BUILDING SCALABLE AND DEPENDABLE

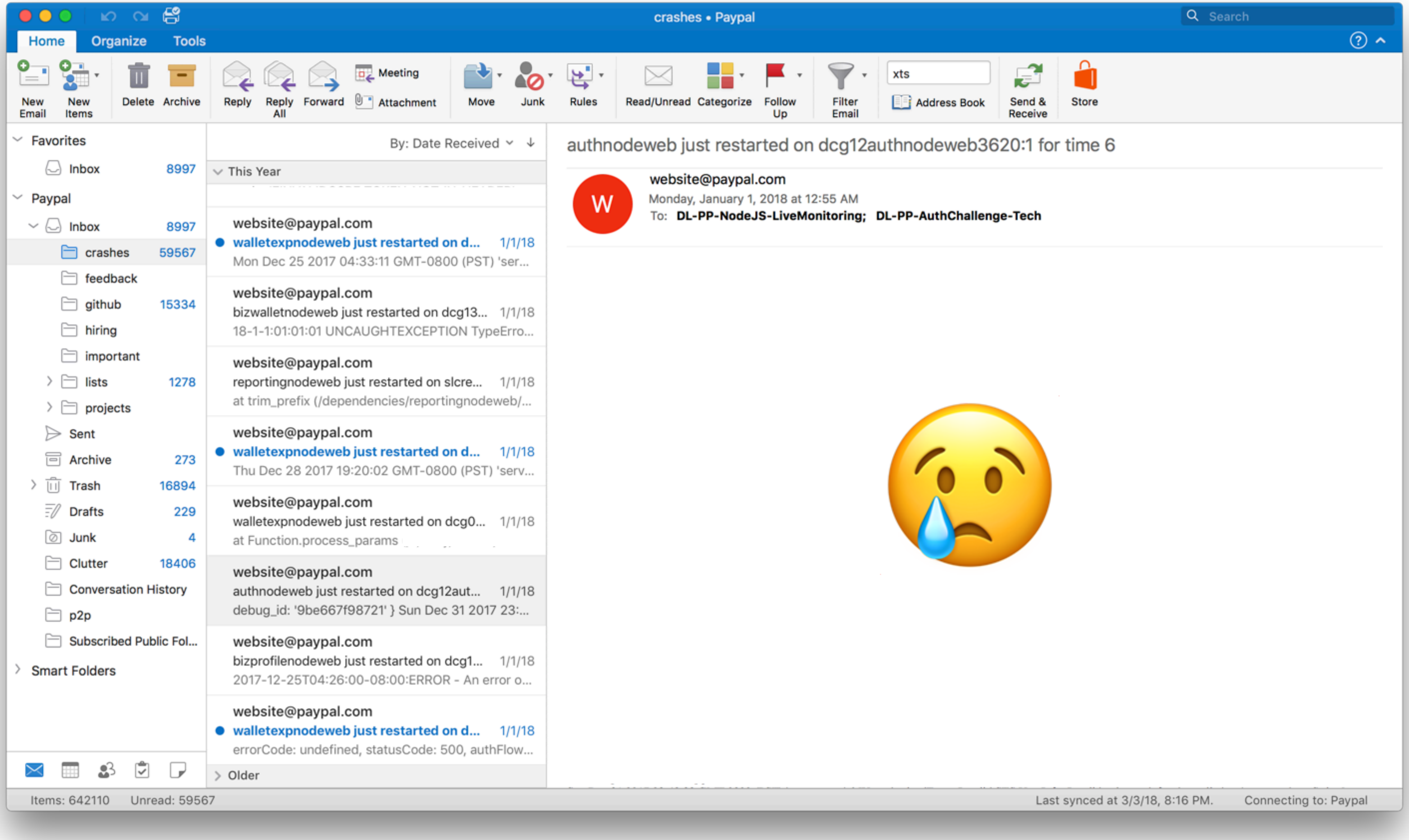
JAMUND FERGUSON

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)
```



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

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

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

authnodeweb just restarted on dcg12authnodeweb3620:1 for time 6

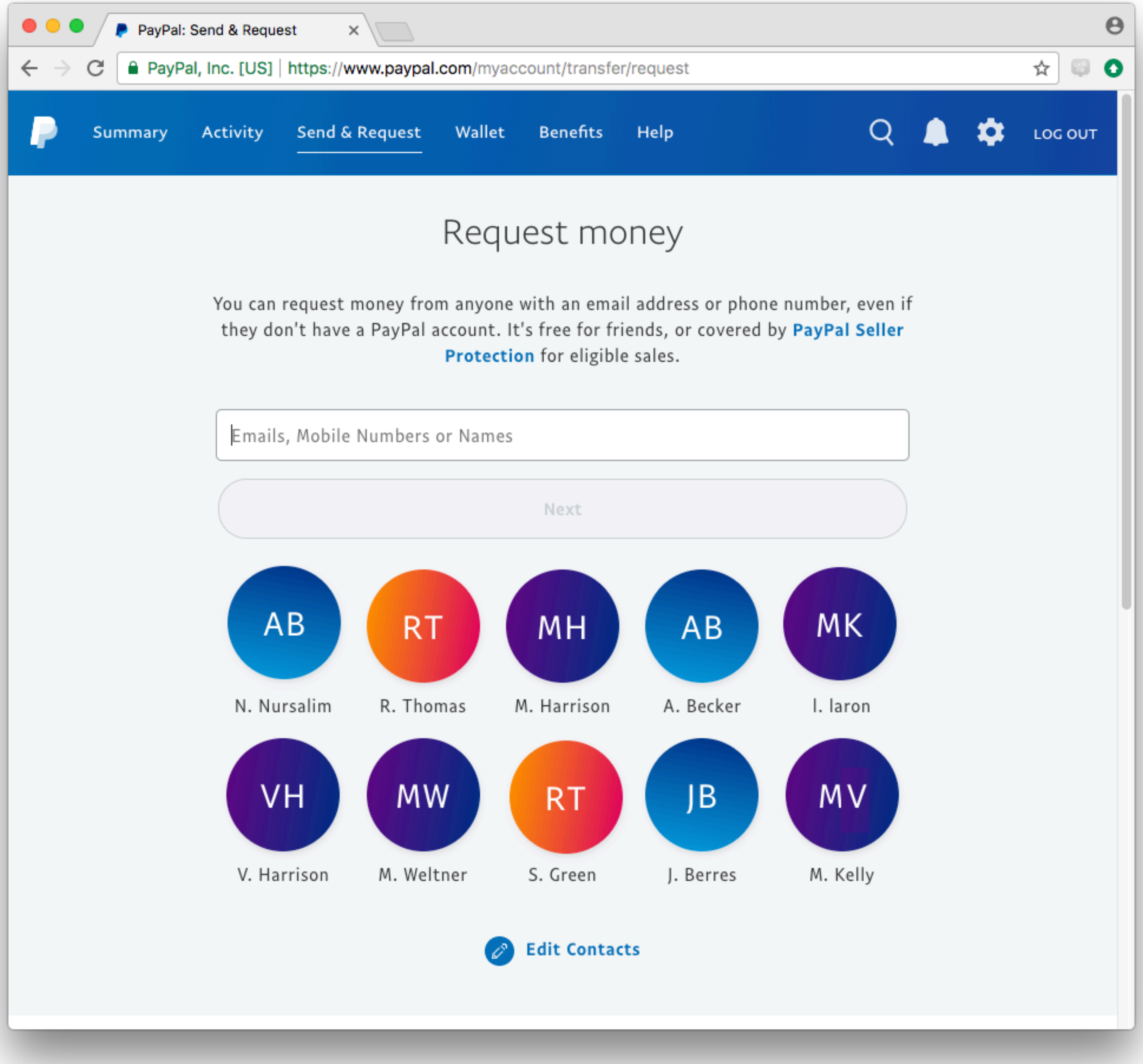
W
website@paypal.com
Monday, January 1, 2018 at 12:55 AM
To: **DL-PP-NodeJS-LiveMonitoring; DL-PP-AuthChallenge-Tech**

Items: 642110 Unread: 59567 Last synced at 3/3/18, 8:16 PM. Connecting to: Paypal

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

WHAT CAN WE DO?

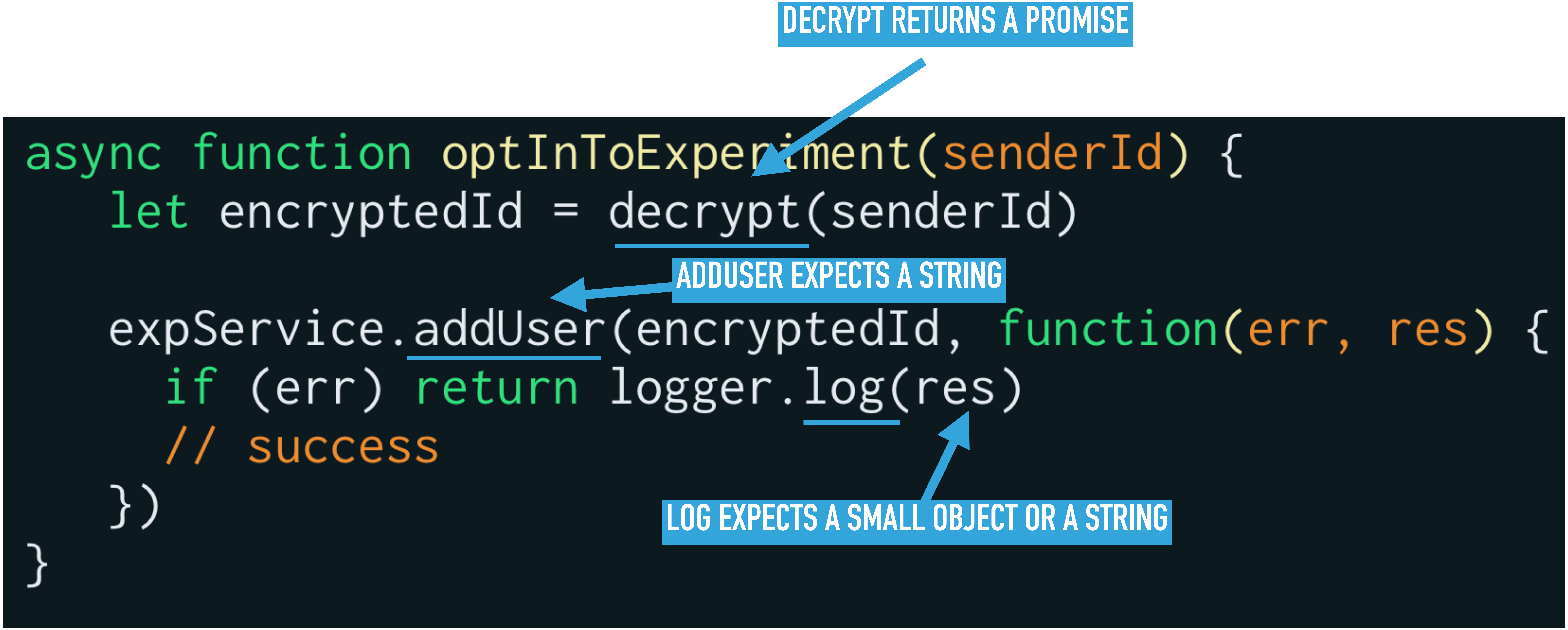
- Compare a diff between last known working code
- Look through other logs for more information (nginx logs, access logs, etc)
- Look at system metrics (is there a memory leak or CPU spike?)
- Add console.log statements somewhere???
- Advanced debugging techniques (post-mortem debugging, heapdumps, etc.)



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.

```
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)
  }
}
```

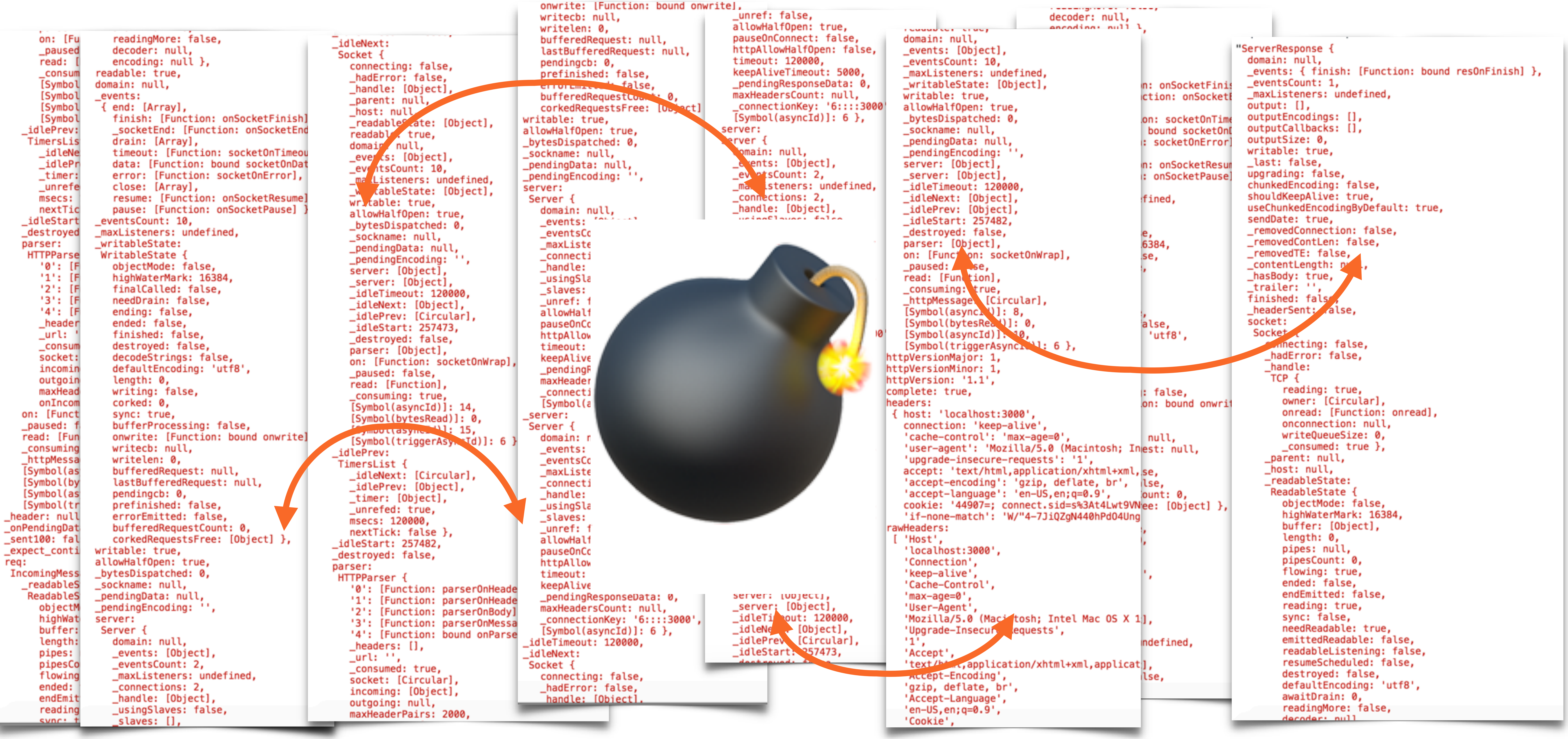


DECRYPT RETURNS A PROMISE

```
async function optInToExperiment(senderId) {
   let encryptedId = decrypt(senderId)

   expService.addUser(encryptedId, function(err, res) {
     if (err) return logger.log(res)
     // success
   })
}
```

ADDUSER EXPECTS A STRING

LOG EXPECTS A SMALL OBJECT OR A STRING



```
on: [Fu
_paused
read: [
_consum
[Symbol
[Symbol
[Symbol
[Symbol
_idlePrev:
 TimersLis
  _idleNe
  _idlePr
  _timer:
  _unrefe
  msecs:
  nextTic
_idleStart
_destroyed
parser:
 HTTPParse
  '0': [F
  '1': [F
  '2': [F
  '3': [F
  '4': [F
  _header
  _url: '
  _consum
  socket:
  incomin
  outgoin
  maxHead
  onIncom
on: [Funct
_paused: f
read: [Fun
_consuming
_httpMessa
[Symbol(as
[Symbol(by
[Symbol(as
[Symbol(tr
_header: null
_onPendingDat
_sent100: fal
_expect_conti
req:
 IncomingMess
  _readableS
   ReadableS
    objectM
    highWat
    buffer:
    length:
    pipes:
    pipesCo
    flowing
    ended:
    endEmit
    reading
```

```
   readingMore: false,
   decoder: null,
   encoding: null },
readable: true,
domain: null,
_events:
 { end: [Array],
   finish: [Function: onSocketFinish]
   _socketEnd: [Function: onSocketEnd
   drain: [Array],
   timeout: [Function: socketOnTimeou
   data: [Function: bound socketOnDat
   error: [Function: socketOnError],
   close: [Array],
   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: parserOnHeade
   '1': [Function: parserOnHeade
   '2': [Function: parserOnBody]
   '3': [Function: parserOnMessa
   '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:
   _eventsCo
   _maxListe
   _connecti
   _handle:
   _usingSla
   _slaves:
   _unref: f
   allowHalf
   pauseOnCo
   httpAllow
   timeout:
   keepAlive
   _pendingR
   maxHeader
   _connecti
   [Symbol(a
_server:
 Server {
   domain: n
   _events:
   _eventsCo
   _maxListe
   _connecti
   _handle:
   _usingSla
   _slaves:
   _unref: f
   allowHalf
   pauseOnCo
   httpAllow
   timeout:
   keepAlive
   _pendingResponseData: 0,
   maxHeadersCount: null,
   _connectionKey: '6::::3000',
   [Symbol(asyncId)]: 6 },
_idleTimeout: 120000,
_idleNext:
 Socket {
   connecting: false,
   _hadError: false,
   _handle: [Object],
```

