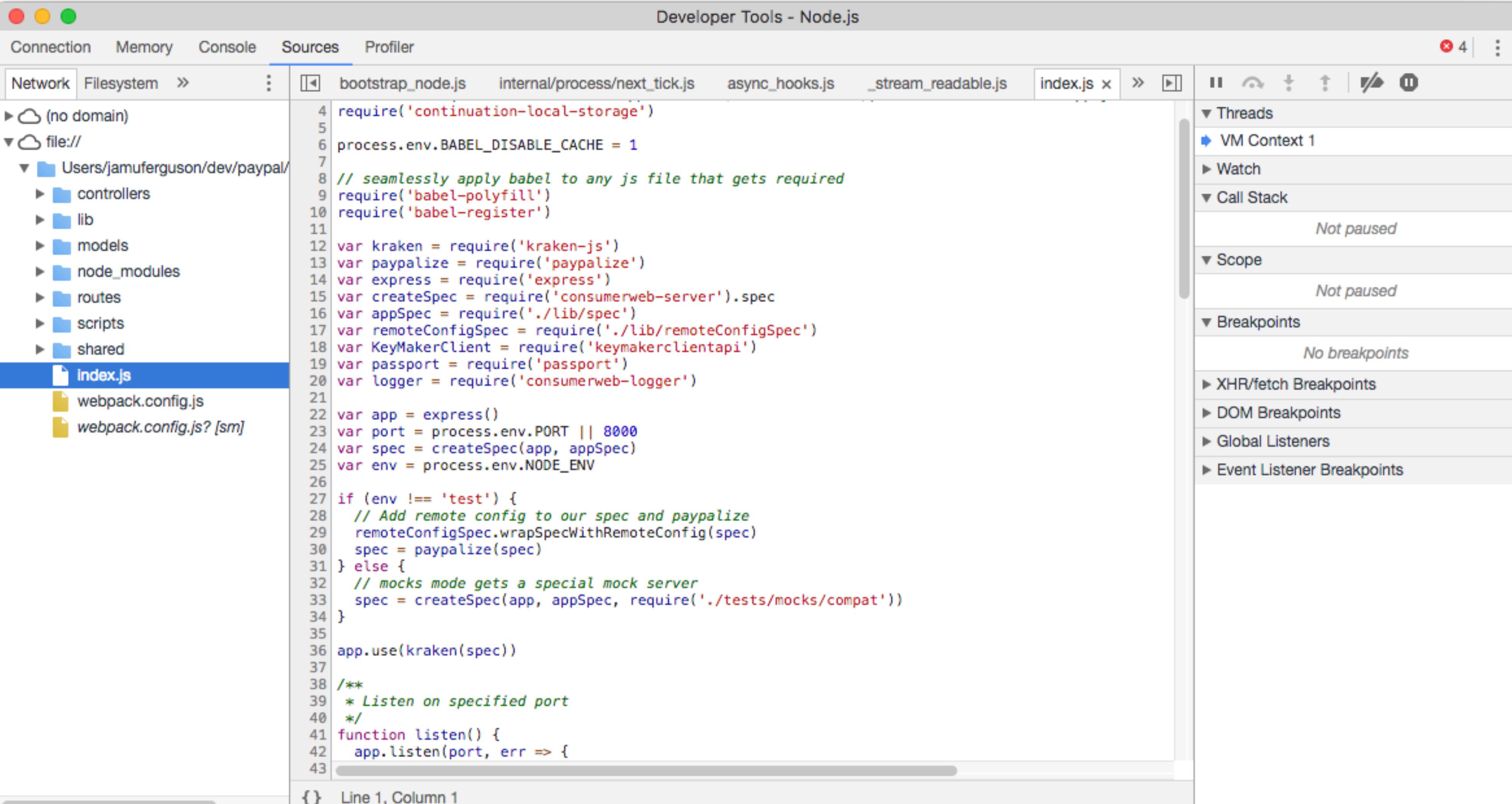
TURN IT ON

```
var inspector = require('inspector')  
inspector.open()
```

TURN IT OFF

```
inspector.close()
```

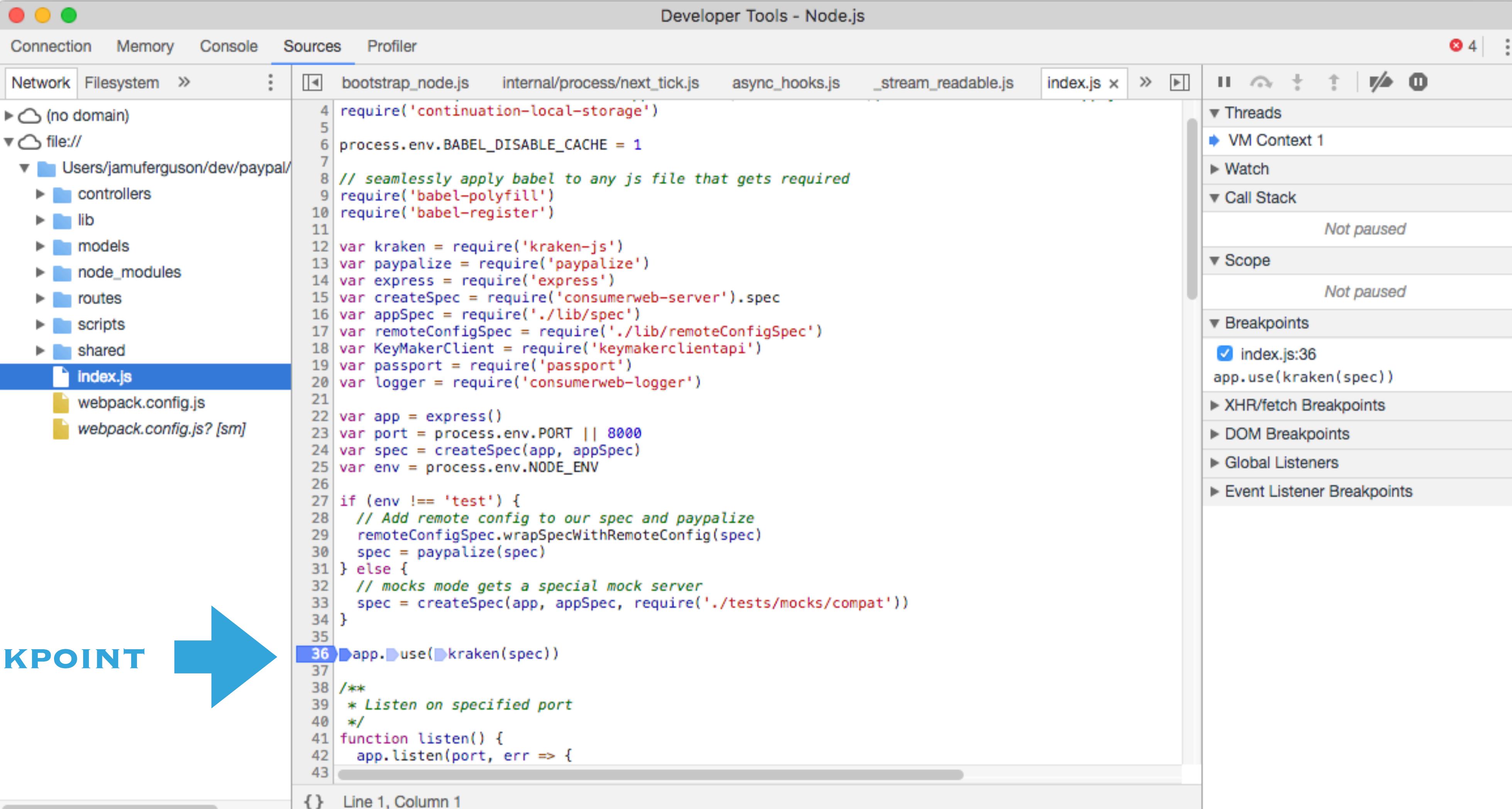
THE BUILT-IN INSPECTOR MODULE



The screenshot shows the Node.js Developer Tools interface with the "Sources" tab selected. The left sidebar displays a file tree under "file://". The "Index.js" file is selected and highlighted in blue. The main pane shows the source code for "index.js". The right sidebar contains various developer tools like "Threads", "VM Context 1", "Watch", "Call Stack", "Scope", "Breakpoints", "XHR/fetch Breakpoints", "DOM Breakpoints", "Global Listeners", and "Event Listener Breakpoints".

```
4 require('continuation-local-storage')
5
6 process.env.BABEL_DISABLE_CACHE = 1
7
8 // seamlessly apply babel to any js file that gets required
9 require('babel-polyfill')
10 require('babel-register')
11
12 var kraken = require('kraken-js')
13 var paypalize = require('paypalize')
14 var express = require('express')
15 var createSpec = require('consumerweb-server').spec
16 var appSpec = require('./lib/spec')
17 var remoteConfigSpec = require('./lib/remoteConfigSpec')
18 var KeyMakerClient = require('keymakerclientapi')
19 var passport = require('passport')
20 var logger = require('consumerweb-logger')
21
22 var app = express()
23 var port = process.env.PORT || 8000
24 var spec = createSpec(app, appSpec)
25 var env = process.env.NODE_ENV
26
27 if (env !== 'test') {
28   // Add remote config to our spec and paypalize
29   remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
30   spec = paypalize(spec)
31 } else {
32   // mocks mode gets a special mock server
33   spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
34 }
35
36 app.use(kraken(spec))
37
38 /**
39 * Listen on specified port
40 */
41 function listen() {
42   app.listen(port, err => {
43     {} Line 1, Column 1
44   })
45 }
```

THE BUILT-IN INSPECTOR MODULE



The screenshot shows the Node.js Developer Tools interface with the "Sources" tab selected. On the left, the file structure is displayed, showing a local directory "Users/jamuferguson/dev/paypal/" containing "controllers", "lib", "models", "node_modules", "routes", "scripts", "shared", and "Index.js". "Index.js" is currently selected and highlighted in blue. In the main code editor area, the "index.js" file is open, showing its contents. A large blue arrow points from the text "SET BREAKPOINT" to the line number 36 in the code, which is annotated with a small blue box. The code at line 36 is: `36 app.use(kraken(spec))`. The right panel contains various developer tools like Threads, VM Context 1, Watch, Call Stack, Scope, Breakpoints, XHR/fetch Breakpoints, DOM Breakpoints, Global Listeners, and Event Listener Breakpoints. The "Breakpoints" section shows a checked checkbox next to "index.js:36", indicating it is active.

```
4 require('continuation-local-storage')
5
6 process.env.BABEL_DISABLE_CACHE = 1
7
8 // seamlessly apply babel to any js file that gets required
9 require('babel-polyfill')
10 require('babel-register')
11
12 var kraken = require('kraken-js')
13 var paypalize = require('paypalize')
14 var express = require('express')
15 var createSpec = require('consumerweb-server').spec
16 var appSpec = require('./lib/spec')
17 var remoteConfigSpec = require('./lib/remoteConfigSpec')
18 var KeyMakerClient = require('keymakerclientapi')
19 var passport = require('passport')
20 var logger = require('consumerweb-logger')
21
22 var app = express()
23 var port = process.env.PORT || 8000
24 var spec = createSpec(app, appSpec)
25 var env = process.env.NODE_ENV
26
27 if (env !== 'test') {
28   // Add remote config to our spec and paypalize
29   remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
30   spec = paypalize(spec)
31 } else {
32   // mocks mode gets a special mock server
33   spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
34 }
35
36 app.use(kraken(spec))
37
38 /**
39 * Listen on specified port
40 */
41 function listen() {
42   app.listen(port, err => {
43     if (err) {
44       console.error(`Error listening on port ${port}: ${err.message}`)
45     } else {
46       console.log(`Listening on port ${port}`)
47     }
48   })
49 }
50
51 listen()
```