```
   _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,
```

```
   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],
   _consuming: 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; In
   'upgrade-insecure-requests': '1',
   accept: 'text/html,application/xhtml+xml,
   'accept-encoding': 'gzip, deflate, br',
   'accept-language': 'en-US,en;q=0.9',
   cookie: '44907=; connect.sid=s%3At4Lwt9VN
   'if-none-match': 'W/"4-7JiQZgN440hPdO4Ung
rawHeaders:
 [ 'Host',
   'localhost:3000',
   'Connection',
   'keep-alive',
   'Cache-Control',
   'max-age=0',
   'User-Agent',
   'Mozilla/5.0 (Macintosh; Intel Mac OS X 1
   'Upgrade-Insecure-Requests',
   '1',
   'Accept',
   'text/html,application/xhtml+xml,applicat
   'Accept-Encoding',
   'gzip, deflate, br',
   'Accept-Language',
   'en-US,en;q=0.9',
   'Cookie',
```

```
decoder: null,
encoding: null },

n: onSocketFinis
ction: onSocketE
on: socketOnTime
 bound socketOnD
: socketOnError]
n: onSocketResu
: onSocketPause]
fined,
e,
6384,
se,
alse,
'utf8',
: false,
on: bound onwrit
null,
est: null,
se,
lse,
ount: 0,
ee: [Object] },
ndefined,
lse,
```

```
"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

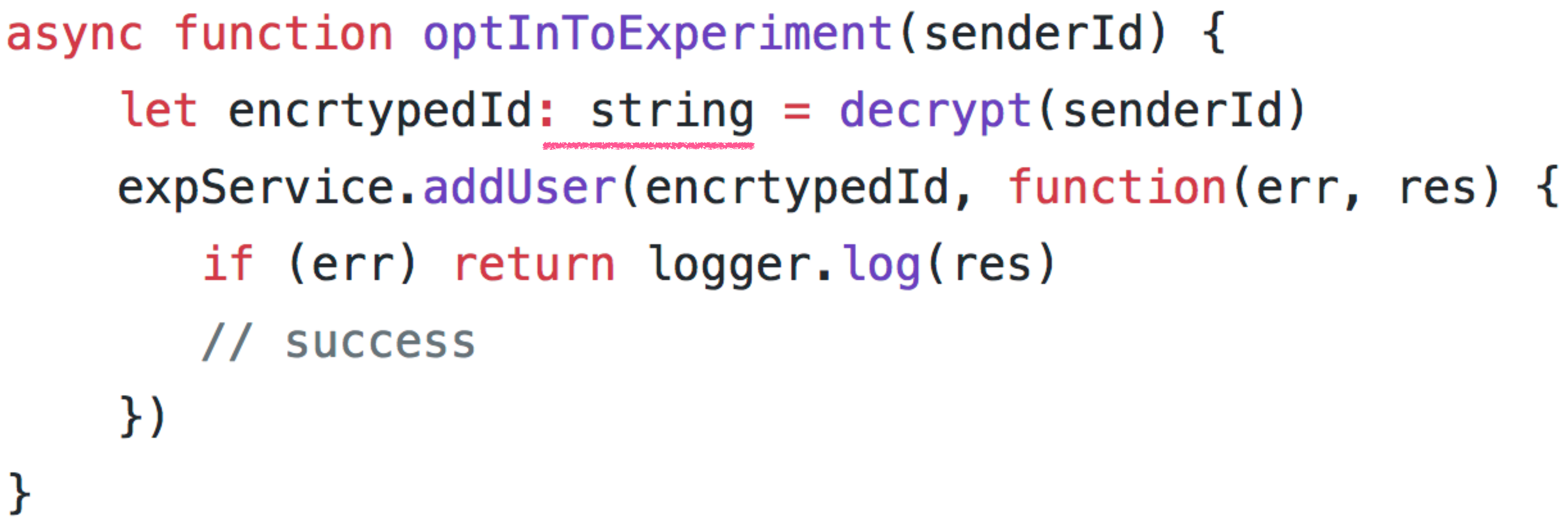
- We need better **static analysis**
- We need better **debugging tools**
- We need a **consistent way to handle errors**
- We need to **better understand our logging & monitoring**

STATIC ANALYSIS WITH

FLOWTYPE & ESLINT

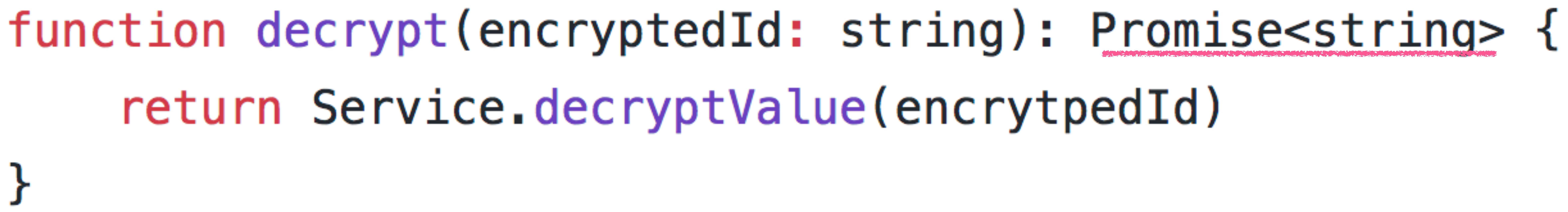
TYPE CHECKING COULD HAVE CAUGHT THAT BUG WITH 2-LINES OF CODE

PREVENTING BUGS WITH TYPES



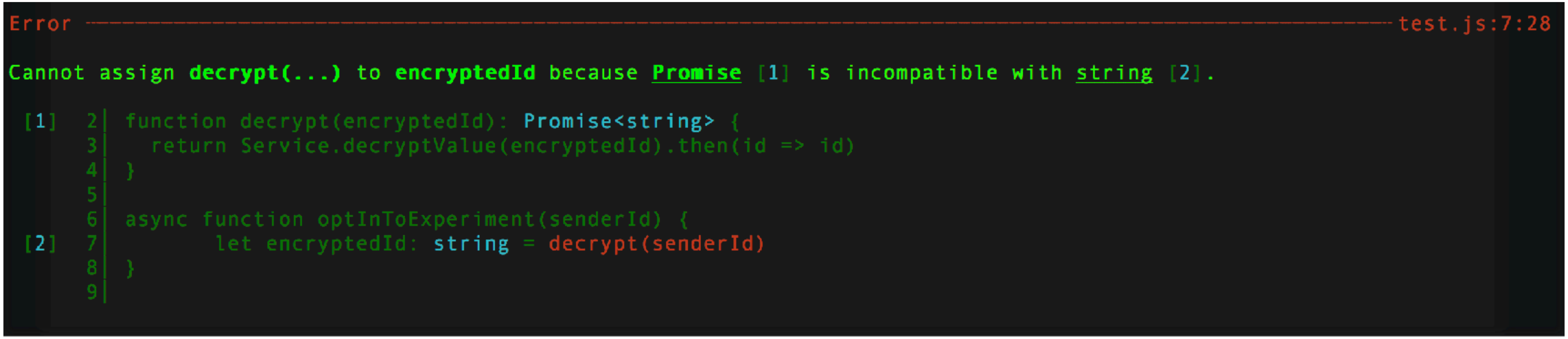
```
async function optInToExperiment(senderId) {
    let encrtypedId: string = decrypt(senderId)
    expService.addUser(encrtypedId, function(err, res) {
        if (err) return logger.log(res)
        // success
    })
}
```

PREVENTING BUGS WITH TYPES



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

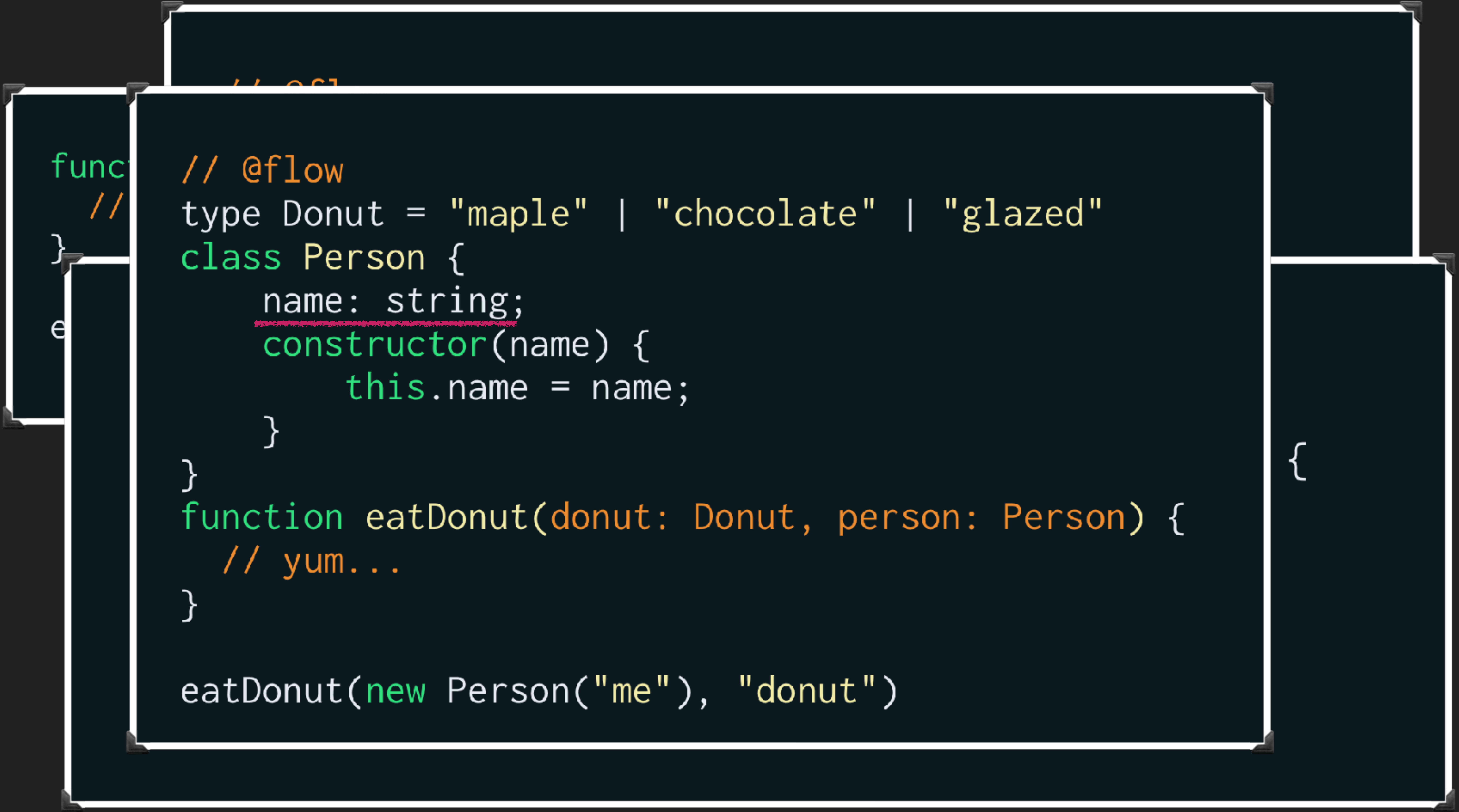
FLOW WON'T LET THAT SLIDE



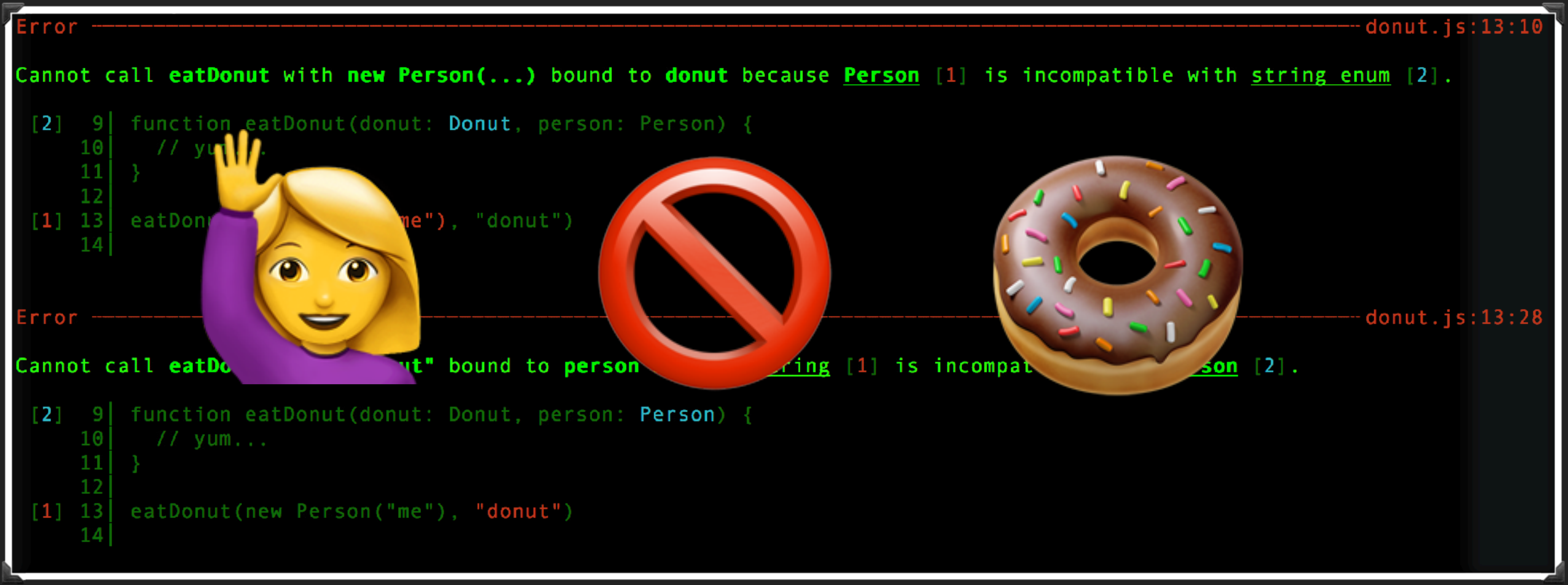
```
Error ------------------------------------------------------------------------------------------------ test.js:7:28

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

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



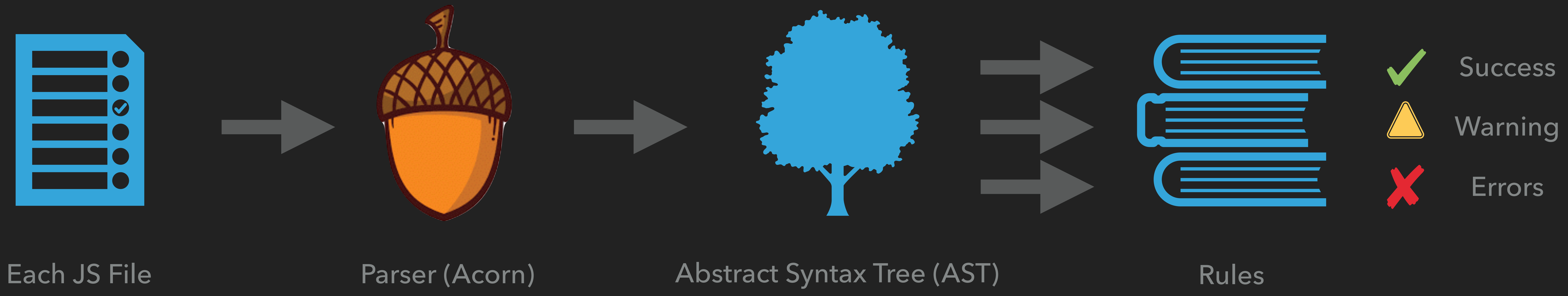
```
// @flow
type Donut = "maple" | "chocolate" | "glazed"
class Person {
    name: string;
    constructor(name) {
        this.name = name;
    }
}
function eatDonut(donut: Donut, person: Person) {
  // yum...
}

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|   // yu    .
      11| }
      12|
 [1]  13| eatDon         me"), "donut")
      14|


Error ------                  ------------------------------------------------------------------------ donut.js:13:28

Cannot call eatDo          ut" bound to person          ring [1] is incompat              son [2].

 [2]   9| function eatDonut(donut: Donut, person: Person) {
      10|   // yum...
      11| }
      12|
 [1]  13| eatDonut(new Person("me"), "donut")
      14|
```

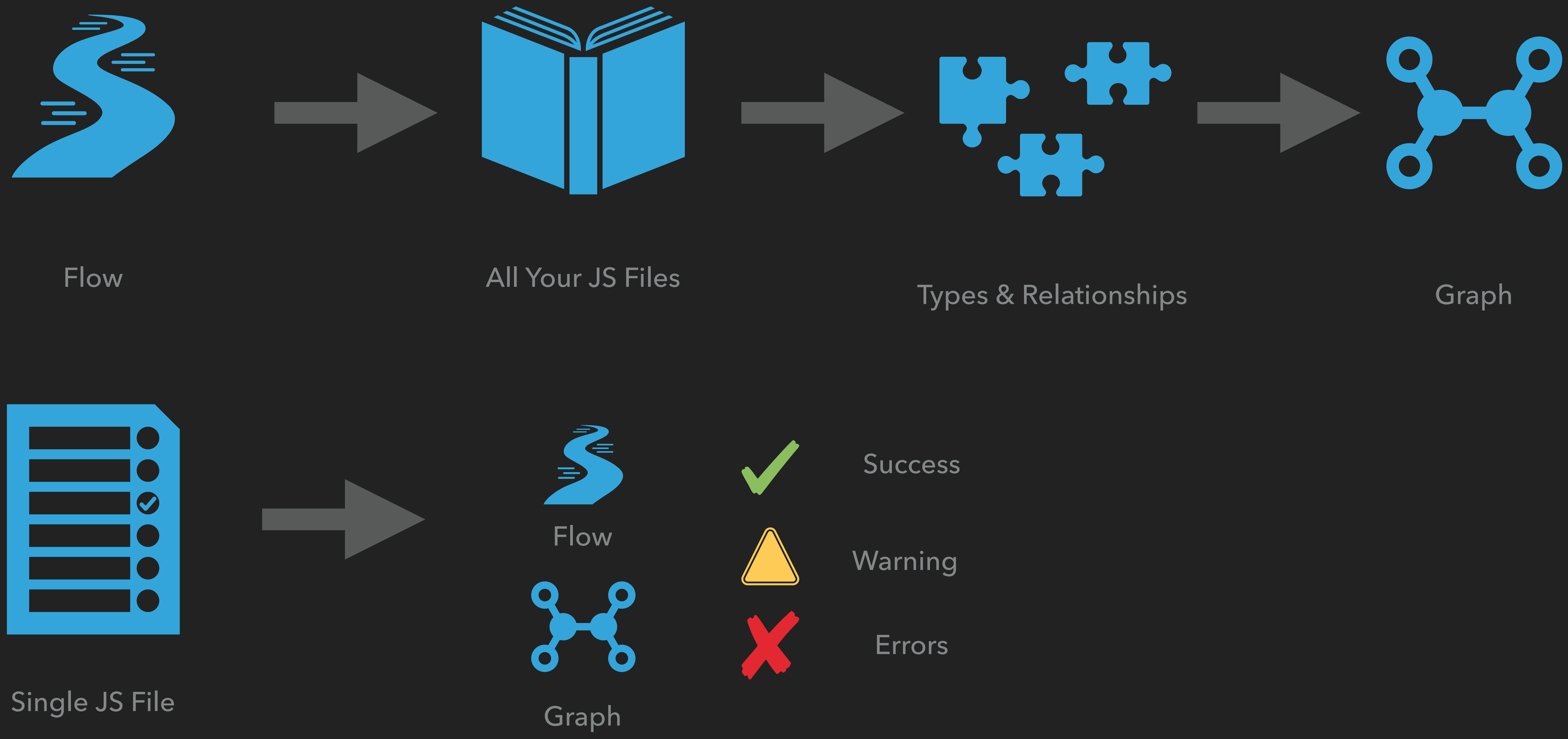
Type systems have a lot in common with linters

Architecture of JavaScript Linter (ESLint)



Linters can only think about one file at a time

Architecture of JavaScript Type System (Flow)



Flow

All Your JS Files

Types & Relationships

Graph

Single JS File

Flow

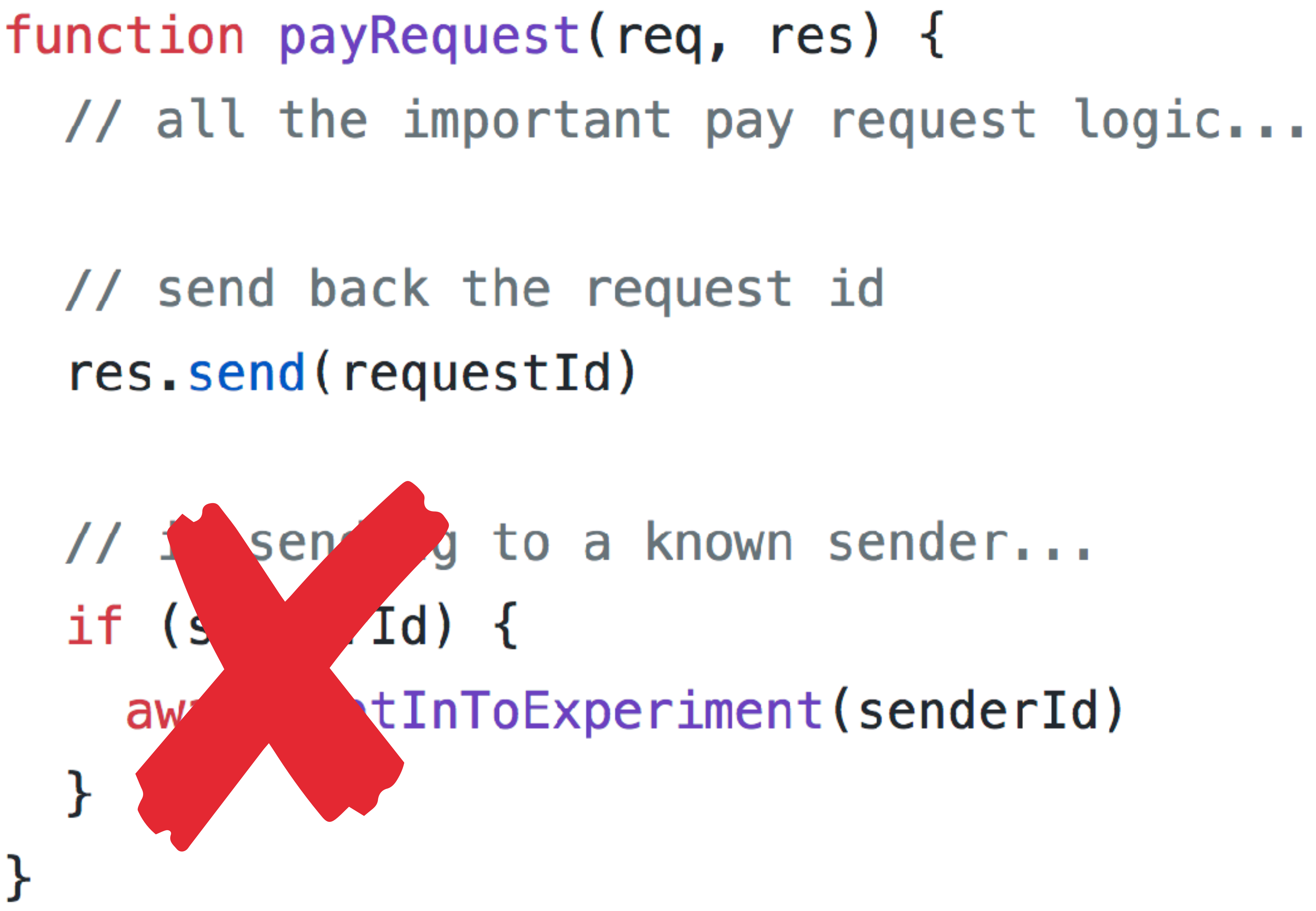
Graph

✓ Success

⚠ Warning

✗ Errors

COULD A LINTER HAVE HELPED US WITH OUR MYSTERY BUG?



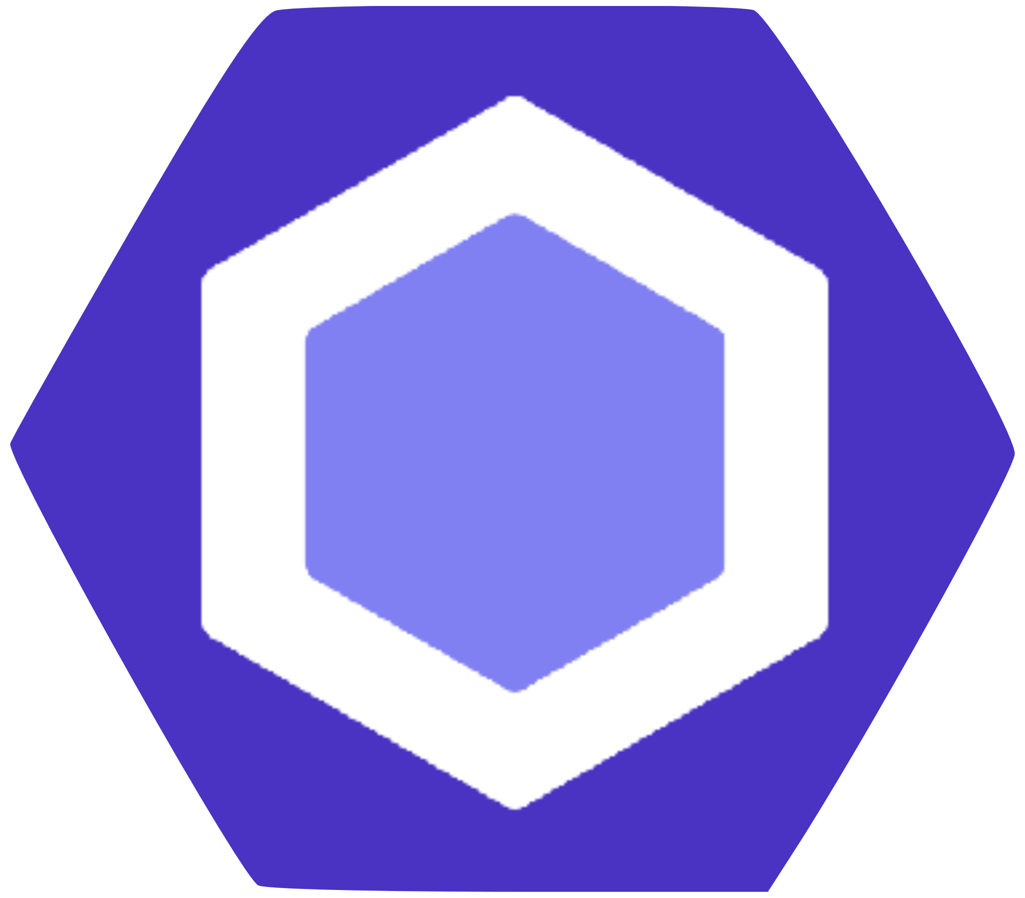
```
function payRequest(req, res) {
  // all the important pay request logic...

  // send back the request id
  res.send(requestId)

  // if sending to a known sender...
  if (senderId) {
    await putInToExperiment(senderId)
  }
}
```

STATIC ANALYSIS

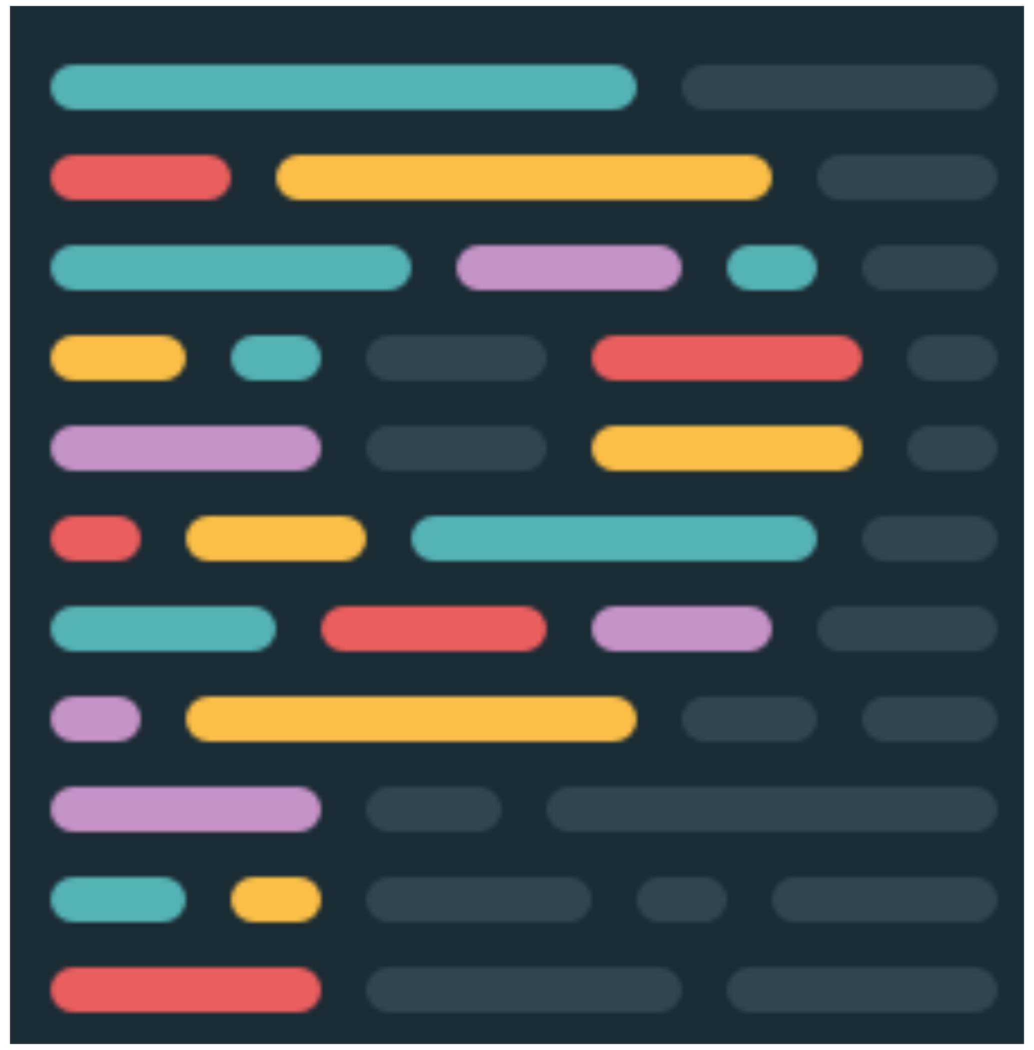
```
async function optInToExperiment(senderId) {
    let encrtypedId = decrypt(senderId)
    expService.addUser(encrtypedId, function(err, res) {
        if (err) return logger.log(res) ✗
        // success
    })
}
```

STATIC ANALYZERS AND FORMATTERS ARE PRETTY COOL

FlowType



ESLint



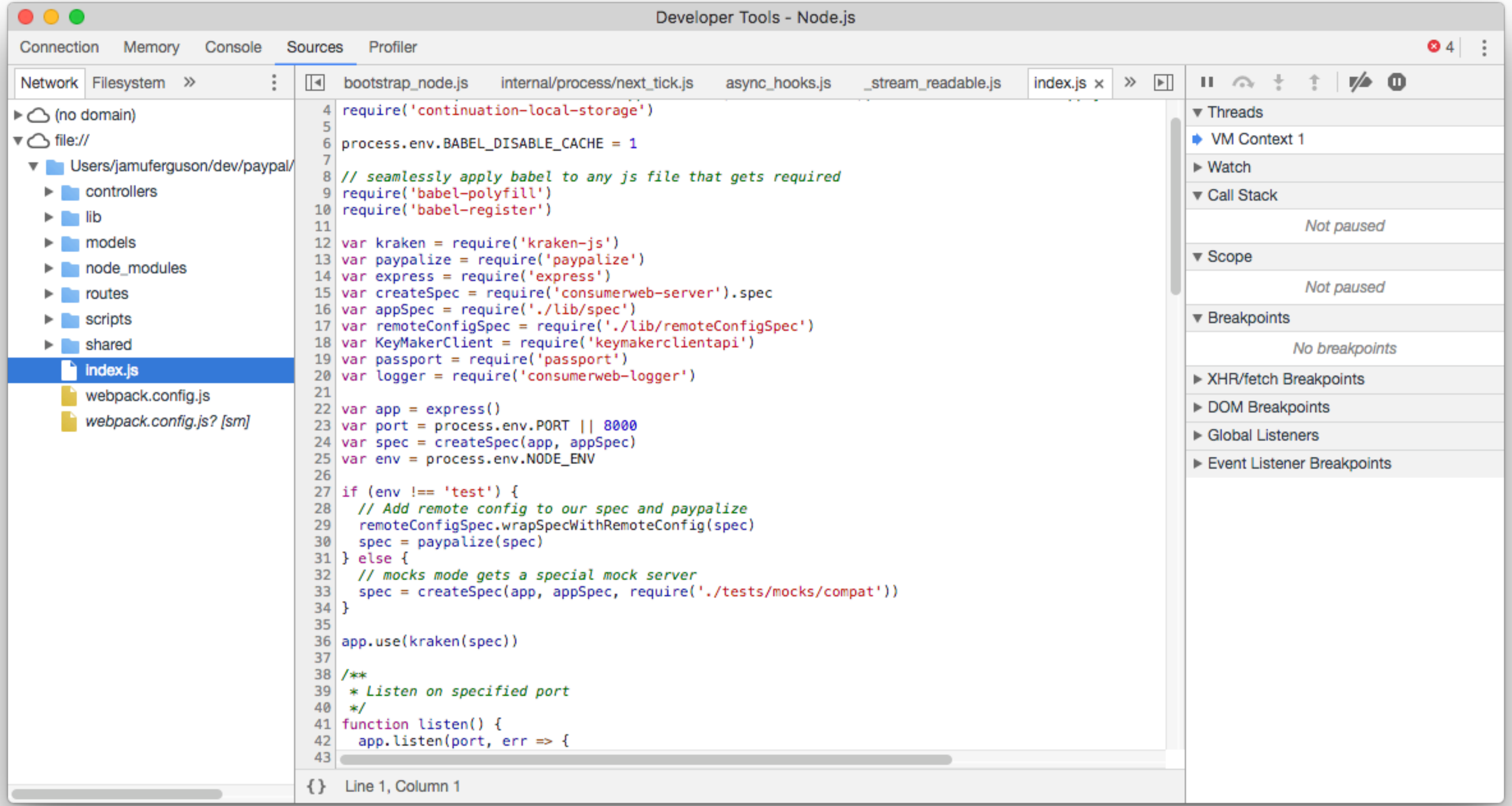
Prettier

Unfortunately, we still get bugs from time to time

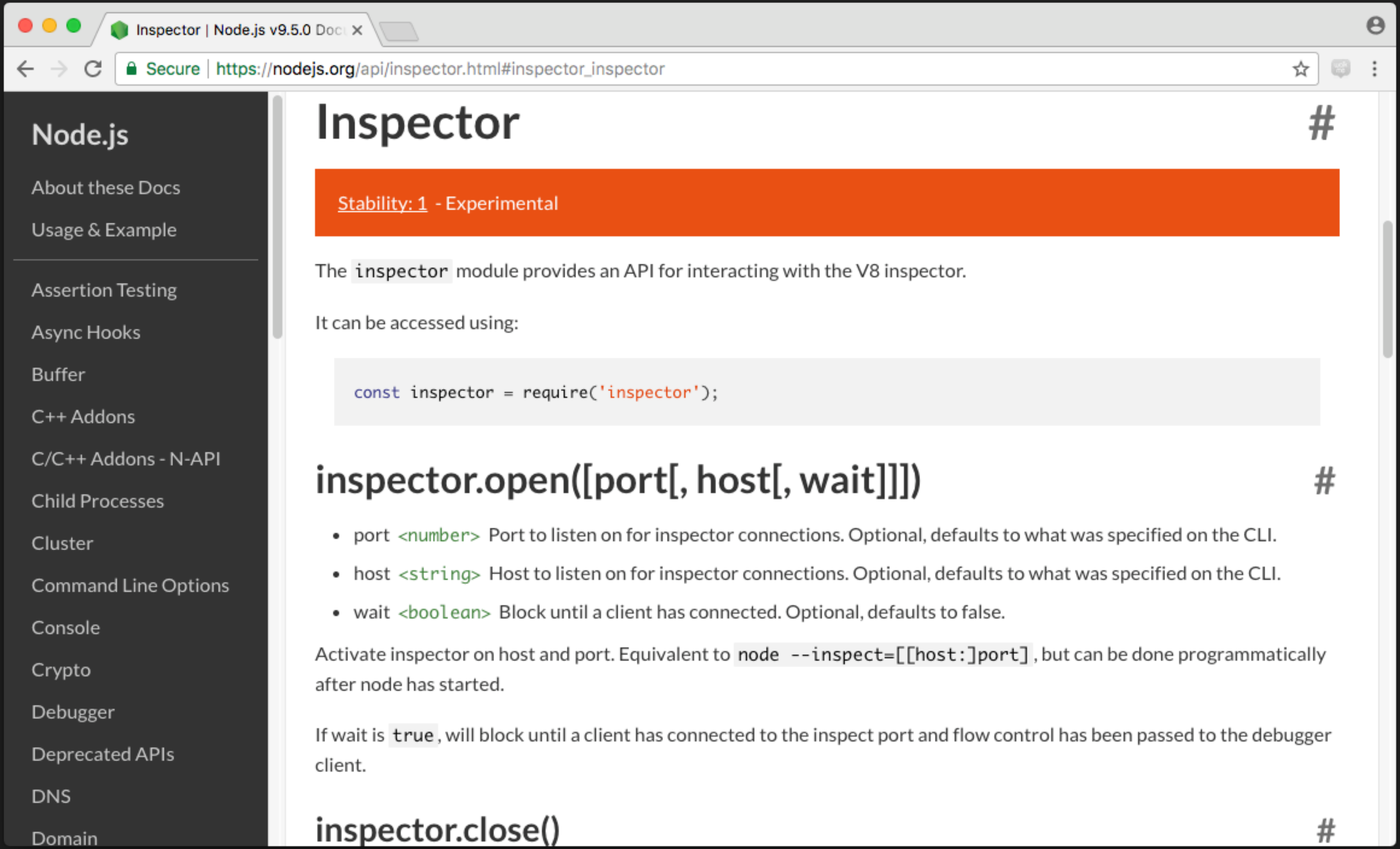
DEBUGGING USING THE

INSPECTOR MODULE

THE BUILT-IN INSPECTOR MODULE