THE BUILT-IN INSPECTOR MODULE

Developer Tools - Node.js

PAUSE ON UNCAUGHT EXCEPTIONS 

Pause on caught exceptions

Connection Memory Console Sources Profiler

Network Filesystem > bootstrap_node.js internal/process/next_tick.js async_hooks.js _stream_readable.js index.js >

File System Tree:

- > Cloud (no domain)
- > file://
 - > Users/jamuferguson/dev/paypal/
 - controllers
 - lib
 - models
 - node_modules
 - routes
 - scripts
 - shared
 - index.js
 - webpack.config.js
 - webpack.config.js? [sm]

Code Editor (index.js):

```
4 require('continuation-local-storage')
5
6 process.env.BABEL_DISABLE_CACHE = 1
7
8 // seamlessly apply babel to any js file that gets required
9 require('babel-polyfill')
10 require('babel-register')
11
12 var kraken = require('kraken-js')
13 var paypalize = require('paypalize')
14 var express = require('express')
15 var createSpec = require('consumerweb-server').spec
16 var appSpec = require('./lib/spec')
17 var remoteConfigSpec = require('./lib/remoteConfigSpec')
18 var KeyMakerClient = require('keymakerclientapi')
19 var passport = require('passport')
20 var logger = require('consumerweb-logger')
21
22 var app = express()
23 var port = process.env.PORT || 8000
24 var spec = createSpec(app, appSpec)
25 var env = process.env.NODE_ENV
26
27 if (env !== 'test') {
28   // Add remote config to our spec and paypalize
29   remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
30   spec = paypalize(spec)
31 } else {
32   // mocks mode gets a special mock server
33   spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
34 }
35
36 app.use(kraken(spec))
37
38 /**
39 * Listen on specified port
40 */
41 function listen() {
42   app.listen(port, err => {
43     if (err) {
44       console.error(`Error listening on port ${port}: ${err.message}`)
45     } else {
46       console.log(`Listening on port ${port}`)
47     }
48   })
49 }
50
51 listen()
```

Breakpoints:

- index.js:36 (checked)
- app.use(kraken(spec))
- XHR/fetch Breakpoints
- DOM Breakpoints
- Global Listeners
- Event Listener Breakpoints

Threads: VM Context 1

Watch

Call Stack: Not paused

Scope: Not paused

Event Listener Breakpoints

Line 1, Column 1

THE BUILT-IN INSPECTOR MODULE

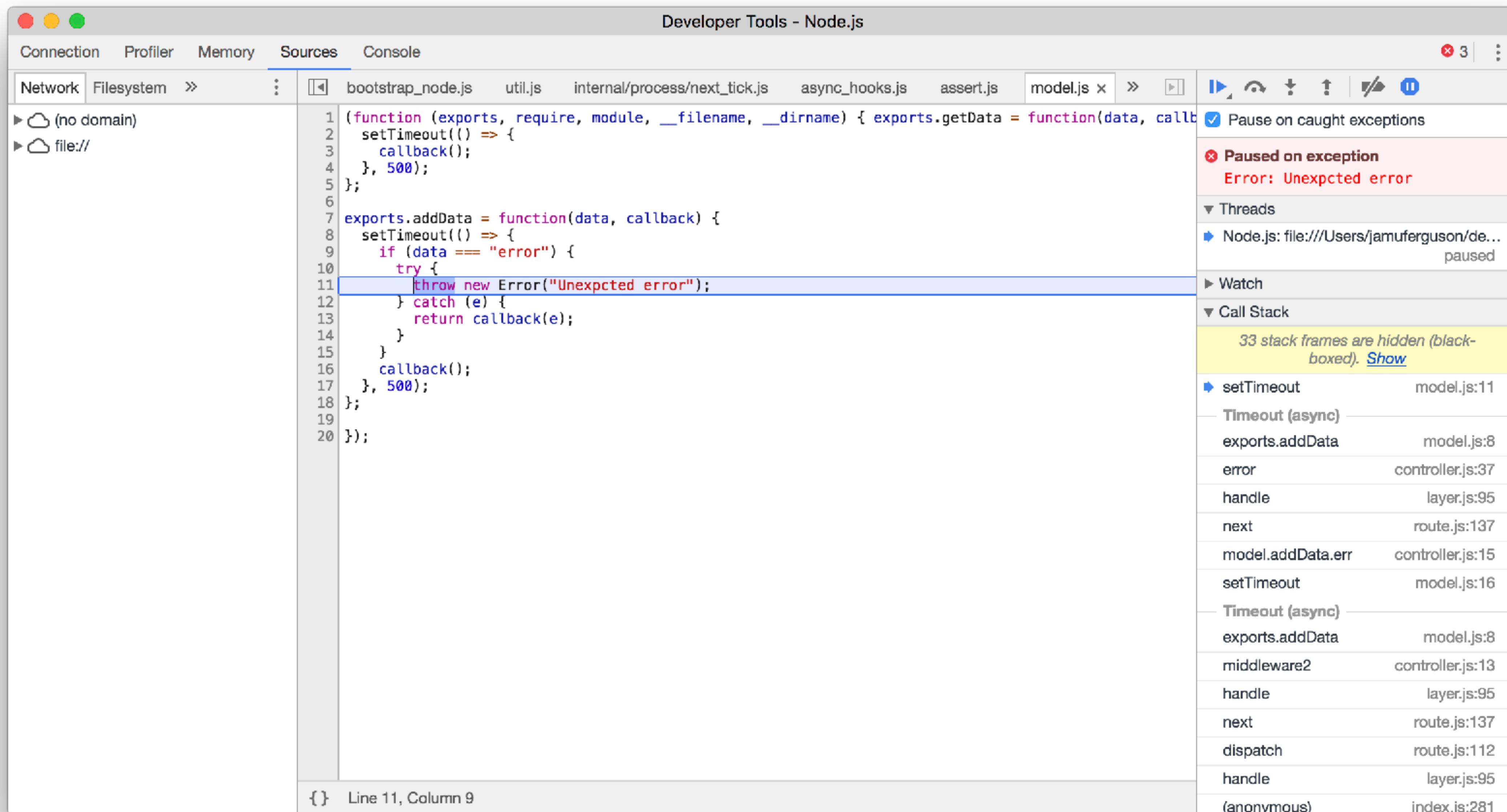
PAUSE ON CAUGHT EXCEPTIONS

The screenshot shows the Node.js Developer Tools interface. The left pane displays a file tree under 'Sources'. The 'index.js' file is selected and shown in the main editor area. A blue arrow points from the title 'PAUSE ON CAUGHT EXCEPTIONS' to the 'Breakpoints' section on the right. In the 'Breakpoints' section, there is a checked checkbox for 'index.js:36 app.use(kraken(spec))'. The right pane also includes sections for 'Threads', 'Watch', 'Call Stack', 'Scope', and various 'Event Listener Breakpoints'.

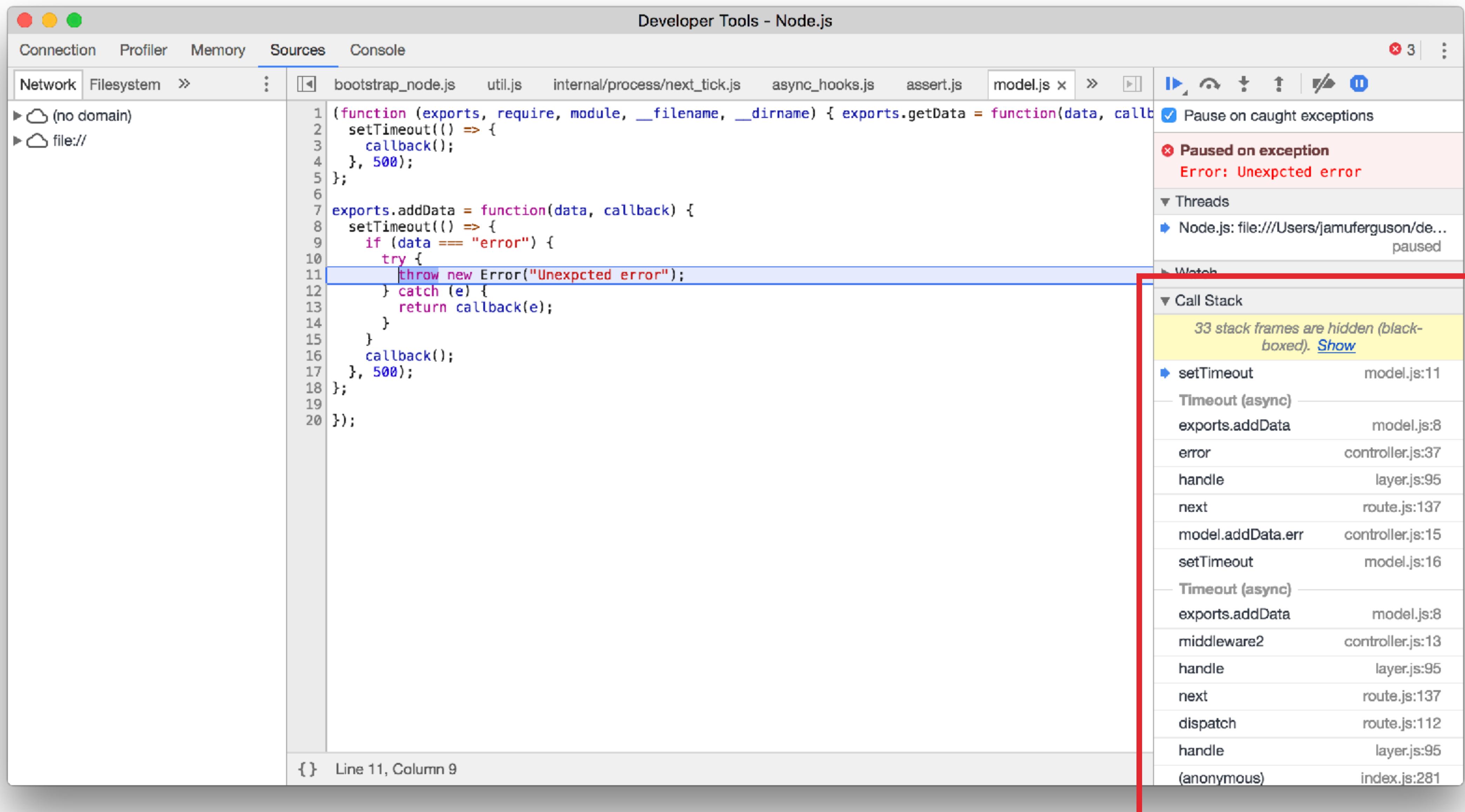
```
4 require('continuation-local-storage')
5
6 process.env.BABEL_DISABLE_CACHE = 1
7
8 // seamlessly apply babel to any js file that gets required
9 require('babel-polyfill')
10 require('babel-register')
11
12 var kraken = require('kraken-js')
13 var paypalize = require('paypalize')
14 var express = require('express')
15 var createSpec = require('consumerweb-server').spec
16 var appSpec = require('./lib/spec')
17 var remoteConfigSpec = require('./lib/remoteConfigSpec')
18 var KeyMakerClient = require('keymakerclientapi')
19 var passport = require('passport')
20 var logger = require('consumerweb-logger')
21
22 var app = express()
23 var port = process.env.PORT || 8000
24 var spec = createSpec(app, appSpec)
25 var env = process.env.NODE_ENV
26
27 if (env !== 'test') {
28   // Add remote config to our spec and paypalize
29   remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
30   spec = paypalize(spec)
31 } else {
32   // mocks mode gets a special mock server
33   spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
34 }
35
36 app.use(kraken(spec))
37
38 /**
39 * Listen on specified port
40 */
41 function listen() {
42   app.listen(port, err => {
43     {} Line 1, Column 1