```
require('continuation-local-storage')

process.env.BABEL_DISABLE_CACHE = 1

// seamlessly apply babel to any js file that gets required
require('babel-polyfill')
require('babel-register')

var kraken = require('kraken-js')
var paypalize = require('paypalize')
var express = require('express')
var createSpec = require('consumerweb-server').spec
var appSpec = require('./lib/spec')
var remoteConfigSpec = require('./lib/remoteConfigSpec')
var KeyMakerClient = require('keymakerclientapi')
var passport = require('passport')
var logger = require('consumerweb-logger')

var app = express()
var port = process.env.PORT || 8000
var spec = createSpec(app, appSpec)
var env = process.env.NODE_ENV

if (env !== 'test') {
  // Add remote config to our spec and paypalize
  remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
  spec = paypalize(spec)
} else {
  // mocks mode gets a special mock server
  spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
}

app.use(kraken(spec))

/**
 * Listen on specified port
 */
function listen() {
  app.listen(port, err => {
```



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]]])

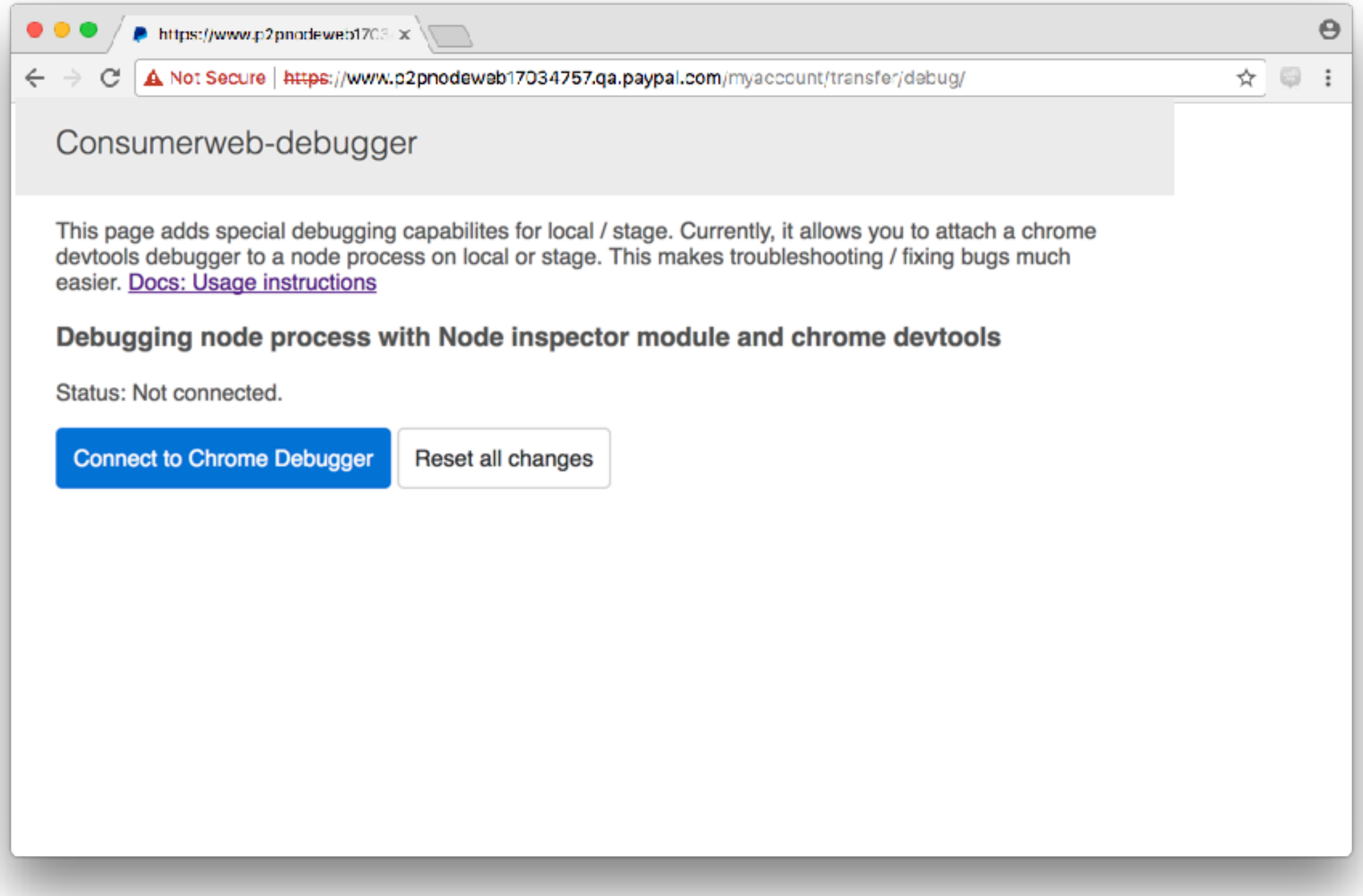
- port <number> Port to listen on for inspector connections. Optional, defaults to what was specified on the CLI.
- host <string> Host to listen on for inspector connections. Optional, defaults to what was specified on the CLI.
- wait <boolean> Block until a client has connected. Optional, defaults to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

Activate inspector on host and port. Equivalent to `node --inspect=[[host:]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



https://www.p2pnodeweb17034757.qa.paypal.com/myaccount/transfer/debug/

Consumerweb-debugger

This page adds special debugging capabilites for local / stage. Currently, it allows you to attach a chrome devtools debugger to a node process on local or stage. This makes troubleshooting / fixing bugs much easier. Docs: Usage instructions

Debugging node process with Node inspector module and chrome devtools

Status: Not connected.

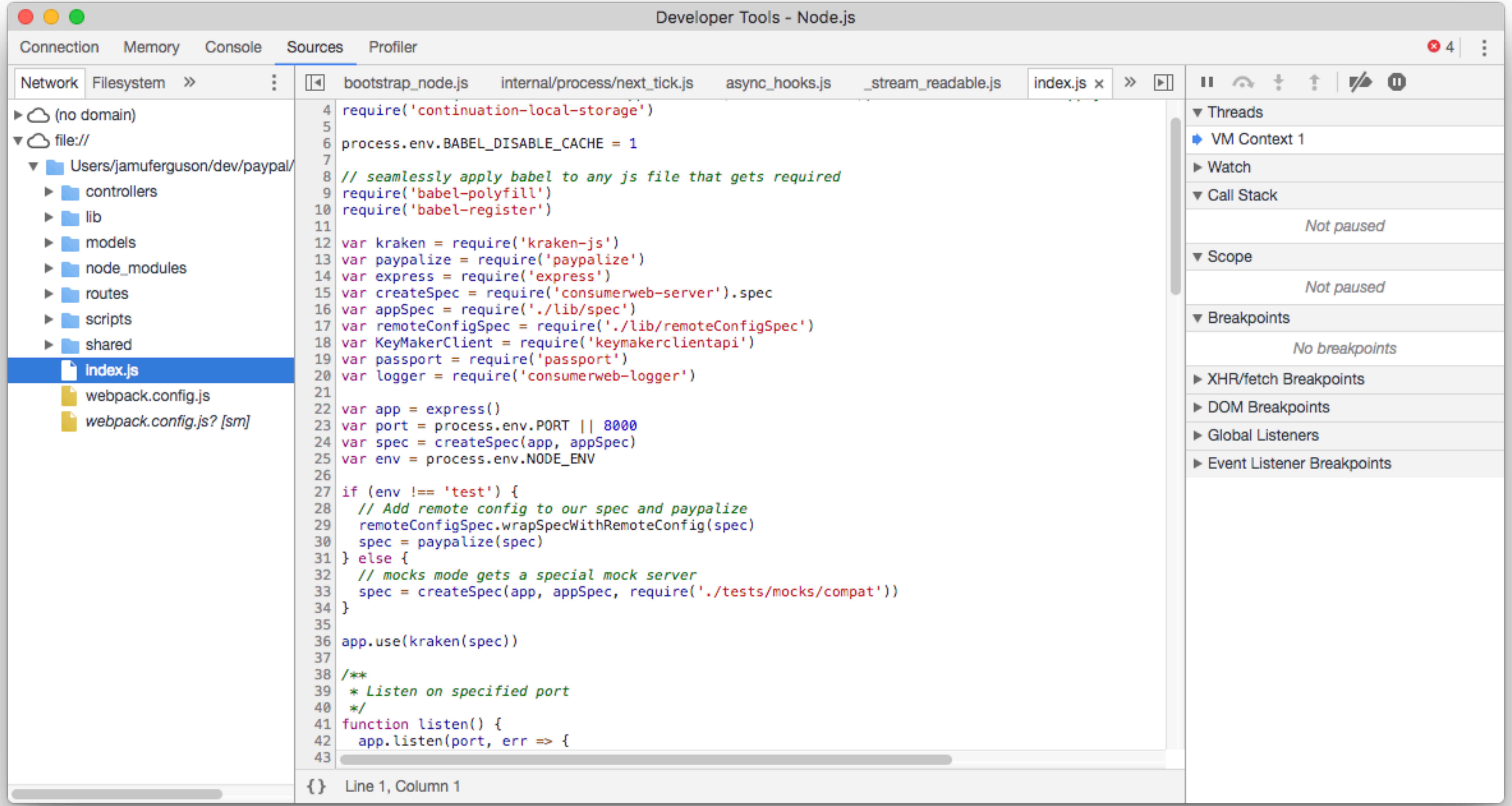
TURN IT ON

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

TURN IT OFF

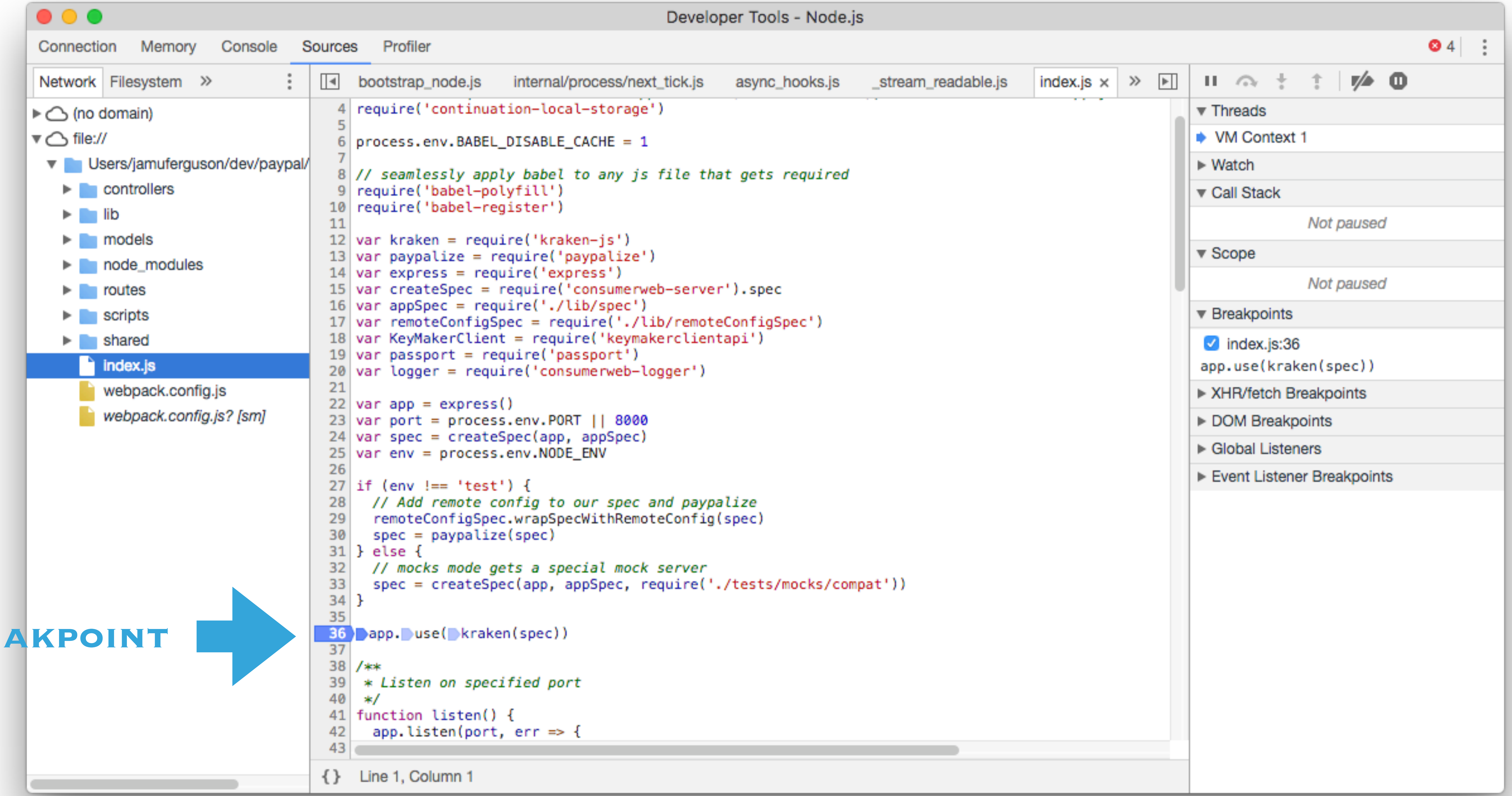
```
inspector.close()
```

THE BUILT-IN INSPECTOR MODULE



```
require('continuation-local-storage')

process.env.BABEL_DISABLE_CACHE = 1

// seamlessly apply babel to any js file that gets required
require('babel-polyfill')
require('babel-register')

var kraken = require('kraken-js')
var paypalize = require('paypalize')
var express = require('express')
var createSpec = require('consumerweb-server').spec
var appSpec = require('./lib/spec')
var remoteConfigSpec = require('./lib/remoteConfigSpec')
var KeyMakerClient = require('keymakerclientapi')
var passport = require('passport')
var logger = require('consumerweb-logger')

var app = express()
var port = process.env.PORT || 8000
var spec = createSpec(app, appSpec)
var env = process.env.NODE_ENV

if (env !== 'test') {
  // Add remote config to our spec and paypalize
  remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
  spec = paypalize(spec)
} else {
  // mocks mode gets a special mock server
  spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
}

app.use(kraken(spec))

/**
 * Listen on specified port
 */
function listen() {
  app.listen(port, err => {
```

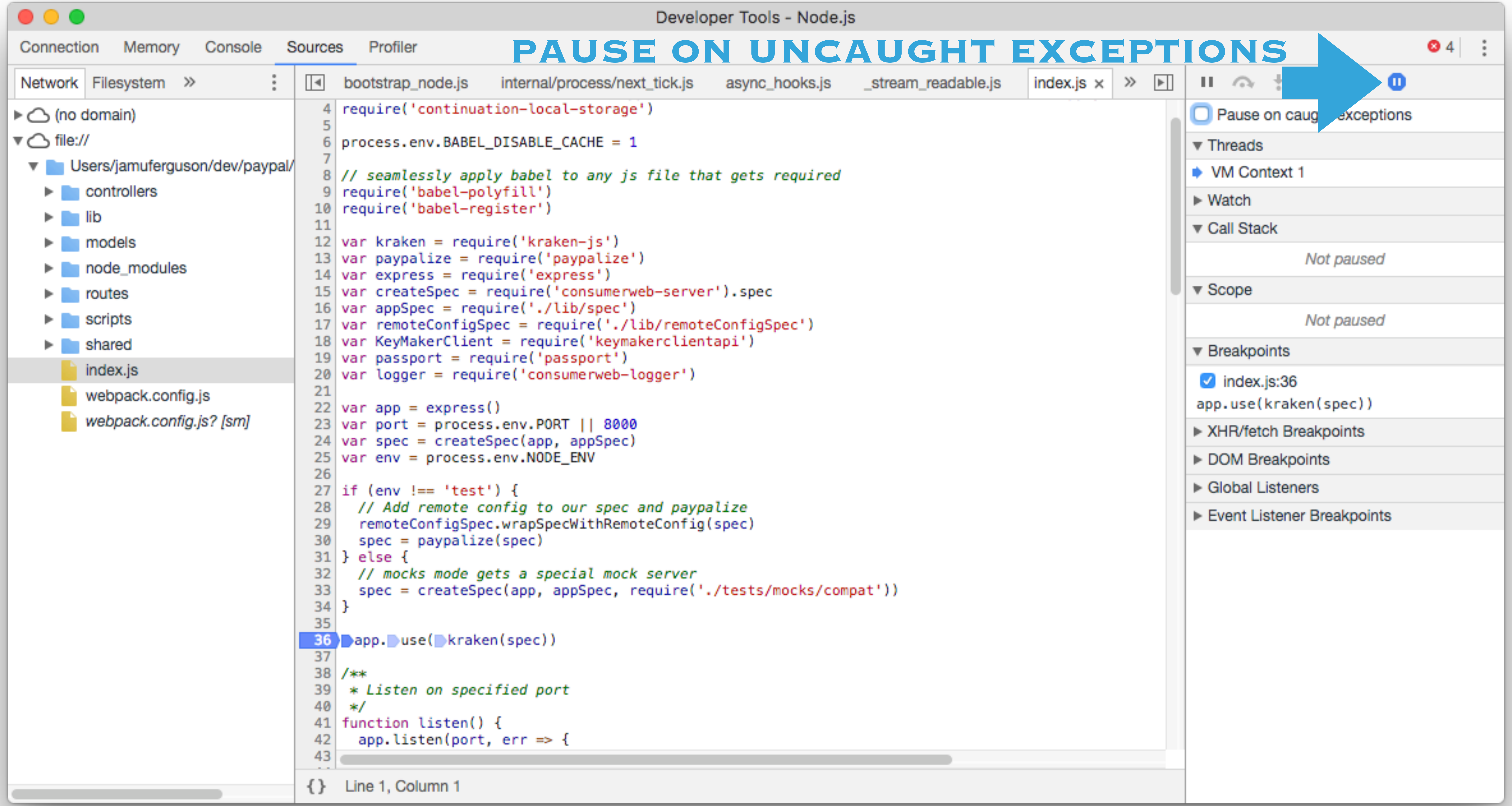
THE BUILT-IN INSPECTOR MODULE



```
require('continuation-local-storage')

process.env.BABEL_DISABLE_CACHE = 1

// seamlessly apply babel to any js file that gets required
require('babel-polyfill')
require('babel-register')

var kraken = require('kraken-js')
var paypalize = require('paypalize')
var express = require('express')
var createSpec = require('consumerweb-server').spec
var appSpec = require('./lib/spec')
var remoteConfigSpec = require('./lib/remoteConfigSpec')
var KeyMakerClient = require('keymakerclientapi')
var passport = require('passport')
var logger = require('consumerweb-logger')

var app = express()
var port = process.env.PORT || 8000
var spec = createSpec(app, appSpec)
var env = process.env.NODE_ENV

if (env !== 'test') {
  // Add remote config to our spec and paypalize
  remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
  spec = paypalize(spec)
} else {
  // mocks mode gets a special mock server
  spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
}

app.use(kraken(spec))

/**
 * Listen on specified port
 */
function listen() {
  app.listen(port, err => {
```

Set Breakpoint

THE BUILT-IN INSPECTOR MODULE



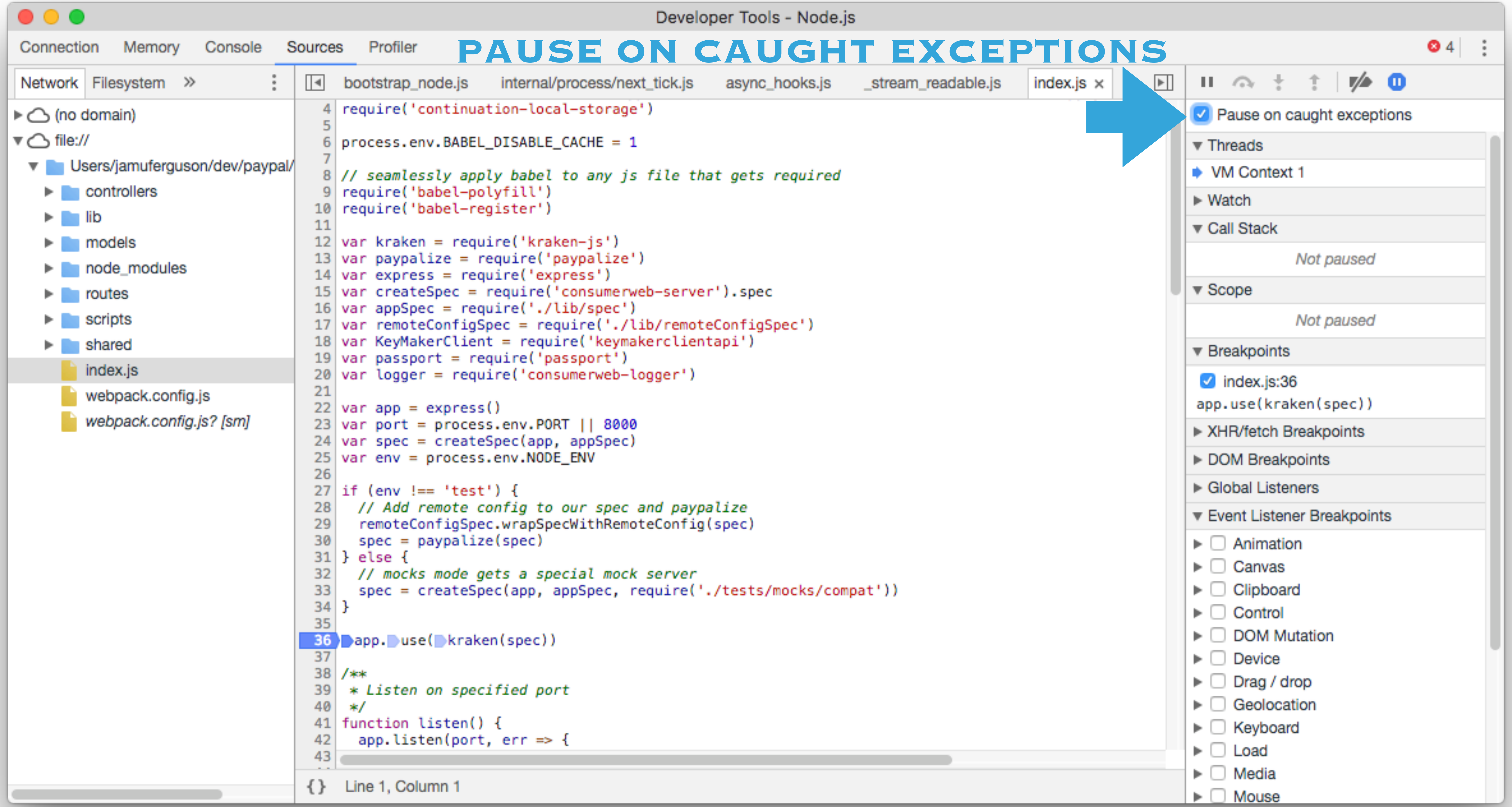
PAUSE ON UNCAUGHT EXCEPTIONS

☐ Pause on caug exceptions

☑ index.js:36
app.use(kraken(spec))

```
require('continuation-local-storage')

process.env.BABEL_DISABLE_CACHE = 1

// seamlessly apply babel to any js file that gets required
require('babel-polyfill')
require('babel-register')

var kraken = require('kraken-js')
var paypalize = require('paypalize')
var express = require('express')
var createSpec = require('consumerweb-server').spec
var appSpec = require('./lib/spec')
var remoteConfigSpec = require('./lib/remoteConfigSpec')
var KeyMakerClient = require('keymakerclientapi')
var passport = require('passport')
var logger = require('consumerweb-logger')

var app = express()
var port = process.env.PORT || 8000
var spec = createSpec(app, appSpec)
var env = process.env.NODE_ENV

if (env !== 'test') {
  // Add remote config to our spec and paypalize
  remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
  spec = paypalize(spec)
} else {
  // mocks mode gets a special mock server
  spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
}

app.use(kraken(spec))

/**
 * Listen on specified port
 */
function listen() {
  app.listen(port, err => {
```

THE BUILT-IN INSPECTOR MODULE



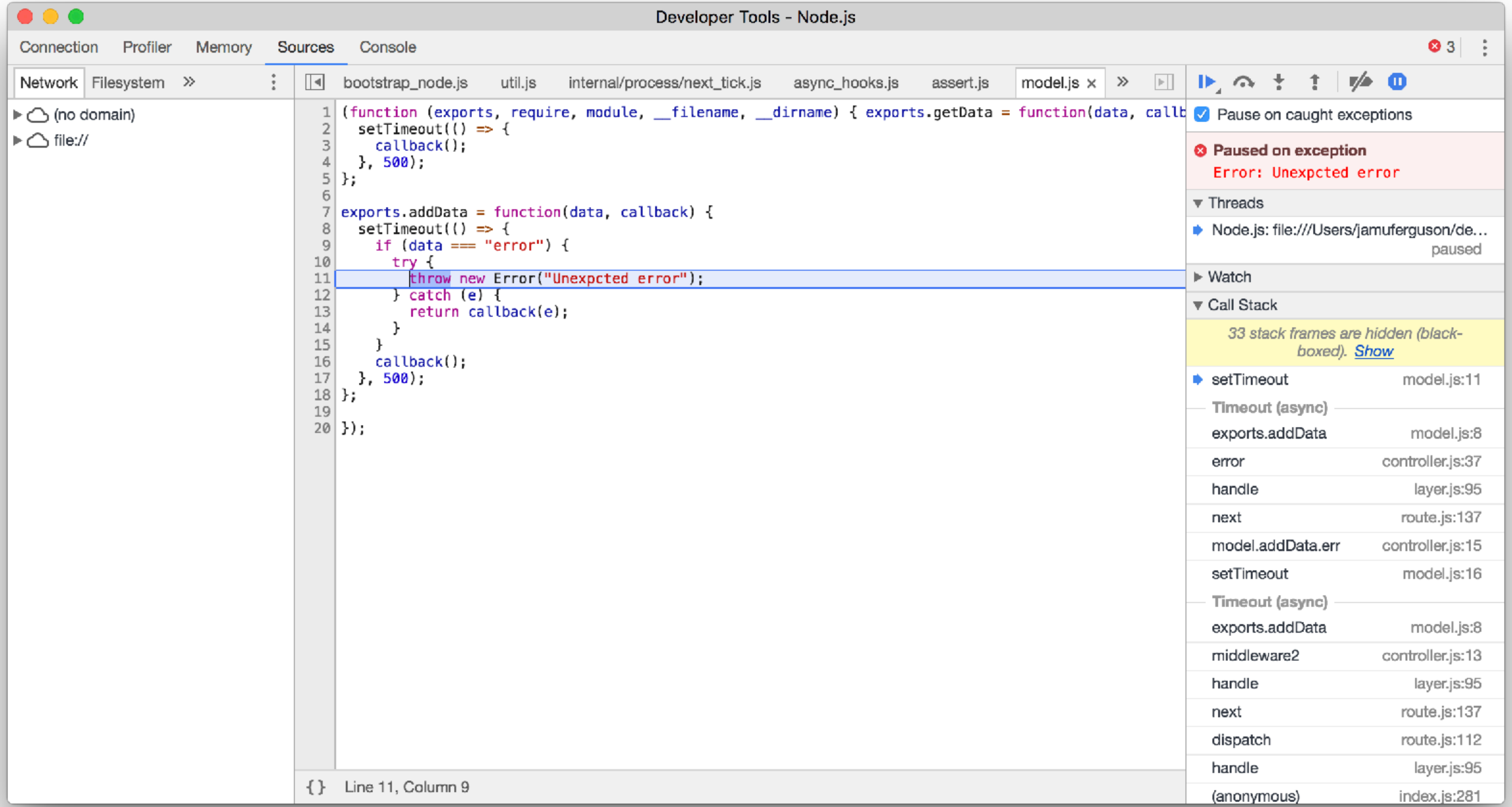
PAUSE ON CAUGHT EXCEPTIONS

☑ Pause on caught exceptions

☑ index.js:36
app.use(kraken(spec))

```
require('continuation-local-storage')

process.env.BABEL_DISABLE_CACHE = 1

// seamlessly apply babel to any js file that gets required
require('babel-polyfill')
require('babel-register')

var kraken = require('kraken-js')
var paypalize = require('paypalize')
var express = require('express')
var createSpec = require('consumerweb-server').spec
var appSpec = require('./lib/spec')
var remoteConfigSpec = require('./lib/remoteConfigSpec')
var KeyMakerClient = require('keymakerclientapi')
var passport = require('passport')
var logger = require('consumerweb-logger')

var app = express()
var port = process.env.PORT || 8000
var spec = createSpec(app, appSpec)
var env = process.env.NODE_ENV

if (env !== 'test') {
  // Add remote config to our spec and paypalize
  remoteConfigSpec.wrapSpecWithRemoteConfig(spec)
  spec = paypalize(spec)
} else {
  // mocks mode gets a special mock server
  spec = createSpec(app, appSpec, require('./tests/mocks/compat'))
}

app.use(kraken(spec))

/**
 * Listen on specified port
 */
function listen() {
  app.listen(port, err => {
```

THE BUILT-IN INSPECTOR MODULE



```
(function (exports, require, module, __filename, __dirname) { exports.getData = function(data, callb
  setTimeout(() => {
    callback();
  }, 500);
};

exports.addData = function(data, callback) {
  setTimeout(() => {
    if (data === "error") {
      try {
        throw new Error("Unexpcted error");
      } catch (e) {
        return callback(e);
      }
    }
    callback();
  }, 500);
};