```

THE BUILT-IN INSPECTOR MODULE



THE BUILT-IN INSPECTOR MODULE



THE BUILT-IN INSPECTOR MODULE

THE BUILT-IN INSPECTOR MODULE

Developer Tools - Node.js

Connection Profiler Memory Sources Console

Network » : bootstrap_node.js internal/process/next_tick.js async_hooks.js assert.js vm.js model.js x »

▶ Cloud (no domain) 1 (function (exports, require, module, __filename, __dirname) { var { MissingParamError } = require("./e

▼ file:// 2

 Users/jamuferguson/d 3 exports.getData = function(data, callback) {

 setTimeout(() => {

 callback();

 }, 500);

 };

 exports.addData = function(data, callback) {

 setTimeout(() => {

 if (data === "special") {

 try {

 throw new MissingParamError("You are missing the name param");

 } catch (e) {

});

✓ Pause on caught exceptions

✖ Paused on exception
Error: Unexpcted error

Threads

Node.js: file:///Users/jamufer... paused

Watch

Call Stack

ontimeout timers.js:475

tryOnTimeout timers.js:310

listOnTimeout timers.js:270

Timeout (async)

init

internal/inspec...ync_hook.js:1

emitInitNative

async_hooks.js:472

emitInitScript

async_hooks.js:388

Timeout timers.js:579

createSingleTimeout

timers.js:459

setTimeout timers.js:443

exports.addData

model.js:10

error controller.js:48

{ } Line 20, Column 8



Don't try this in production

THE BUILT-IN INSPECTOR MODULE

The screenshot shows a web browser window displaying the Node.js v9.5.0 Documentation for the `inspector.Session` class. The URL in the address bar is https://nodejs.org/api/inspector.html#inspector_class_inspector_session.

Node.js sidebar:

- About these Docs
- Usage & Example
- Assertion Testing
- Async Hooks
- Buffer
- C++ Addons
- C/C++ Addons - N-API
- Child Processes
- Cluster
- Command Line Options
- Console
- Crypto
- Debugger
- Deprecated APIs
- DNS
- Domain
- ECMAScript Modules

Class: inspector.Session

The `inspector.Session` is used for dispatching messages to the V8 inspector back-end and receiving message responses and notifications.

Constructor: new inspector.Session()

Added in: v8.0.0

Create a new instance of the `inspector.Session` class. The inspector session needs to be connected through `session.connect()` before the messages can be dispatched to the inspector backend.

`inspector.Session` is an `EventEmitter` with the following events:

Event: 'inspectorNotification'

Added in: v8.0.0

- `<Object>` The notification message object

Emitted when any notification from the V8 Inspector is received.

```
session.on('inspectorNotification', (message) => console.log(message.method));
// Debugger.paused
// Debugger.resumed
```

It is also possible to subscribe only to notifications with specific method:

THE BUILT-IN INSPECTOR MODULE

```
var session = new inspector.Session();

session.connect();

session.post("Runtime.enable");
session.post("Debugger.enable");

session.post("Debugger.setAsyncCallStackDepth", { maxDepth: 20 });
session.post("Debugger.setPauseOnExceptions", { state: "all" });

session.on('Debugger.paused', ({ params }) => {
  // params.callFrames
  // params.reason
  // params.asyncStackTrace
});
```

THE BUILT-IN INSPECTOR MODULE

The screenshot shows the Chrome DevTools Protocol Viewer interface. The left sidebar lists various protocol modules: Domains, DeviceOrientation, DOM, DOMDebugger, DOMSnapshot, DOMStorage, Emulation, HeadlessExperimental, HeapProfiler, IndexedDB, Input, Inspector, IO, and LayerTree. The main content area is titled "Chrome DevTools Protocol Viewer" and displays the "Inspector" module documentation. It includes sections for `Debugger.setPauseOnExceptions`, `Debugger.setReturnValue`, and `Debugger.setScriptSource`. The `Debugger.setScriptSource` section includes parameters for `scriptId`, `scriptSource`, and `dryRun`, and specifies a `callFrames` return object.

Version latest (tip-of-tree)

Chrome DevTools Protocol Viewer

Debugger.`setPauseOnExceptions`

Defines pause on exceptions state. Can be set to stop on all exceptions, uncaught exceptions or no exceptions. Initial pause on exceptions state is `none`.

PARAMETERS

state string
Pause on exceptions mode. none, uncaught, all

Debugger.`setReturnValue`

Changes return value in top frame. Available only at return break position. EXPERIMENTAL

PARAMETERS

newValue [Runtime.CallArgument](#)
New return value.

Debugger.`setScriptSource`

Edits JavaScript source live.

PARAMETERS

scriptId [Runtime.ScriptId](#)
Id of the script to edit.

scriptSource string
New content of the script.

dryRun boolean
If true the change will not actually be applied. Dry run may be used to get result description without actually modifying the code.

RETURN OBJECT

callFrames array [CallFrame](#)
New stack trace in case editing has happened while VM was stopped.

Secure | https://chromedevtools.github.io/devtools-protocol/tot/Debugger

<https://chromedevtools.github.io/devtools-protocol/>

THE BUILT-IN INSPECTOR MODULE

Default Node Error

```
Error: Unexpcted error
    at Timeout.setTimeout [as _onTimeout] (/Users/jamuferguson/dev/qcon/model.js:10:23)
    at ontimeout (timers.js:475:11)
    at tryOnTimeout (timers.js:310:5)
    at Timer.listOnTimeout (timers.js:270:5)
```

Inspector Based Error

```
Error: Unexpcted error
    at init (internal/inspector_async_hook.js:18:14)
    at emitInitNative (async_hooks.js:471:42)
    at emitInitScript (async_hooks.js:387:2)
    at Timeout (timers.js:578:4)
    at createSingleTimeout (timers.js:458:14)
    at setTimeout (timers.js:442:9)
    at exports.addData (~qcon/model.js:7:2)
    at error (~qcon/controller.js:36:8)
    at handle (~qcon/node_modules/express/lib/router/layer.js:94:4)
    at next (~qcon/node_modules/express/lib/router/route.js:136:12)
    at model.addData.err (~qcon/controller.js:14:4)
    at setTimeout (~qcon/model.js:15:4)
    at ontimeout (timers.js:474:10)
    at tryOnTimeout (timers.js:309:4)
    at listOnTimeout (timers.js:269:4)
```

**FIND A DEBUGGING APPROACH THAT
WORKS FOR YOU AND YOUR TEAM**

ERROR HANDLING USING

ASYNC/AWAIT

ERROR HANDLING WITH ASYNC/AWAIT

```
// here is an async function
async function getNumber() {
  return 4 // actually returns a Promise
}
```

```
// the same function using promises
function getNumber() {
  return Promise.resolve(4)
}
```

Errors thrown inside async
functions get converted into
rejected Promises



ERROR HANDLING WITH ASYNC/AWAIT

```
async function badNews() {  
    throw new Error('bad news')  
}  
  
function badNews() {  
    return Promise.reject(new Error('bad news'))  
}
```

Async Middleware Pattern

ERROR HANDLING WITH ASYNC/AWAIT

```
async function getUsers(req, res) {  
  let users = await getUsersForId(req.params.id)  
  res.json(users)  
}  
  
server.get('/users', getUsers)
```

THIS IS PRETTY NICE

ERROR HANDLING WITH ASYNC/AWAIT

ERRORS WILL BUBBLE UP



```
async function getUsers(req, res) {  
  let users = await getUsersForId(req.params.id)  
  res.json(users)  
}  
  
server.get('/users', getUsers)
```



BUT WE DON'T ACTUALLY CATCH IT 💣

ERROR HANDLING WITH ASYNC/AWAIT

```
Fri, 01 Sep 2017 17:21:39 GMT unhandledRejection
Error: Reached 5 failures on API User::UserRead_load_by_account. Circuit breaker is open now
and API is not available.
    at Breaker._run
(/dependencies/p2pnodeweb/cronus/scripts/node_modules/levee/lib/breaker.js:74:18)
    at Breaker.run
(/dependencies/p2pnodeweb/cronus/scripts/node_modules/levee/lib/breaker.js:60:15)
    at circuitCallback (/dependencies/p2pnodeweb/cronus/scripts/node_modules/servicecore-
hystrix/index.js:71:21)
    at onresponse
(/dependencies/p2pnodeweb/cronus/scripts/node_modules/levee/lib/breaker.js:121:18)
    at __container__
(/dependencies/p2pnodeweb/cronus/scripts/node_modules/levee/lib/zalgo.js:17:26)
    at
```

ERROR HANDLING WITH ASYNC/AWAIT

```
RETURN A STANDARD MIDDLEWARE FUNCTION  
EXECUTE THE ASYNC MIDDLEWARE  
PASS THOSE TO THE EXPRESS ERROR HANDLER
```

```
asyncMiddleware(req, res, next) {  
    next(err)  
}  
}
```

PASS IN YOUR ASYNC MIDDLEWARE

```
syncMiddleware) {  
    res, next) {  
        CATCH ANY ERRORS
```

```
req, res, next).catch(function(err) {  
    // handle error
```

APPLY AS NEEDED

```
server.get('/users', authenticate, handleError(getUsers))
```

ERROR HANDLING WITH ASYNC/AWAIT

```
Fri, 01 Sep 2017 17:21:39 GMT unhandledRejection
Error: Reached 5 failures on API User::UserRead_load_by_account. Circuit breaker is open now
and API is not available.
    at Breaker._run
(/dependencies/p2pnodeweb/cronus/scripts/node_modules/levee/lib/breaker.js:74:18)
    at Breaker.run
(/dependencies/p2pnodeweb/cronus/scripts/node_modules/levee/lib/breaker.js:60:15)
    at circuitCallback (/dependencies/p2pnodeweb/cronus/scripts/node_modules/servicecore-
hystrix/index.js:71:21)
    at onresponse
(/dependencies/p2pnodeweb/cronus/scripts/node_modules/levee/lib/breaker.js:121:18)
    at __container__
(/dependencies/p2pnodeweb/cronus/scripts/node_modules/levee/lib/zalgo.js:17:26)
    at
```

Make it easy for your engineers
to do the right thing

ERROR HANDLING WITH ASYNC/AWAIT

 README.md

ExpressJS Async Errors

build passing

A dead simple ES6 async/await support hack for [ExpressJS](#)

Shamelessly copied from [express-yields](#)

This has been lightly reworked to handle async rather than generators.

Usage

```
npm install express-async-errors --save
```

Then require this script somewhere **before** you start using it:

Async functions already work fine in Express.

```
const express = require('express');
require('express-async-errors');
const User = require('./models/user');
const app = express();
```

Custom Error Classes

CUSTOM ERRORS

```
async function getCard(params) {  
  if (!params.id) {  
    throw new Error("Missing ID")  
  }  
  // ...  
}
```

CUSTOM ERRORS

```
async function getCard(params) {  
  if (!params.id) {  
    throw new ParamError("id")  
  }  
  // ...  
}
```

CUSTOM ERRORS

```
class ParamError extends Error {  
    constructor(field) {  
        1. super(`Missing param: ${field}`)  
        2. Error.captureStackTrace(this, ParamError)  
        3. this.statusCode = 400  
    }  
}
```

CUSTOM ERRORS

```
function errorHandler(err, req, res, next) {  
  let statusCode = err.statusCode || 500  
  res.status(statusCode).send(err.message)  
}
```

CUSTOM ERRORS

```
async function getCard(params) {  
  if (!params.id) {  
    let err = new Error("Missing ID")  
    err.statusCode = 400  
    throw err  
  }  
  // ...  
}
```

CUSTOM ERRORS

```
async function getCard(params) {  
  if (!params.id) {  
    throw new ParamError("id")  
  }  
  // ...  
}
```

SUMMARY

- ▶ Don't use object literals or strings for errors (**missing stack trace**)
- ▶ Use the **Error** built-in object
- ▶ Subclass **Error** to add **statusCodes** or **to convert error codes into user-friendly error messages** for localization, etc
- ▶ We basically have one error class per micro-service to handle parsing the errors out of the response....

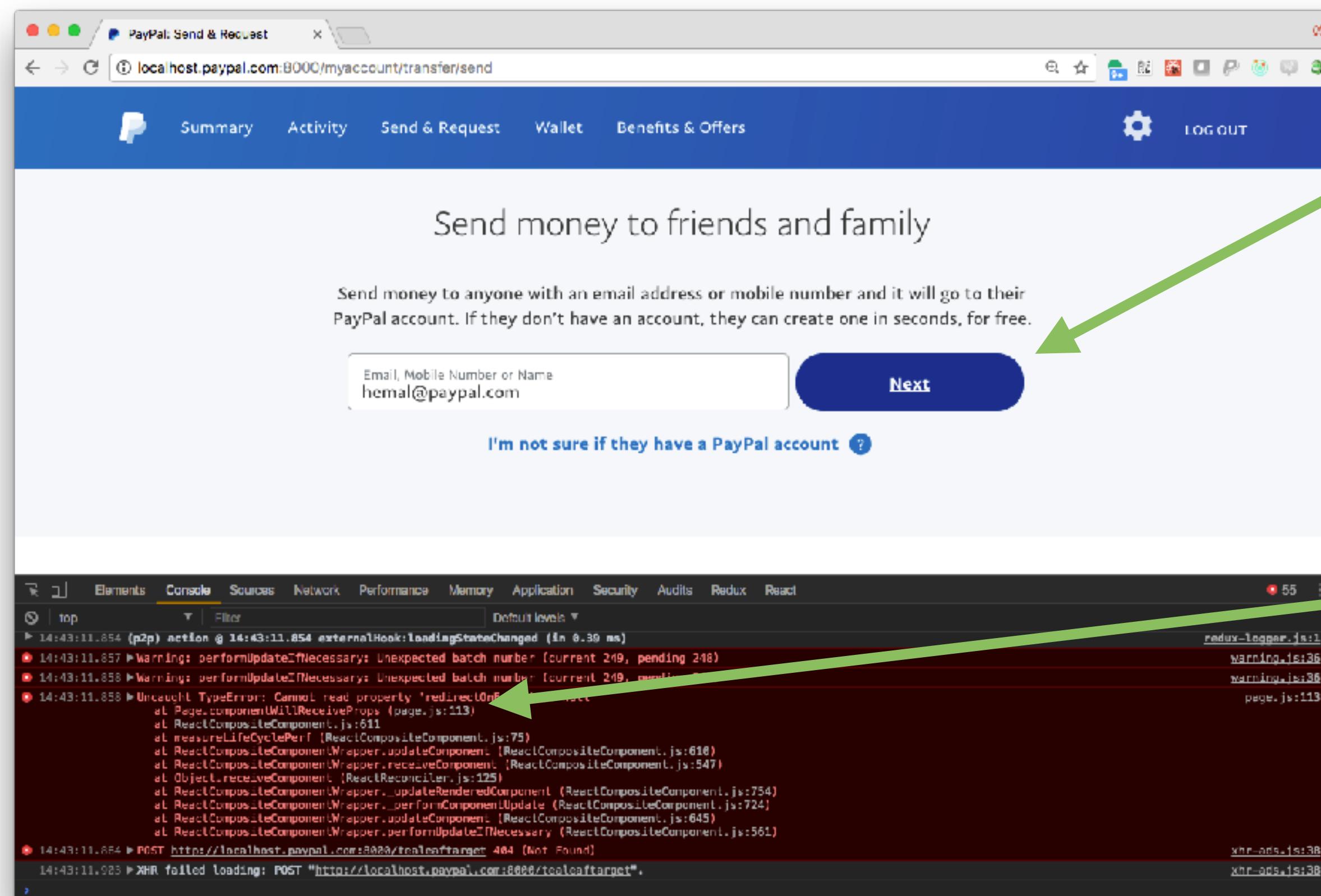


The Mystery of the Client-Side Errors

THE MYSTERY OF THE CLIENT-SIDE ERRORS

CLIENT-SIDE MONITORING

BUTTON DOESN'T WORK

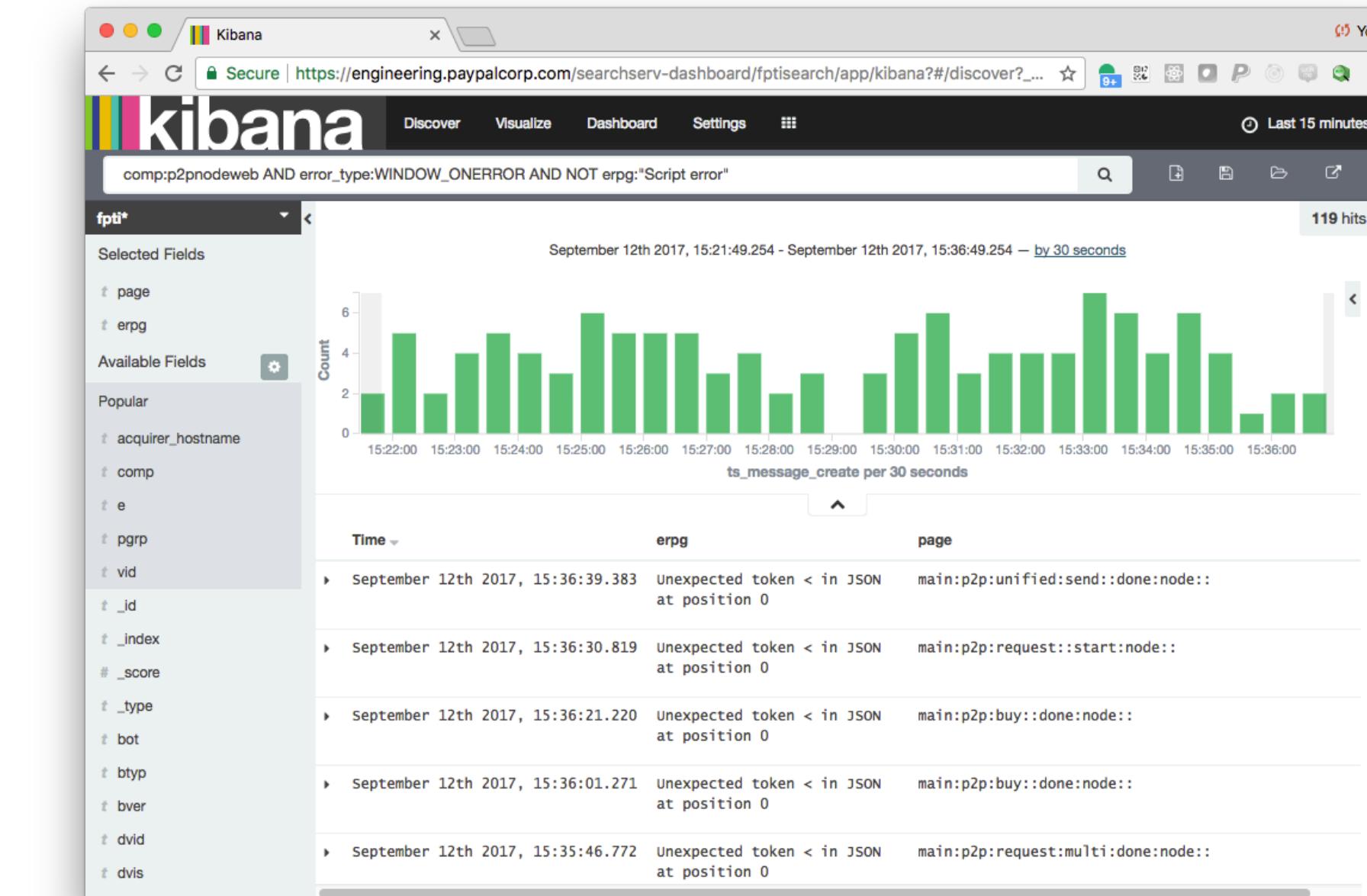
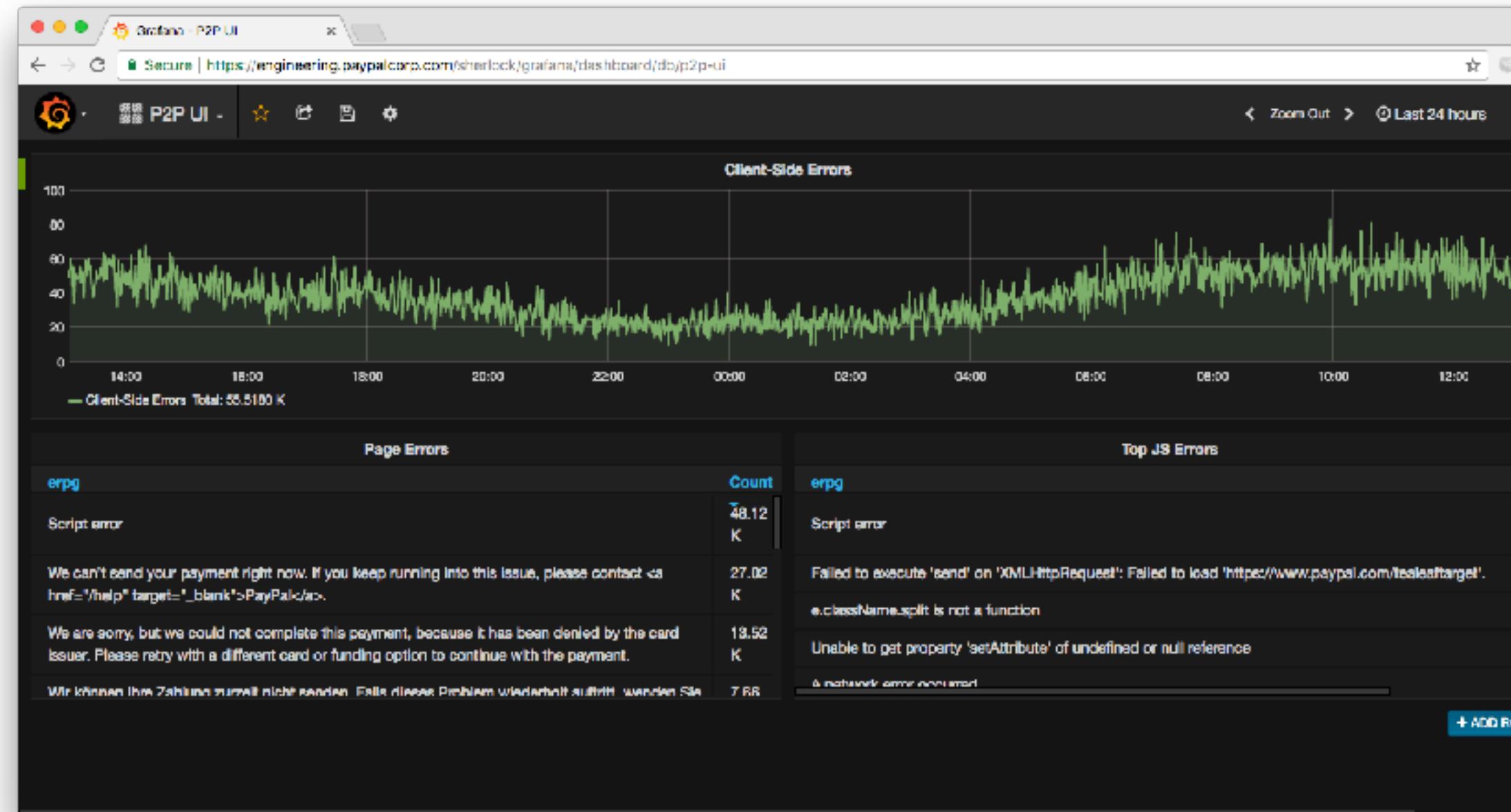