});
```

Line 11, Column 9

☑ Pause on caught exceptions

Paused on exception
`Error: Unexpcted error`

▼ Threads

Node.js: file:///Users/jamuferguson/de...
paused

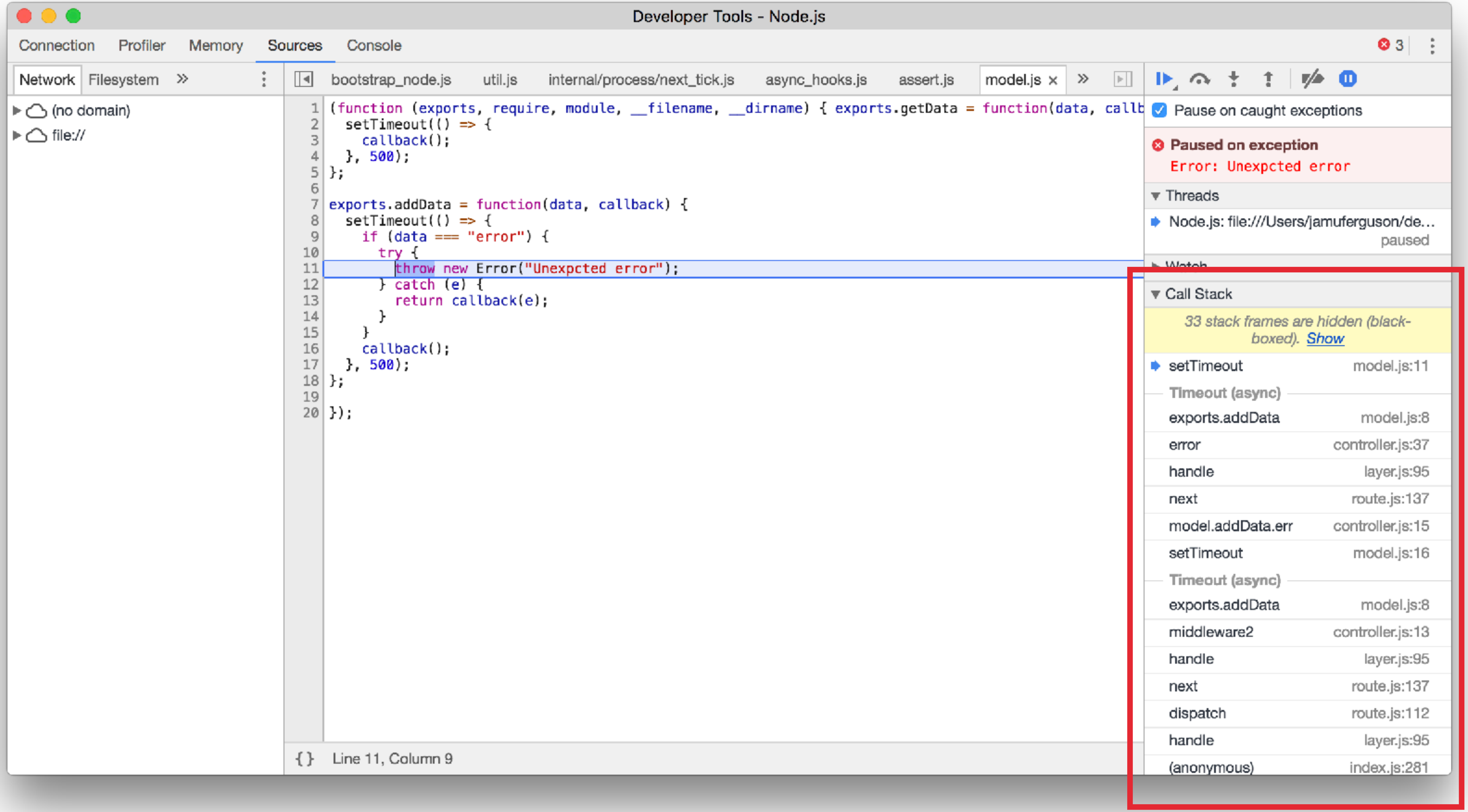
▶ Watch

▼ Call Stack

33 stack frames are hidden (black-boxed). Show

setTimeout	model.js:11
Timeout (async)	
exports.addData	model.js:8
error	controller.js:37
handle	layer.js:95
next	route.js:137
model.addData.err	controller.js:15
setTimeout	model.js:16
Timeout (async)	
exports.addData	model.js:8
middleware2	controller.js:13
handle	layer.js:95
next	route.js:137
dispatch	route.js:112
handle	layer.js:95
(anonymous)	index.js:281

THE BUILT-IN INSPECTOR MODULE



```
(function (exports, require, module, __filename, __dirname) { exports.getData = function(data, callb
  setTimeout(() => {
    callback();
  }, 500);
};

exports.addData = function(data, callback) {
  setTimeout(() => {
    if (data === "error") {
      try {
        throw new Error("Unexpcted error");
      } catch (e) {
        return callback(e);
      }
    }
    callback();
  }, 500);
};

});
```

Line 11, Column 9

☑ Pause on caught exceptions

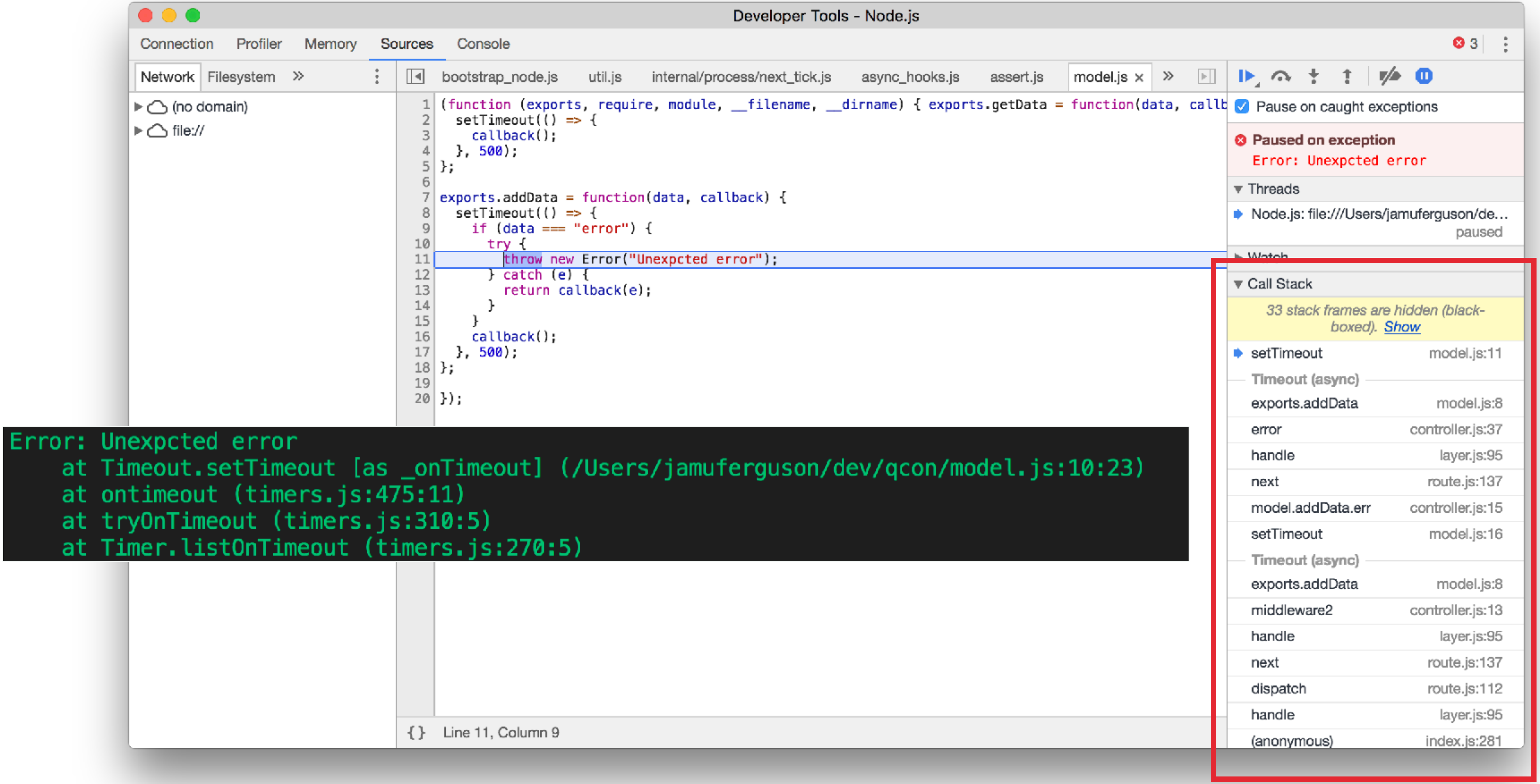
Paused on exception
`Error: Unexpcted error`

Node.js: file:///Users/jamuferguson/de... paused

33 stack frames are hidden (black-boxed). Show

```
setTimeout            model.js:11
Timeout (async)
exports.addData       model.js:8
error                 controller.js:37
handle                layer.js:95
next                  route.js:137
model.addData.err     controller.js:15
setTimeout            model.js:16
Timeout (async)
exports.addData       model.js:8
middleware2           controller.js:13
handle                layer.js:95
next                  route.js:137
dispatch              route.js:112
handle                layer.js:95
(anonymous)           index.js:281
```

THE BUILT-IN INSPECTOR MODULE



```
(function (exports, require, module, __filename, __dirname) { exports.getData = function(data, callb
  setTimeout(() => {
    callback();
  }, 500);
};

exports.addData = function(data, callback) {
  setTimeout(() => {
    if (data === "error") {
      try {
        throw new Error("Unexpcted error");
      } catch (e) {
        return callback(e);
      }
    }
    callback();
  }, 500);
};

});
```

Line 11, Column 9

☑ Pause on caught exceptions

Paused on exception

Error: Unexpcted error

Threads

Node.js: file:///Users/jamuferguson/de... paused

Watch

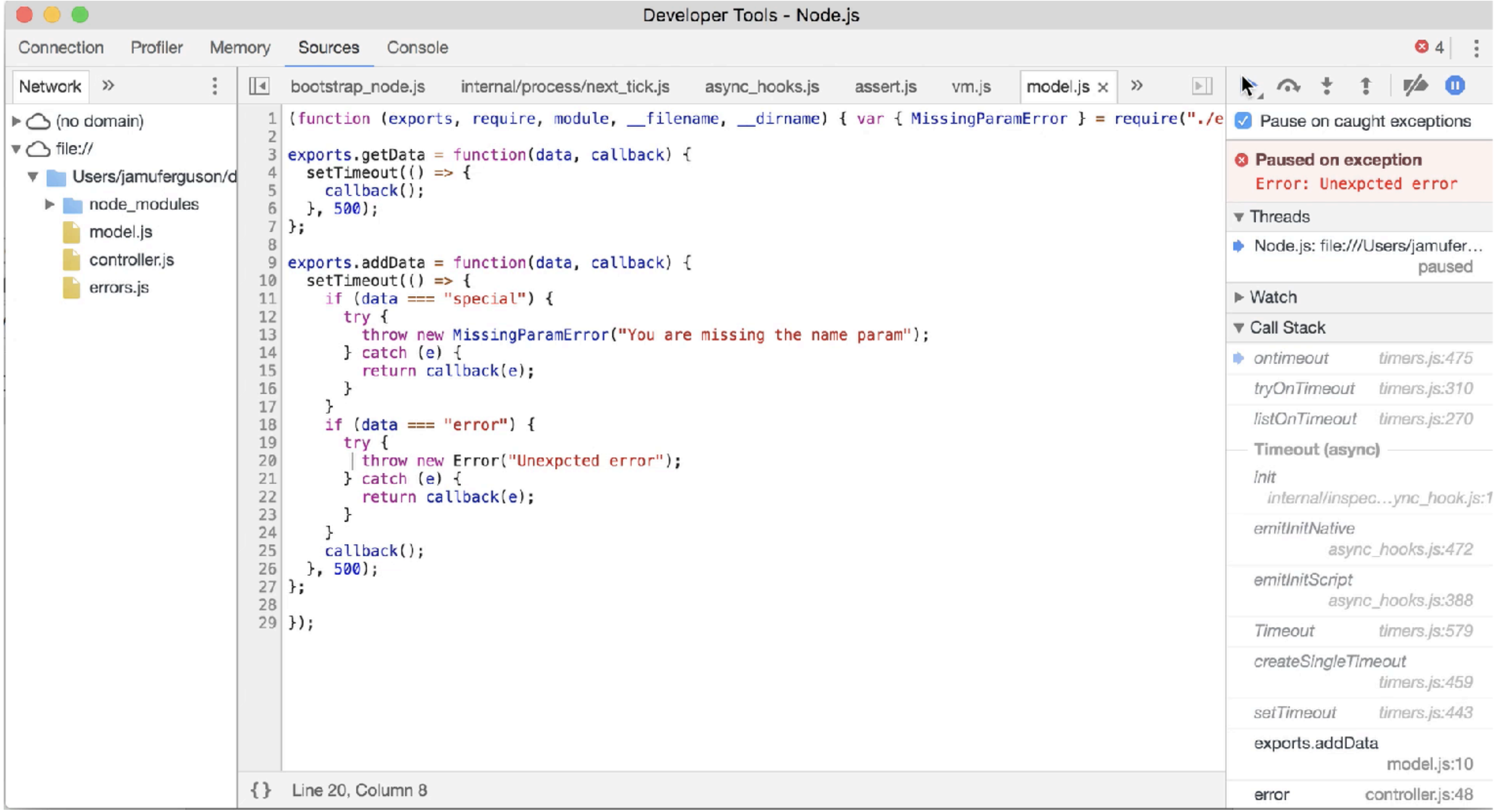
Call Stack

33 stack frames are hidden (black-boxed). Show

setTimeout	model.js:11
Timeout (async)	
exports.addData	model.js:8
error	controller.js:37
handle	layer.js:95
next	route.js:137
model.addData.err	controller.js:15
setTimeout	model.js:16
Timeout (async)	
exports.addData	model.js:8
middleware2	controller.js:13
handle	layer.js:95
next	route.js:137
dispatch	route.js:112
handle	layer.js:95
(anonymous)	index.js:281

```
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)
```

THE BUILT-IN INSPECTOR MODULE



```
(function (exports, require, module, __filename, __dirname) { var { MissingParamError } = require("./e

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) {
        return callback(e);
      }
    }
    if (data === "error") {
      try {
        throw new Error("Unexpcted error");
      } catch (e) {
        return callback(e);
      }
    }
    callback();
  }, 500);
};

});
```

{ } Line 20, Column 8

☑ Pause on caught exceptions

Paused on exception
`Error: Unexpcted error`

▼ Threads

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

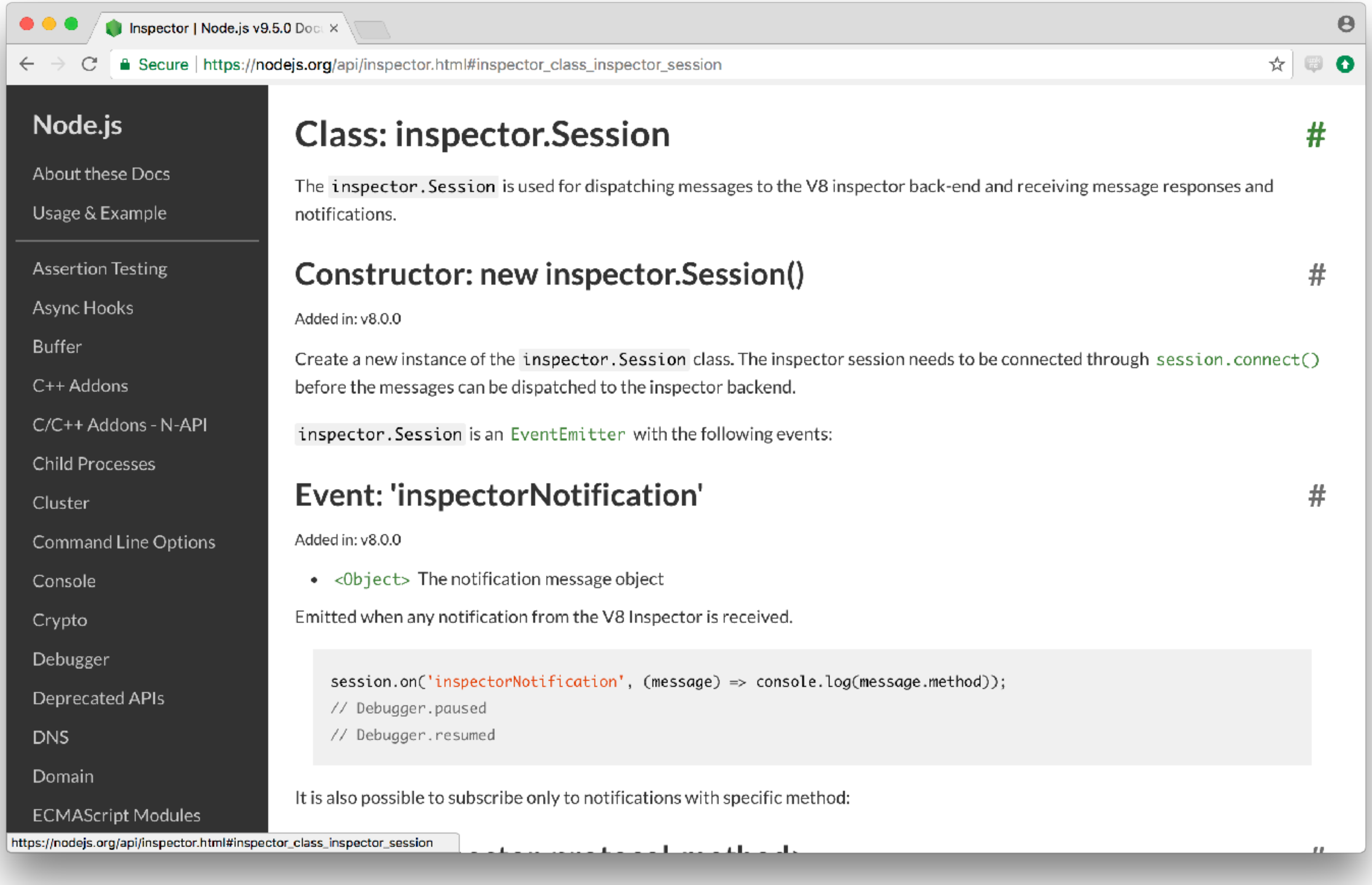
▶ Watch

▼ 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
```