REAL ISSUE USUALLY IN DEV TOOLS

THE MYSTERY OF THE CLIENT-SIDE ERRORS

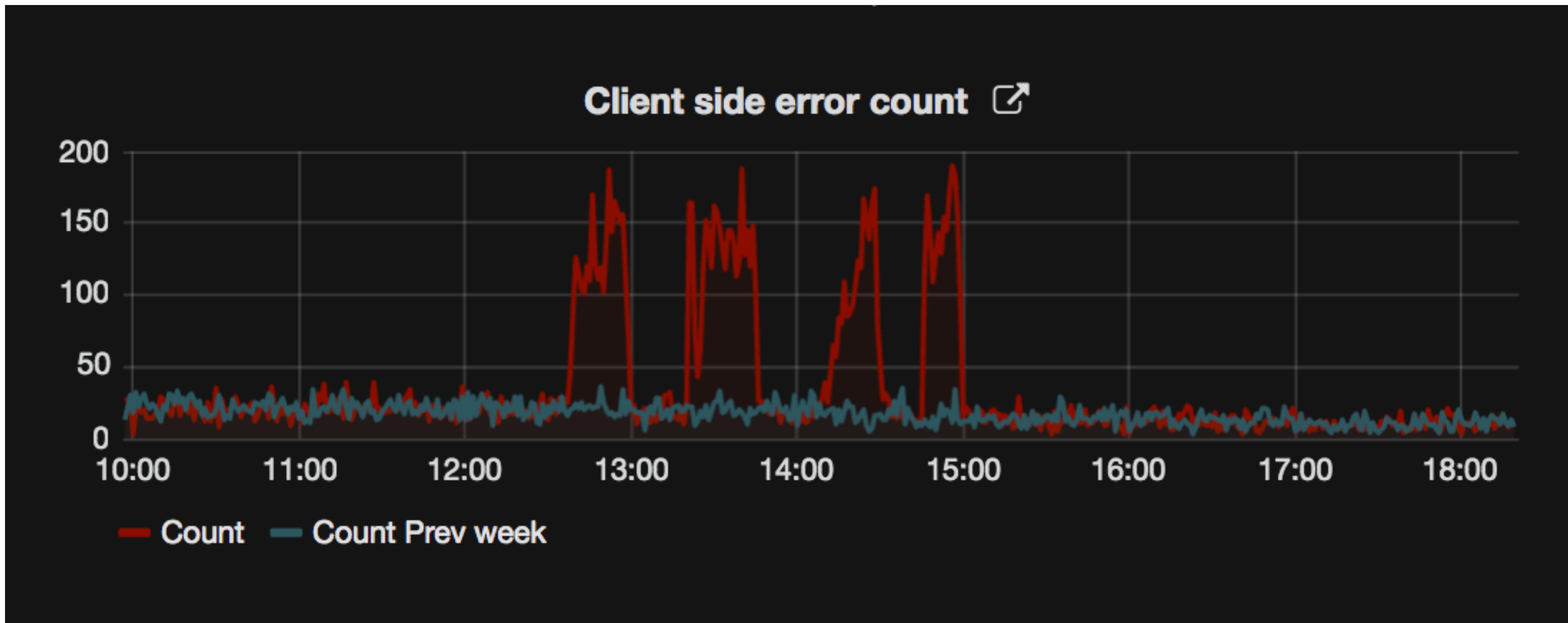
CLIENT-SIDE MONITORING

```
window.onerror = function (msg, url, line, col, error) {  
    // 1. clean up the data  
    // 2. log to server w/AJAX or sendBeacon() API  
}
```



THE MYSTERY OF THE CLIENT-SIDE ERRORS

WE NOTICED A SPIKE DURING DEPLOY



WE CONGRATULATED OURSELVES... THEN ACTUALLY LOOKED INTO THE BUG

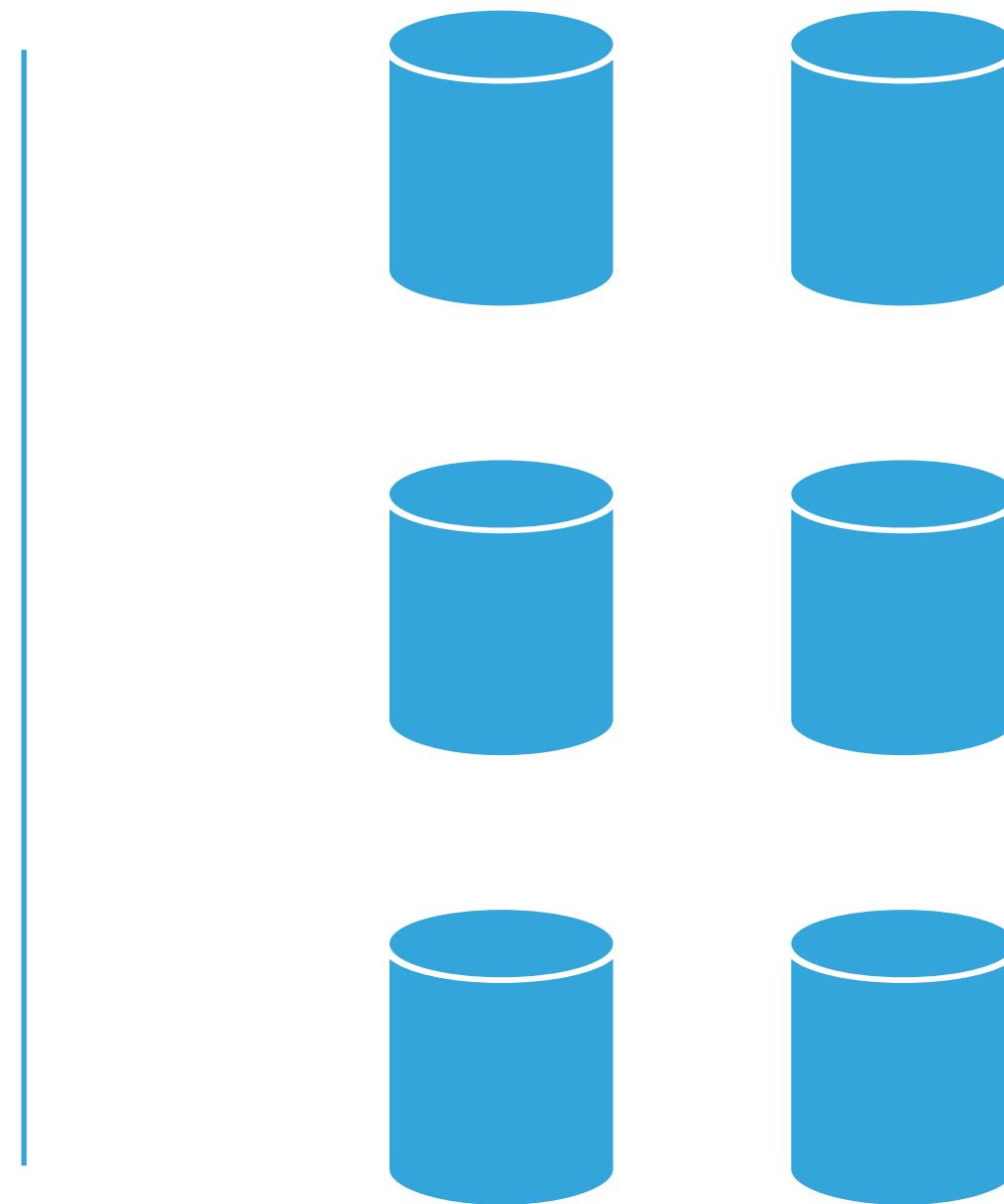
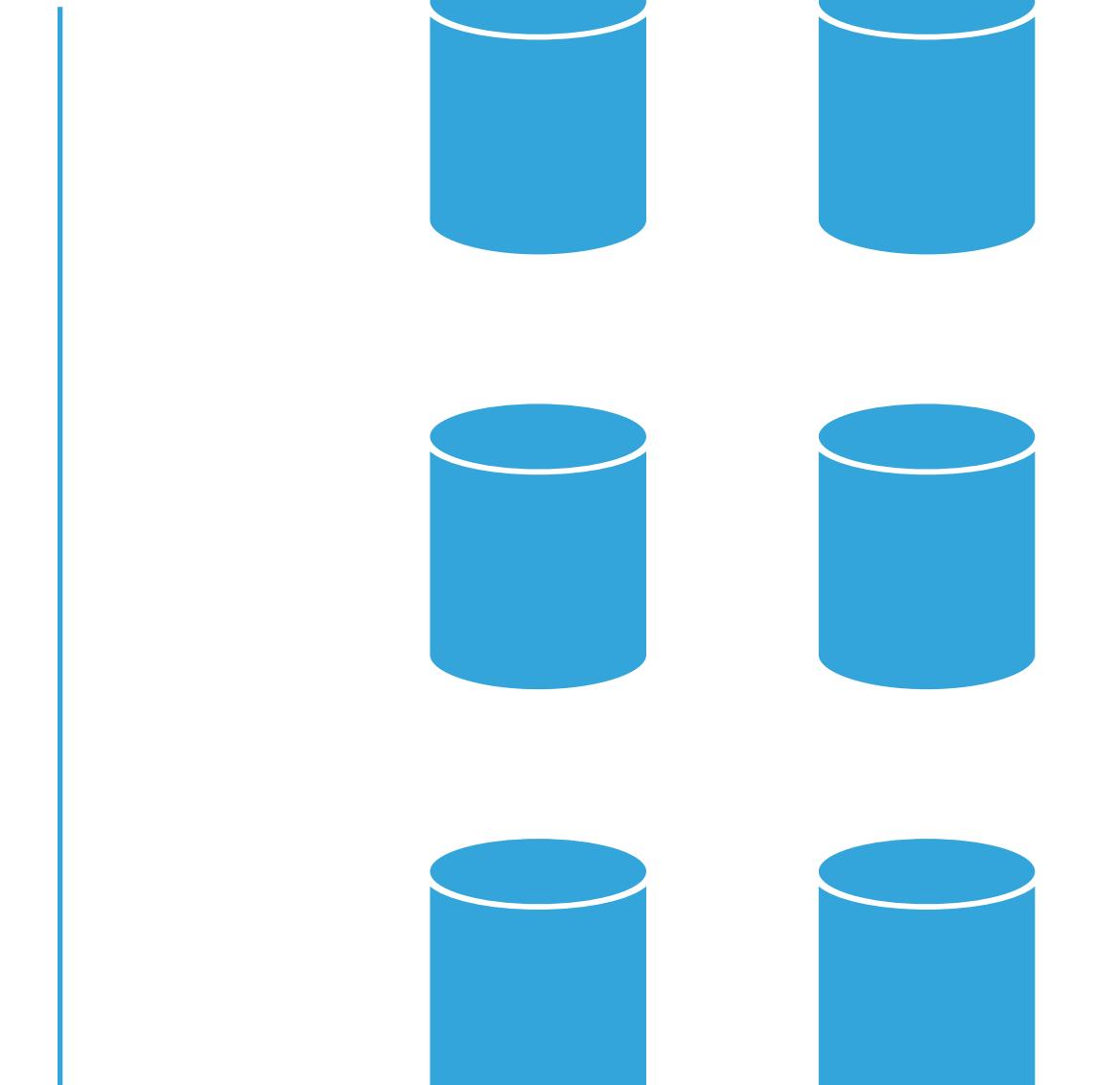
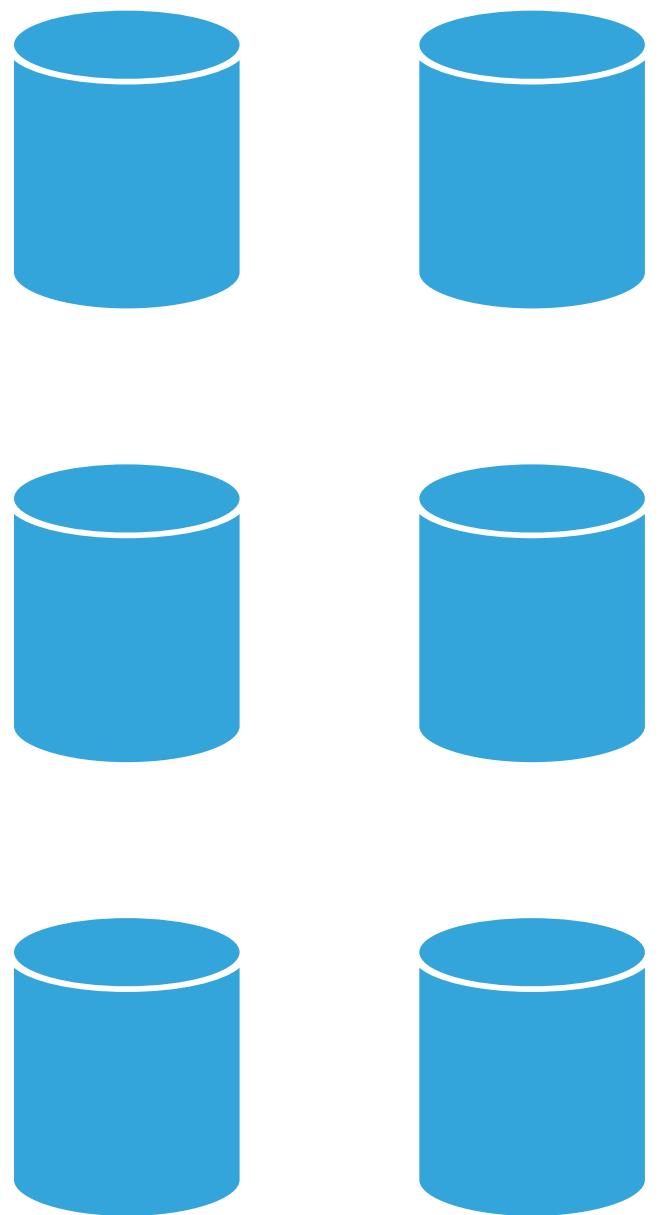


THE MYSTERY OF THE CLIENT-SIDE ERRORS

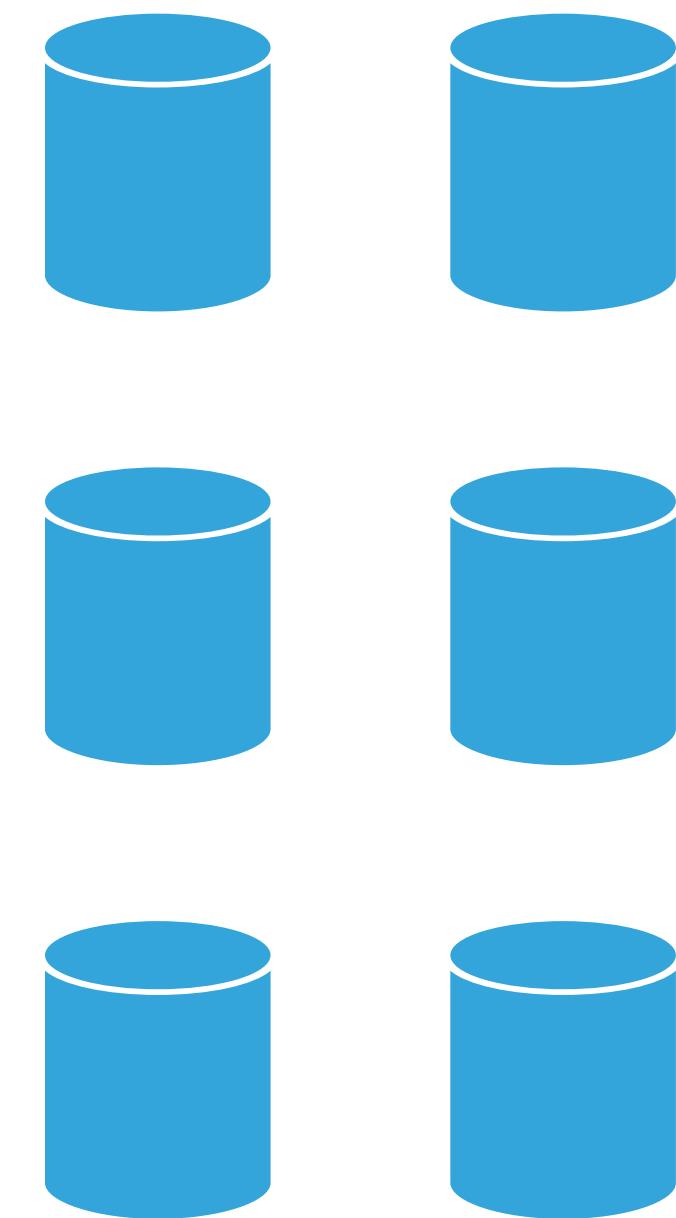
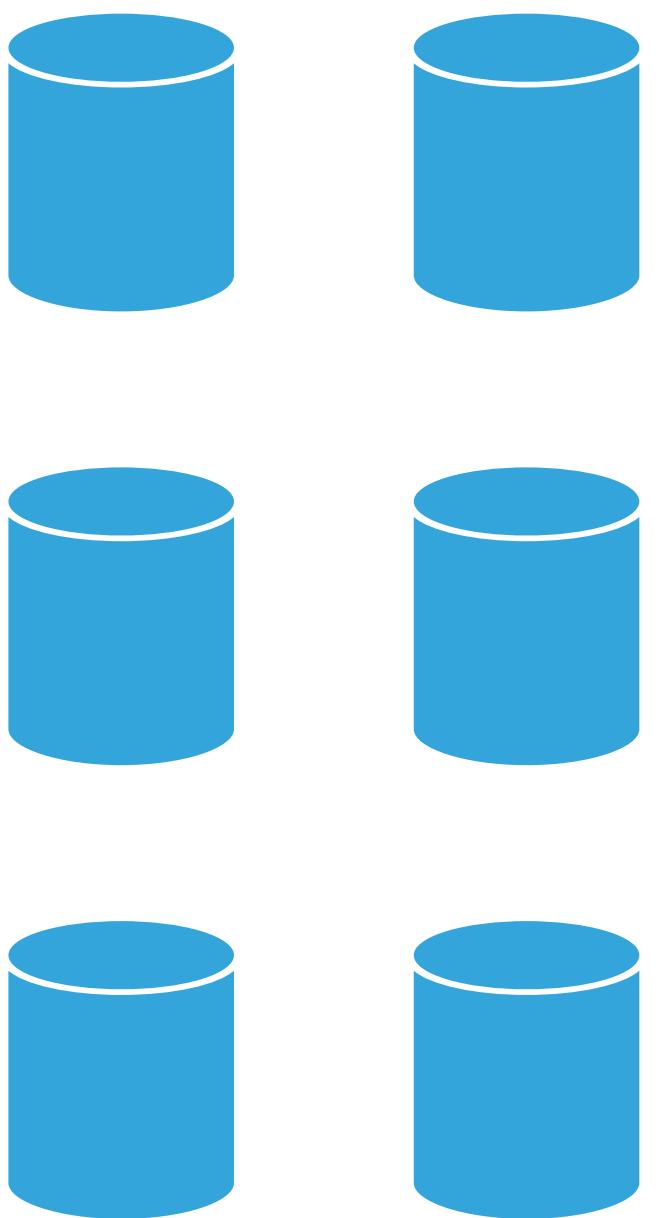
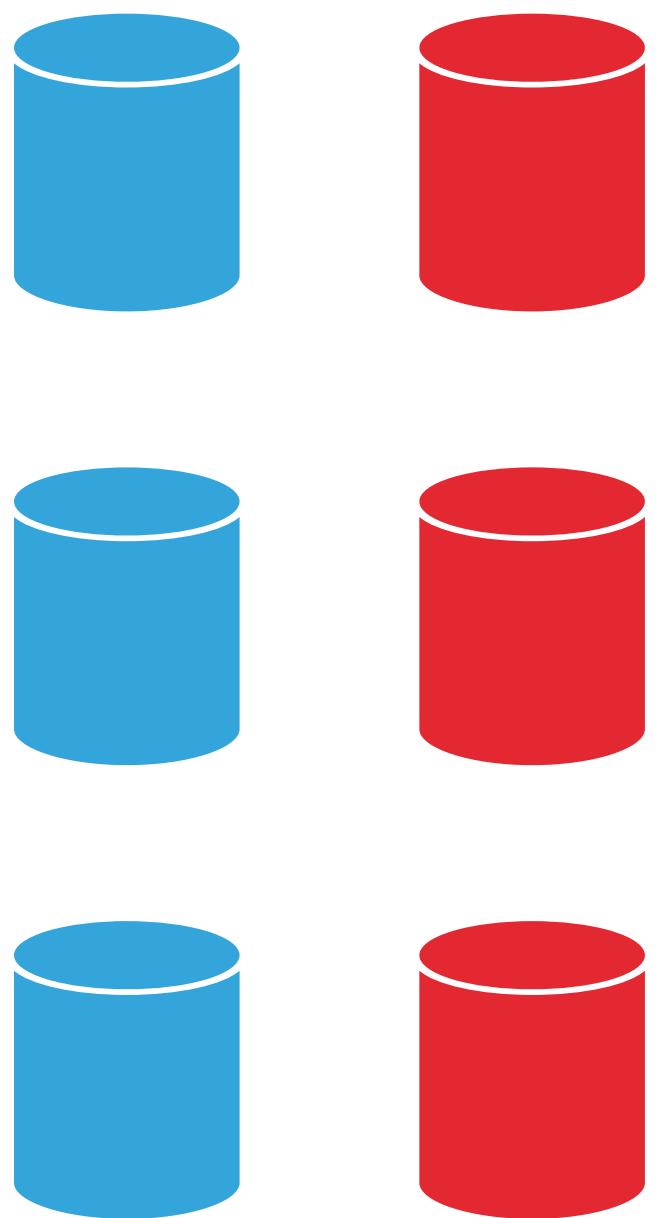
⚠ 23:17:44.285 ▼Unexpected token < in JSON at position 0
 defaultErrorHandler @ [xhr.js?37ff:27](#)
 onError @ [xhr.js?37ff:112](#)