Don’t try this in production

THE BUILT-IN INSPECTOR MODULE



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

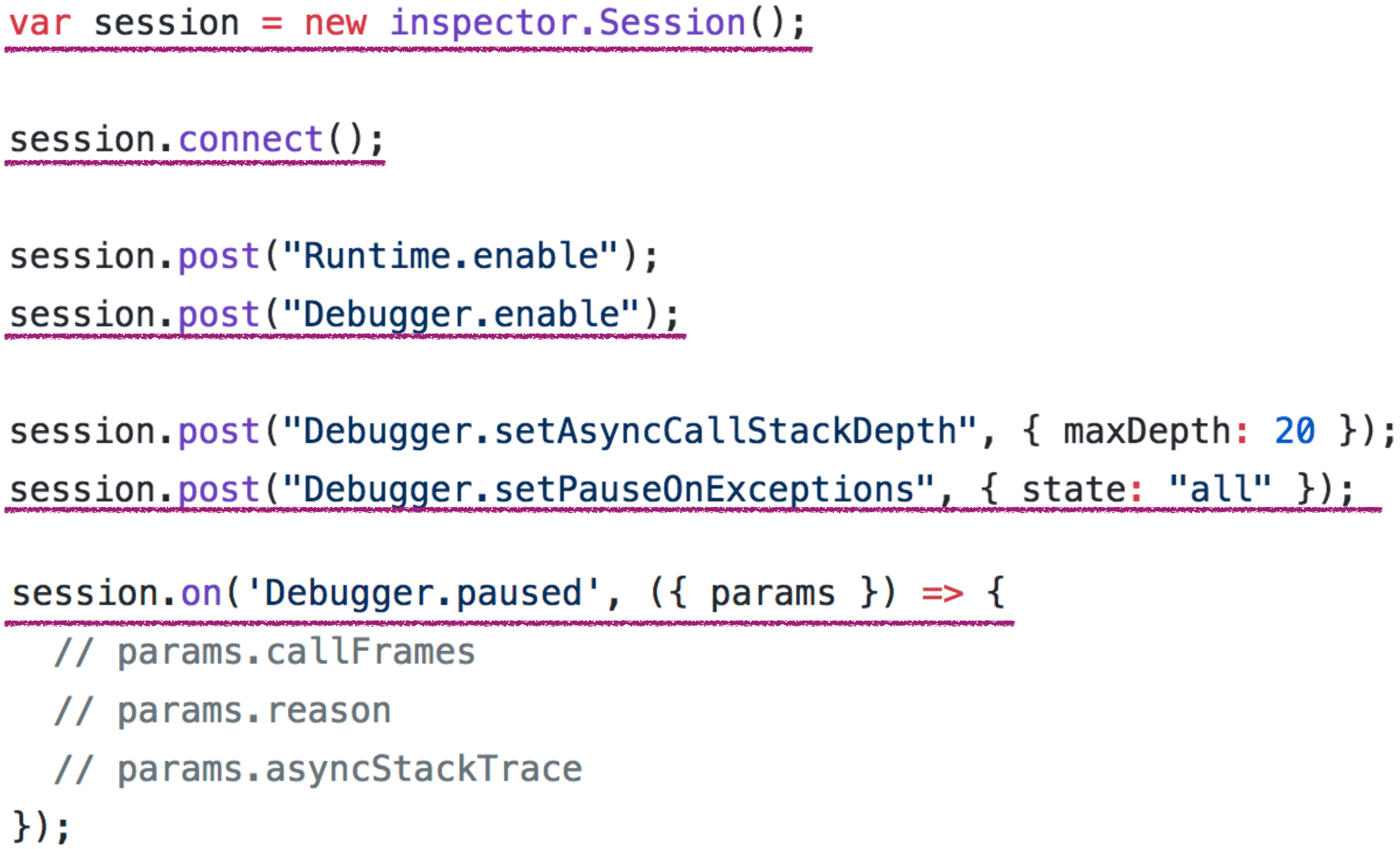
- `<Object>` The notification message object

Emitted when any notification from the V8 Inspector is received.

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

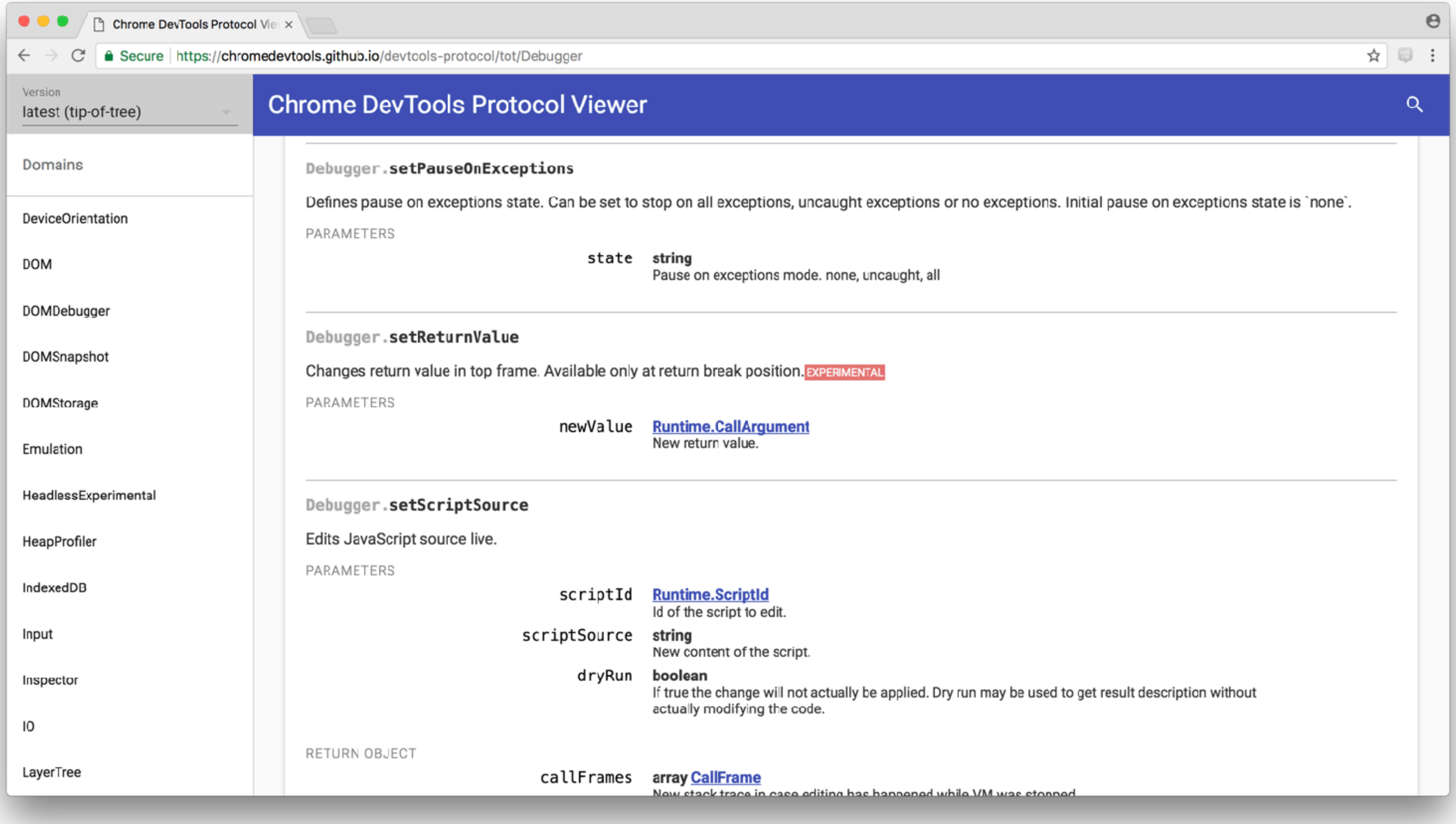
It is also possible to subscribe only to notifications with specific method:

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



Chrome DevTools Protocol Vie
Secure
https://chromedevtools.github.io/devtools-protocol/tot/Debugger

Version
latest (tip-of-tree)

Domains

DeviceOrientation
DOM
DOMDebugger
DOMSnapshot
DOMStorage
Emulation
HeadlessExperimental
HeapProfiler
IndexedDB
Input
Inspector
IO
LayerTree

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.

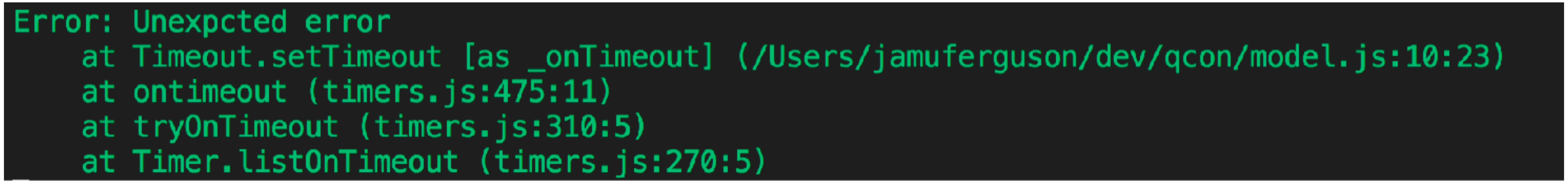
RETURN OBJECT

`callFrames` **array CallFrame**
New stack trace in case editing has happened while VM was stopped.

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

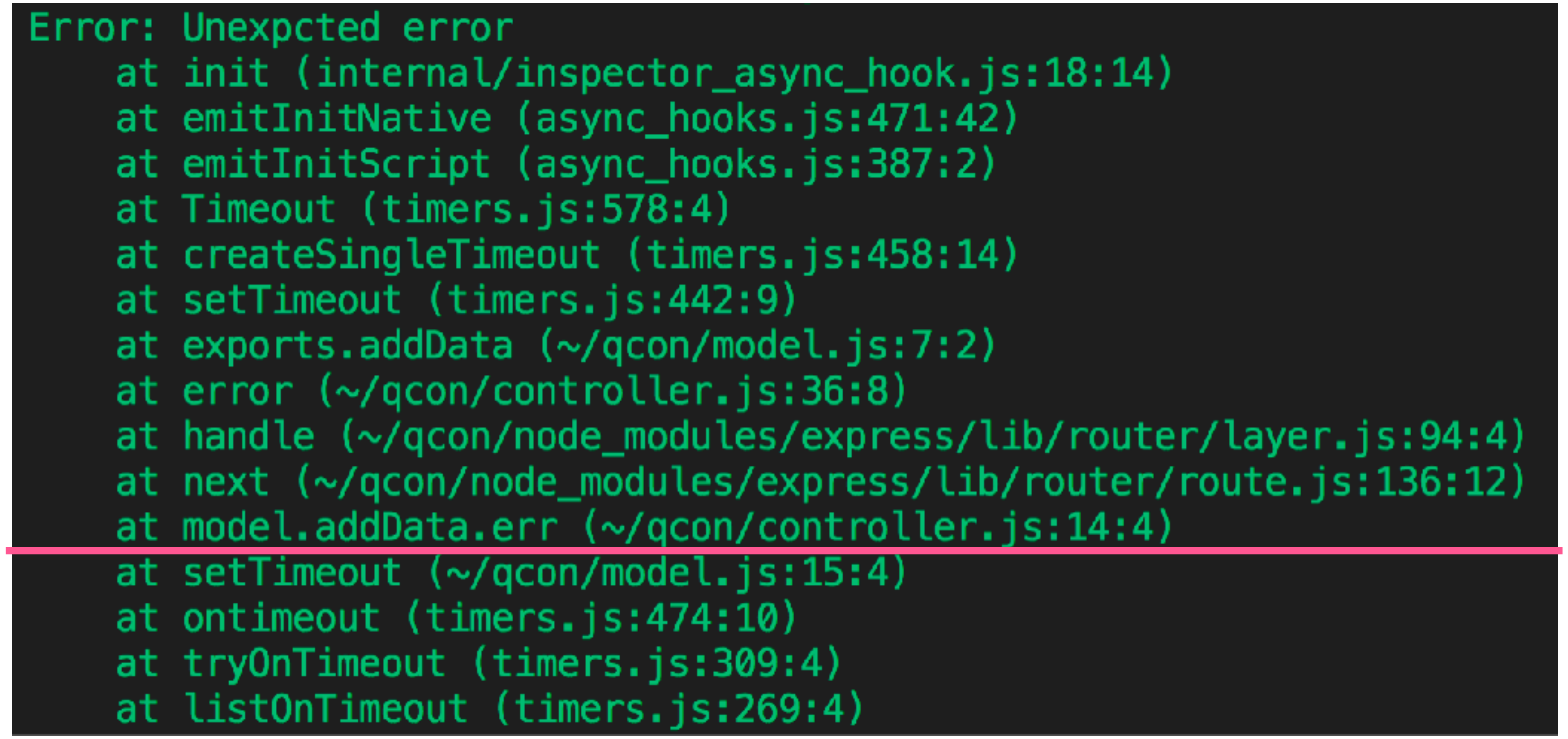
THE BUILT-IN INSPECTOR MODULE

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



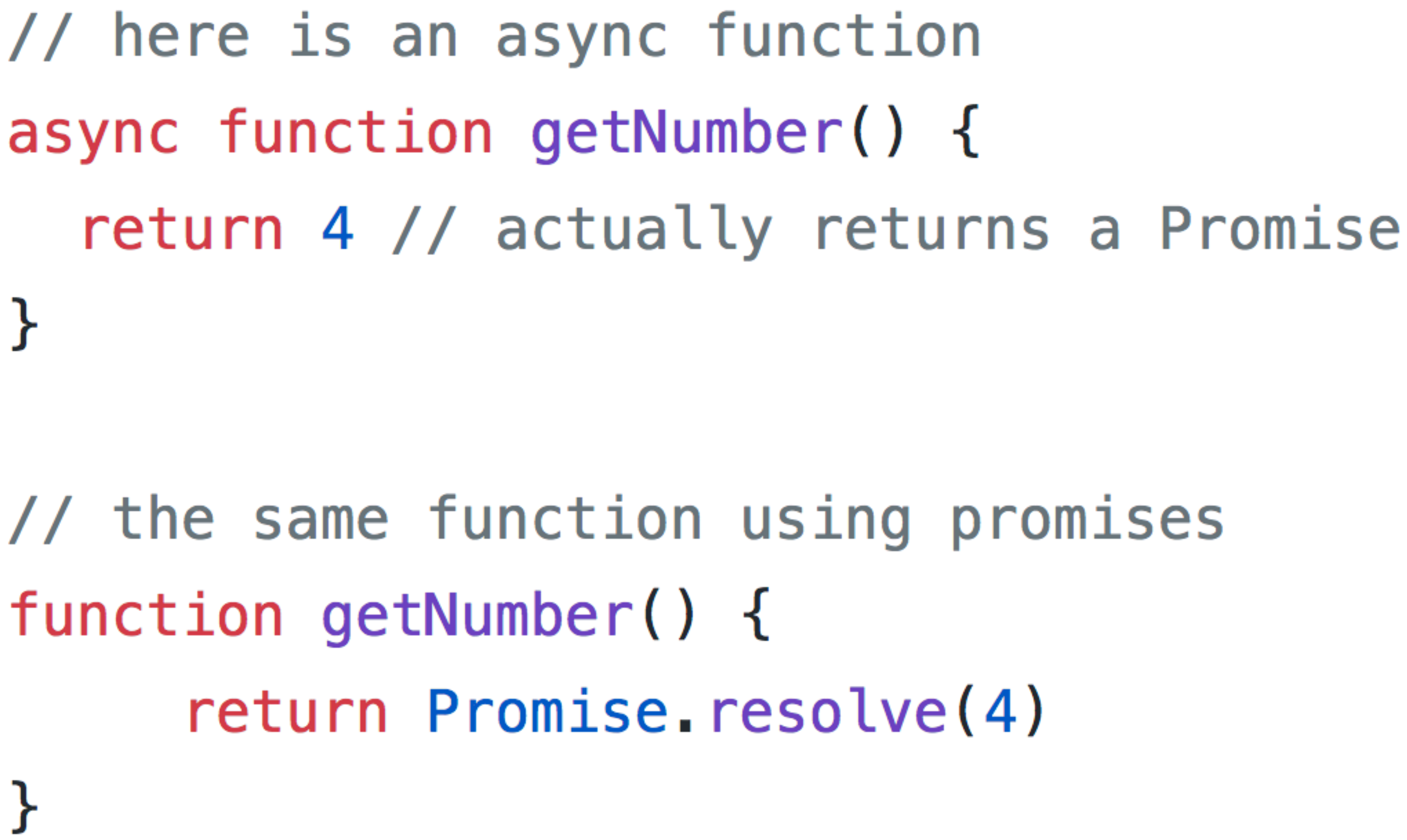
```
Error: Unexpcted error
    at init (internal/inspector_async_hook.js:18:14)
    at emitInitNative (async_hooks.js:471:42)
    at emitInitScript (async_hooks.js:387:2)
    at Timeout (timers.js:578:4)
    at createSingleTimeout (timers.js:458:14)
    at setTimeout (timers.js:442:9)
    at exports.addData (~/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

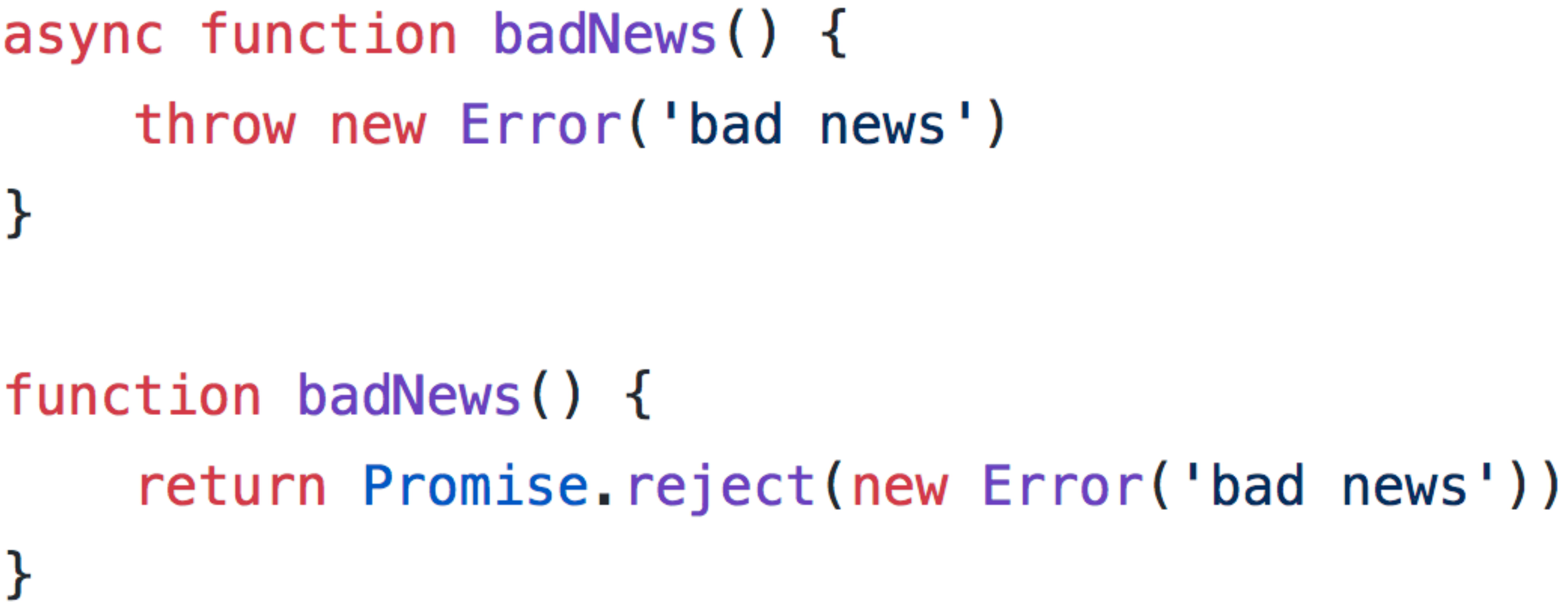
ERROR HANDLING WITH ASYNC/AWAIT