Promise.then (async)
 (anonymous) @ [xhr.js?37ff:98](#)
 (anonymous) @ [Actions.js?b4ed**:307](#)
 (anonymous) @ [index.js:12](#)
 componentDidMount @ [send.js:79](#)
 proxiedComponentDidMount @ [createPrototypeProxy.js:66](#)
 (anonymous) @ [ReactCompositeComponent.js:262](#)
 measureLifeCyclePerf @ [ReactCompositeComponent.js:73](#)
 (anonymous) @ [ReactCompositeComponent.js:261](#)
 notifyAll @ [CallbackQueue.js:74](#)
 close @ [ReactReconcileTransaction.js:78](#)

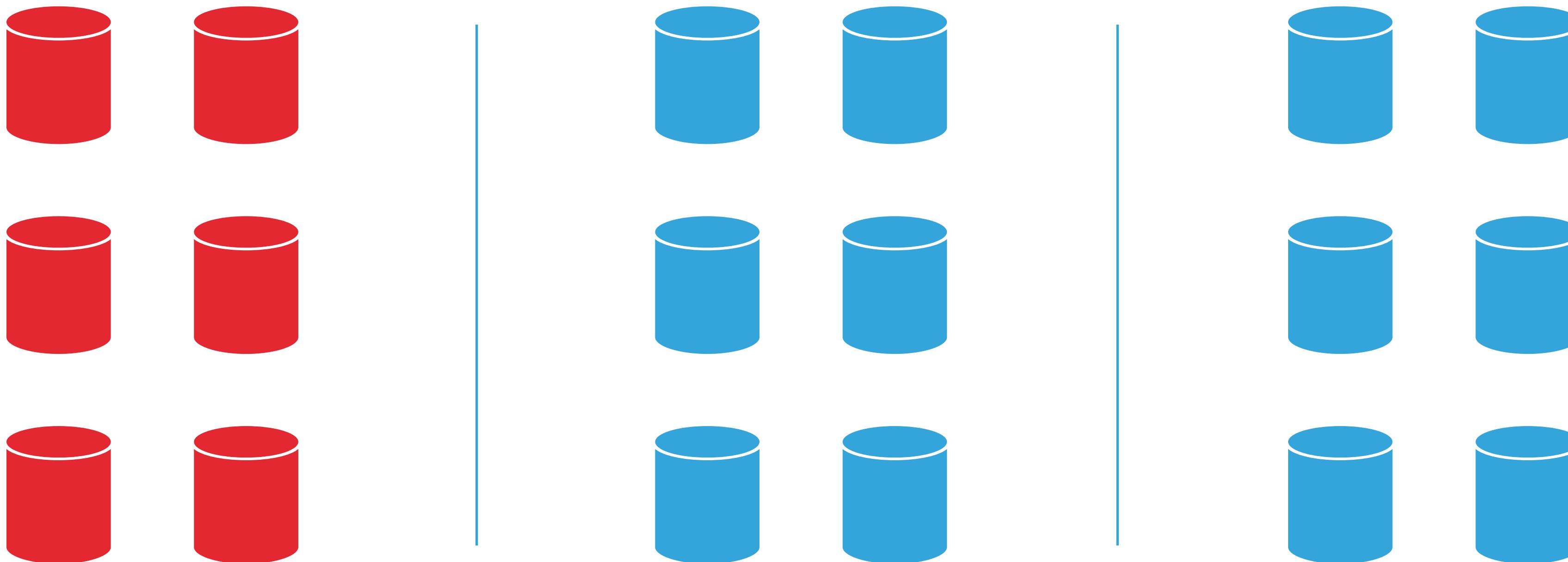
WE HAVE A LOT OF SERVERS



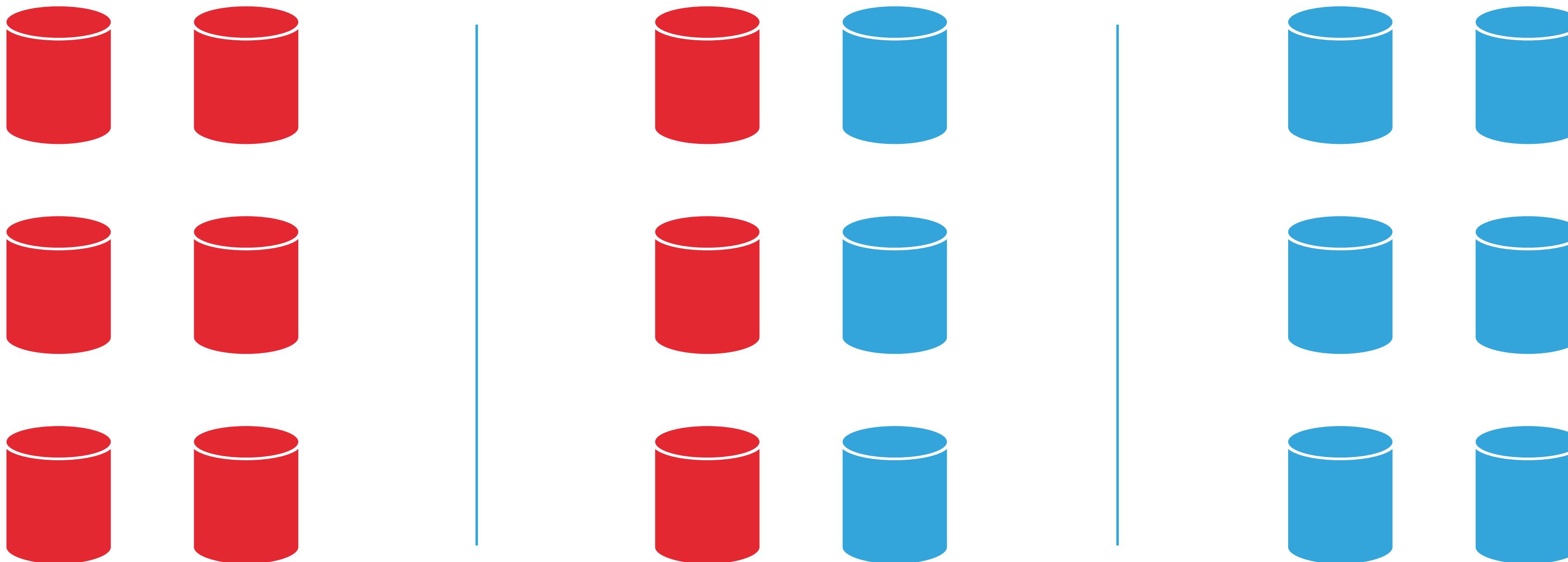
WE HAVE A LOT OF SERVERS



WE HAVE A LOT OF SERVERS

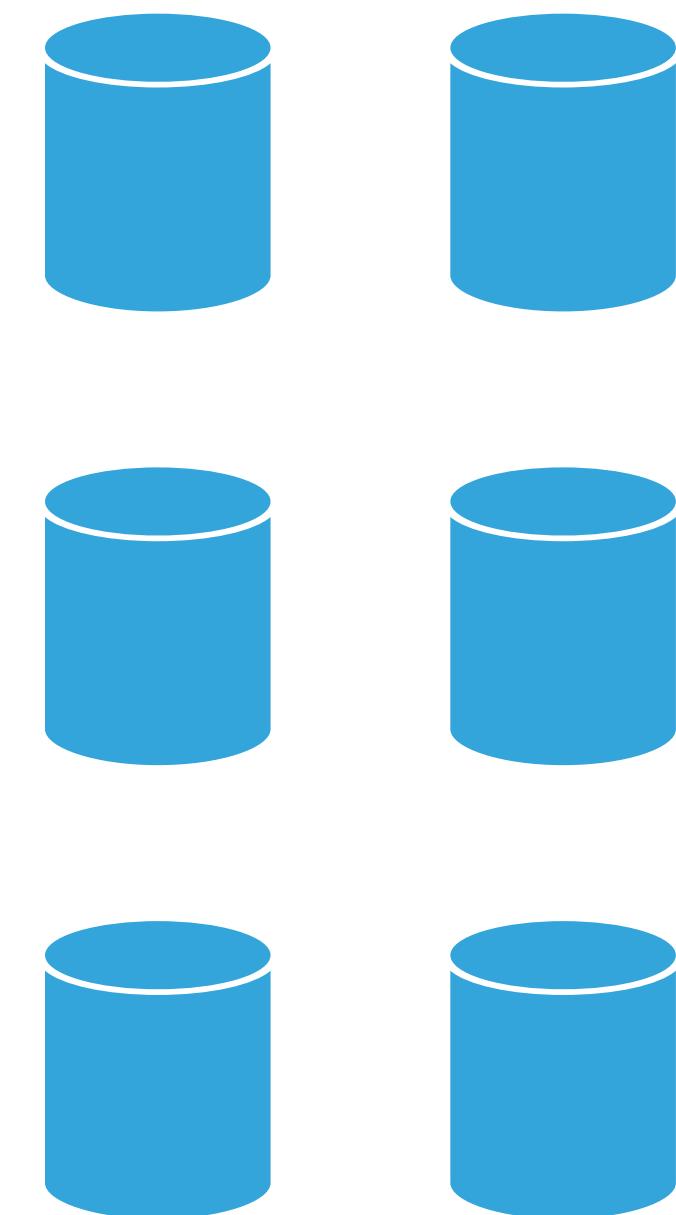
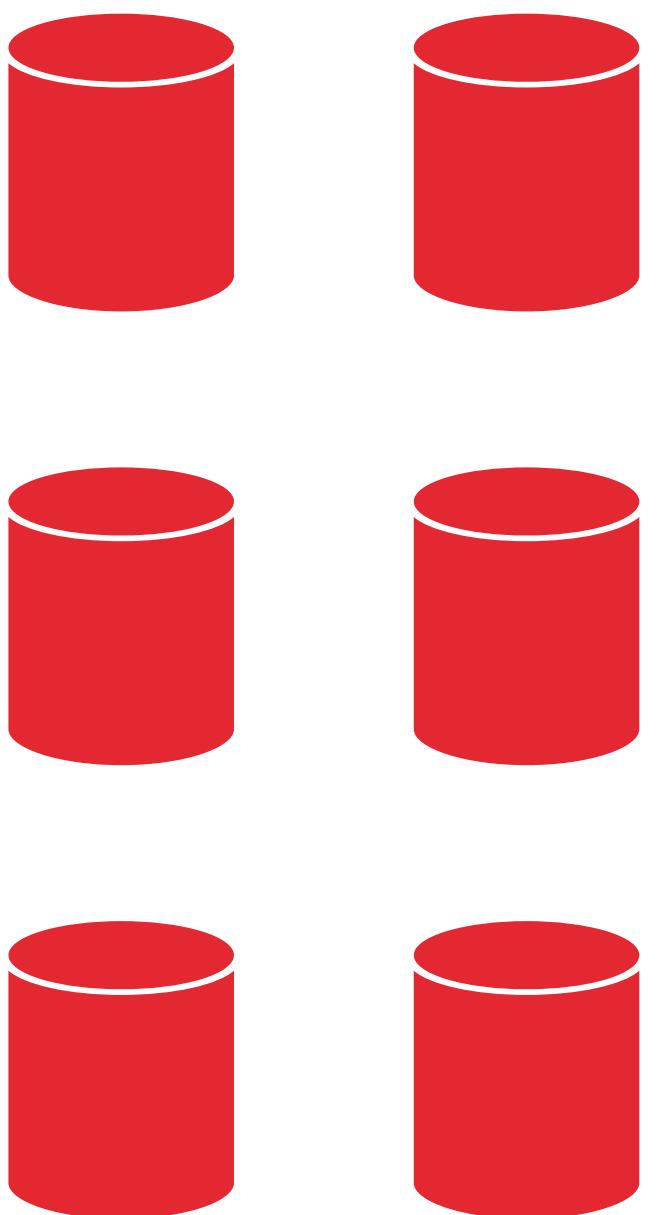
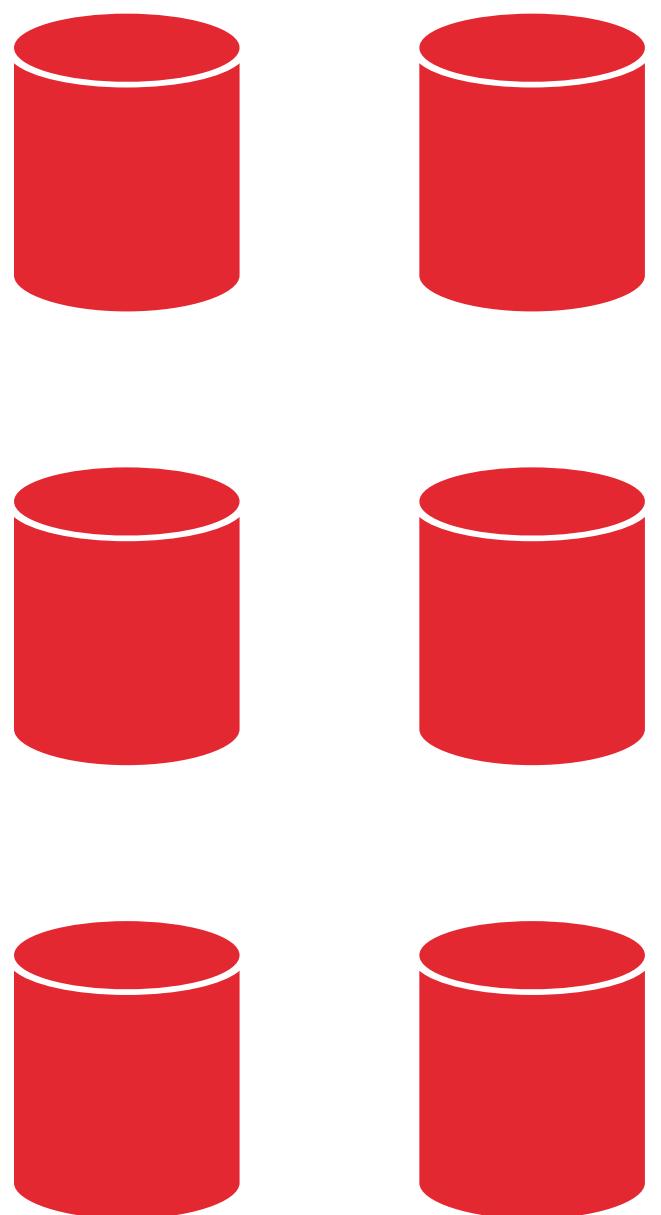


WE HAVE A LOT OF SERVERS

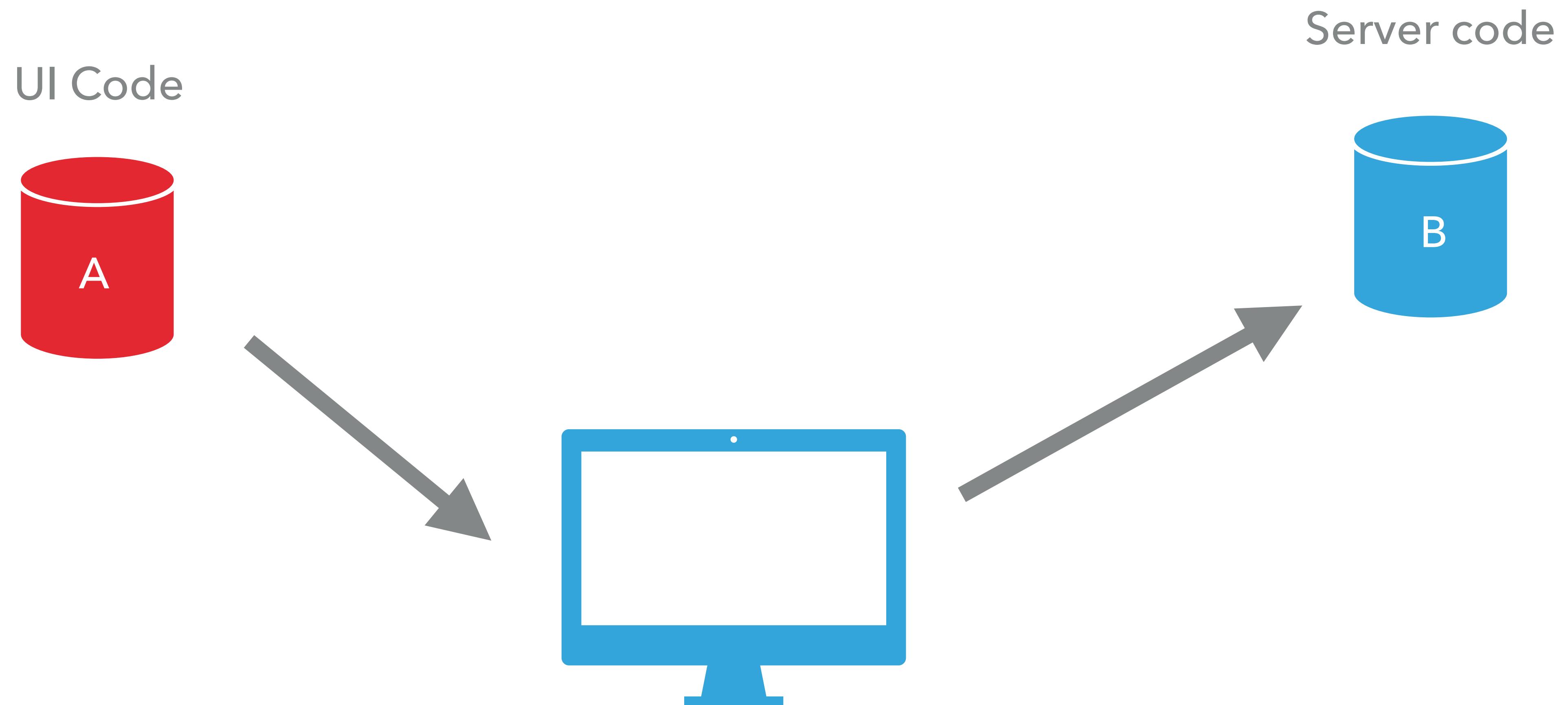


THE MYSTERY OF THE CLIENT-SIDE ERRORS

WE HAVE A LOT OF SERVERS

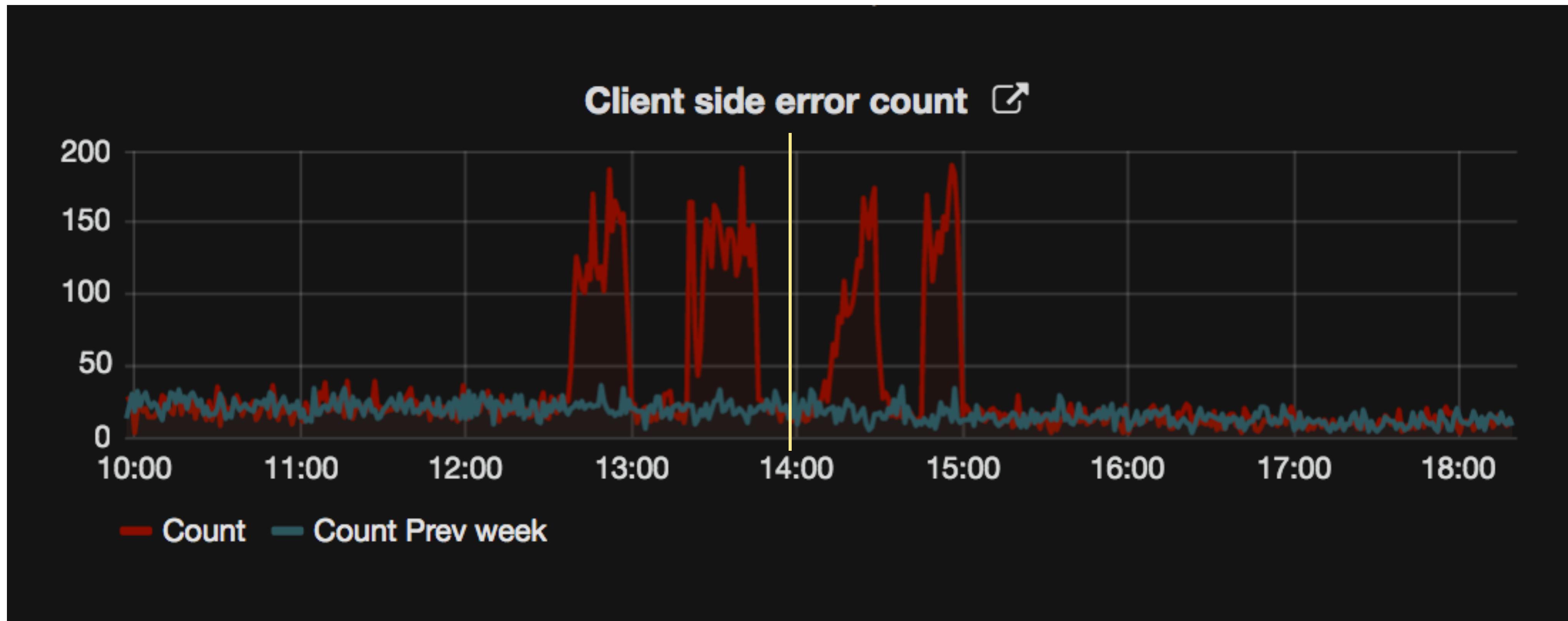


THE MYSTERY OF THE CLIENT-SIDE ERRORS



THE MYSTERY OF THE CLIENT-SIDE ERRORS

ROLLING BACK MADE THINGS WORSE



LESSONS LEARNED

- ▶ UI is a huge monitoring blind-spot
- ▶ Be aware of how the deploy process affects users
- ▶ Try to make your code changes backwards compatible
- ▶ Consider separating UI and server deploys

- ▶ Use static analysis (including types) to catch bugs early
- ▶ Have a plan for debugging apps in production
- ▶ Adopt a consistent approach to error handling
- ▶ Know how to access all of your logs
- ▶ Don't forget to monitor client-side errors

THE END