```
// 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 💡

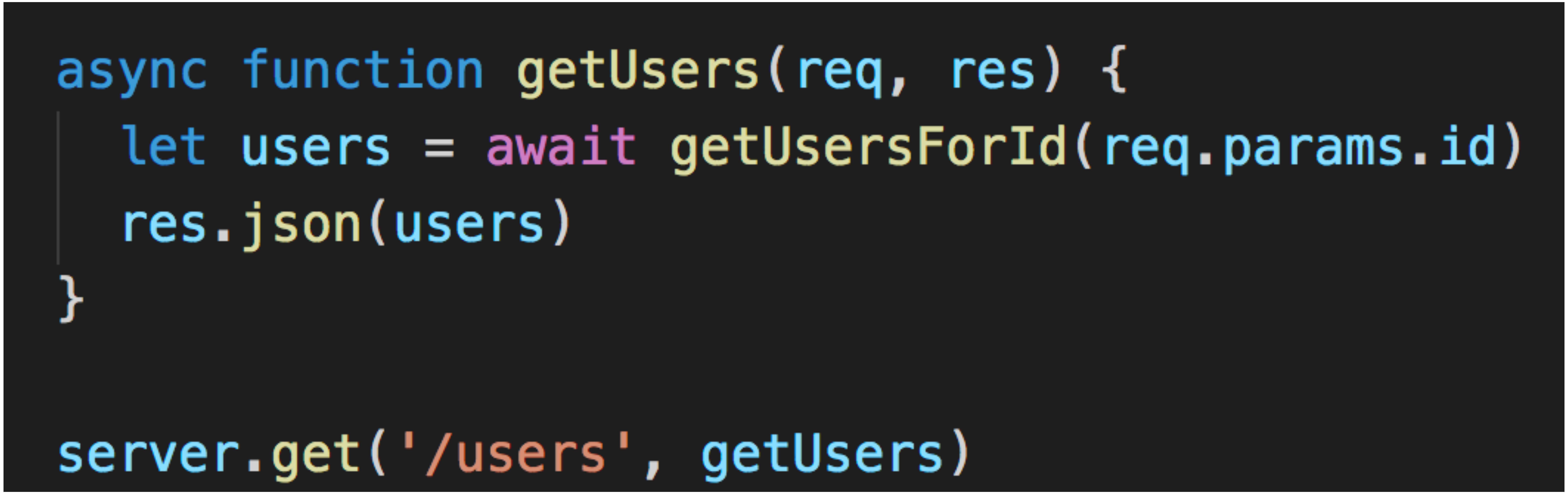
ERROR HANDLING WITH ASYNC/AWAIT



```
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

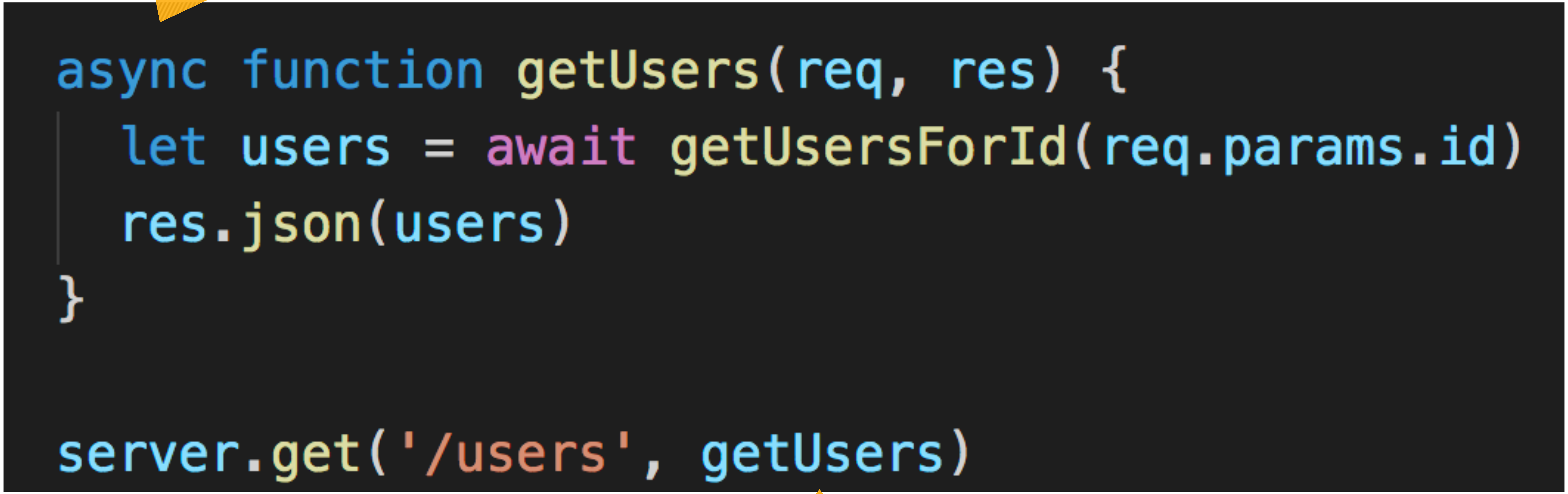


```
async function getUsers(req, res) {
    let users = await getUsersForId(req.params.id)
    res.json(users)
}

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

THIS IS PRETTY NICE

ERROR HANDLING WITH ASYNC/AWAIT

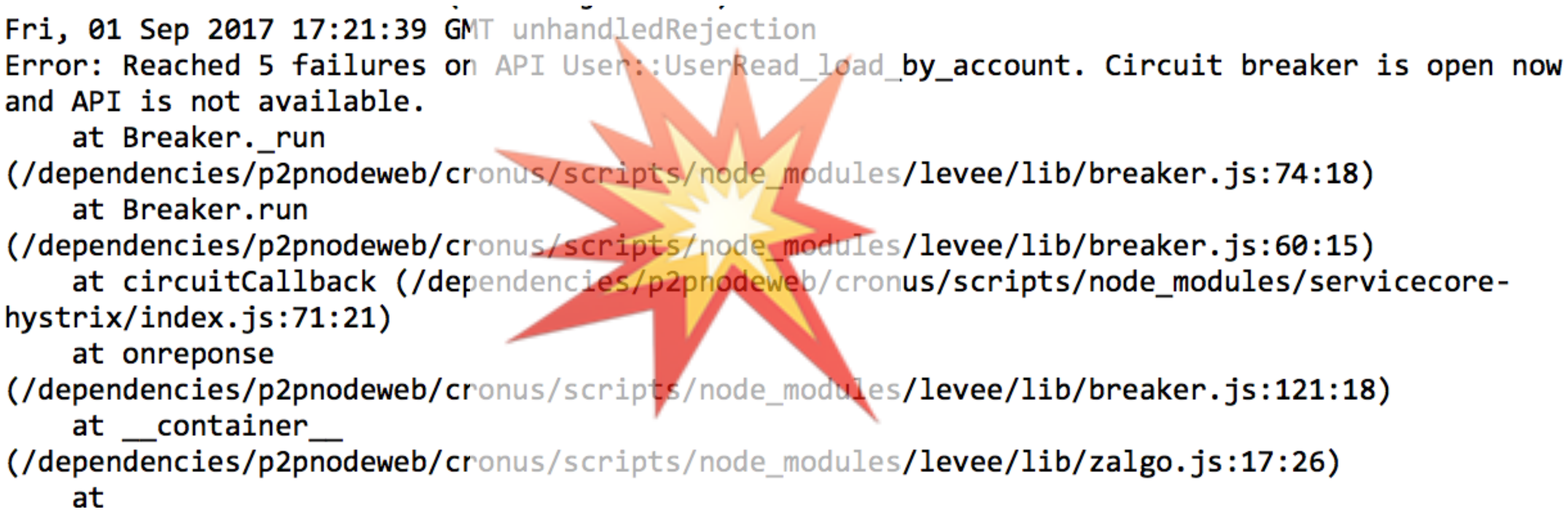
ERRORS WILL BUBBLE UP



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

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

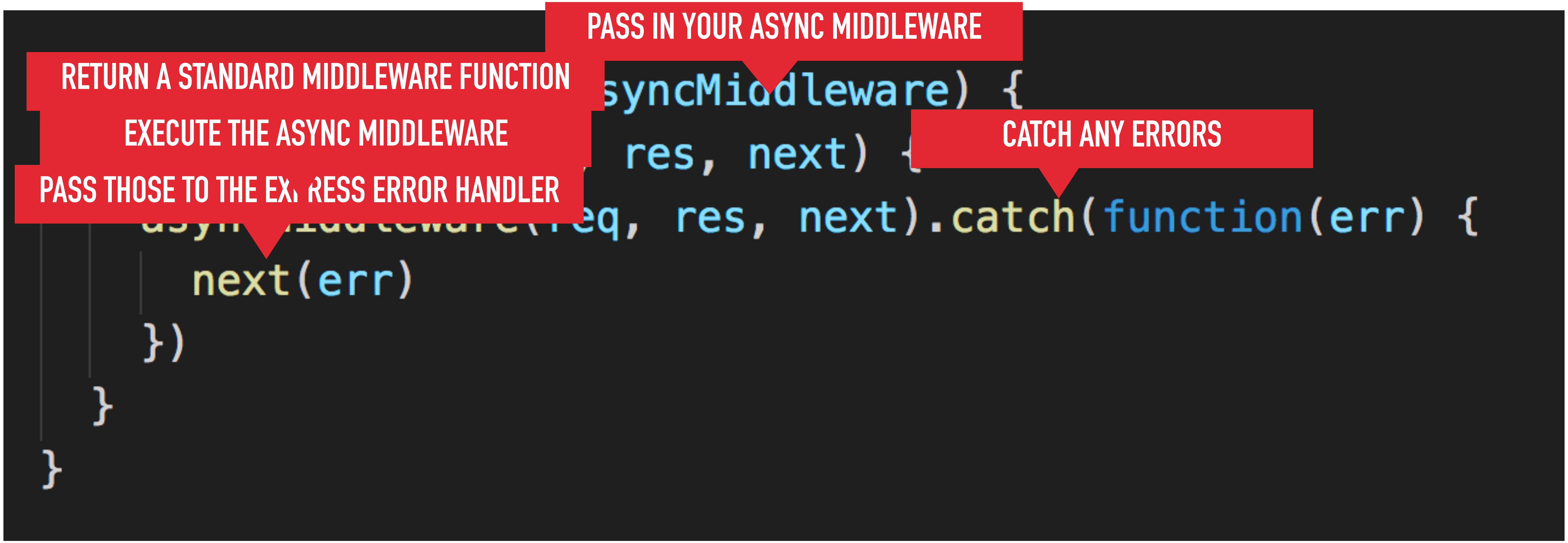
BUT WE DON'T ACTUALLY CATCH IT 💣

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 onreponse
(/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



PASS IN YOUR ASYNC MIDDLEWARE

RETURN A STANDARD MIDDLEWARE FUNCTION

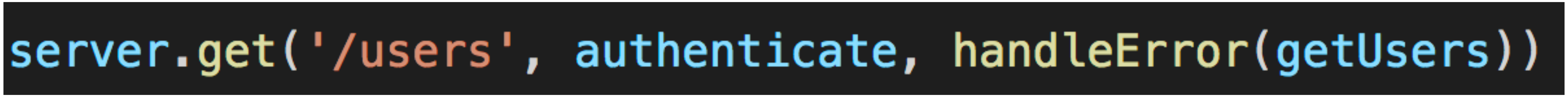
EXECUTE THE ASYNC MIDDLEWARE

CATCH ANY ERRORS

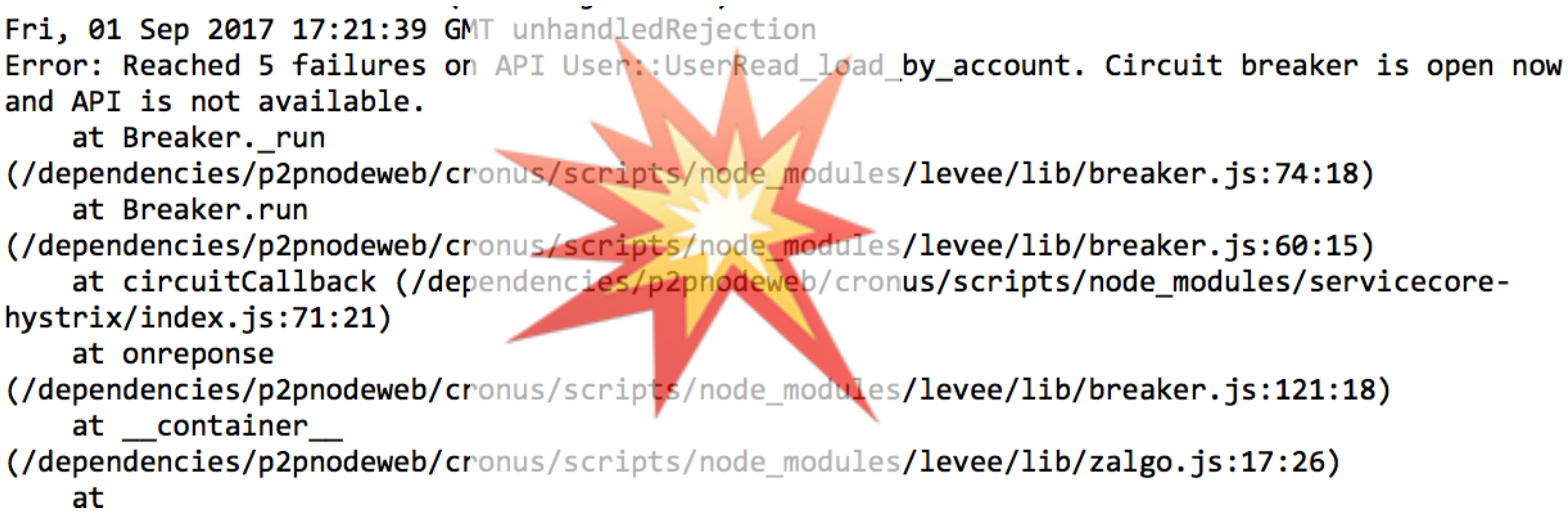
PASS THOSE TO THE EXPRESS ERROR HANDLER

```
                  syncMiddleware) {
                   res, next) {
    asyncMiddleware(req, res, next).catch(function(err) {
      next(err)
    })
  }
}
```

APPLY AS NEEDED



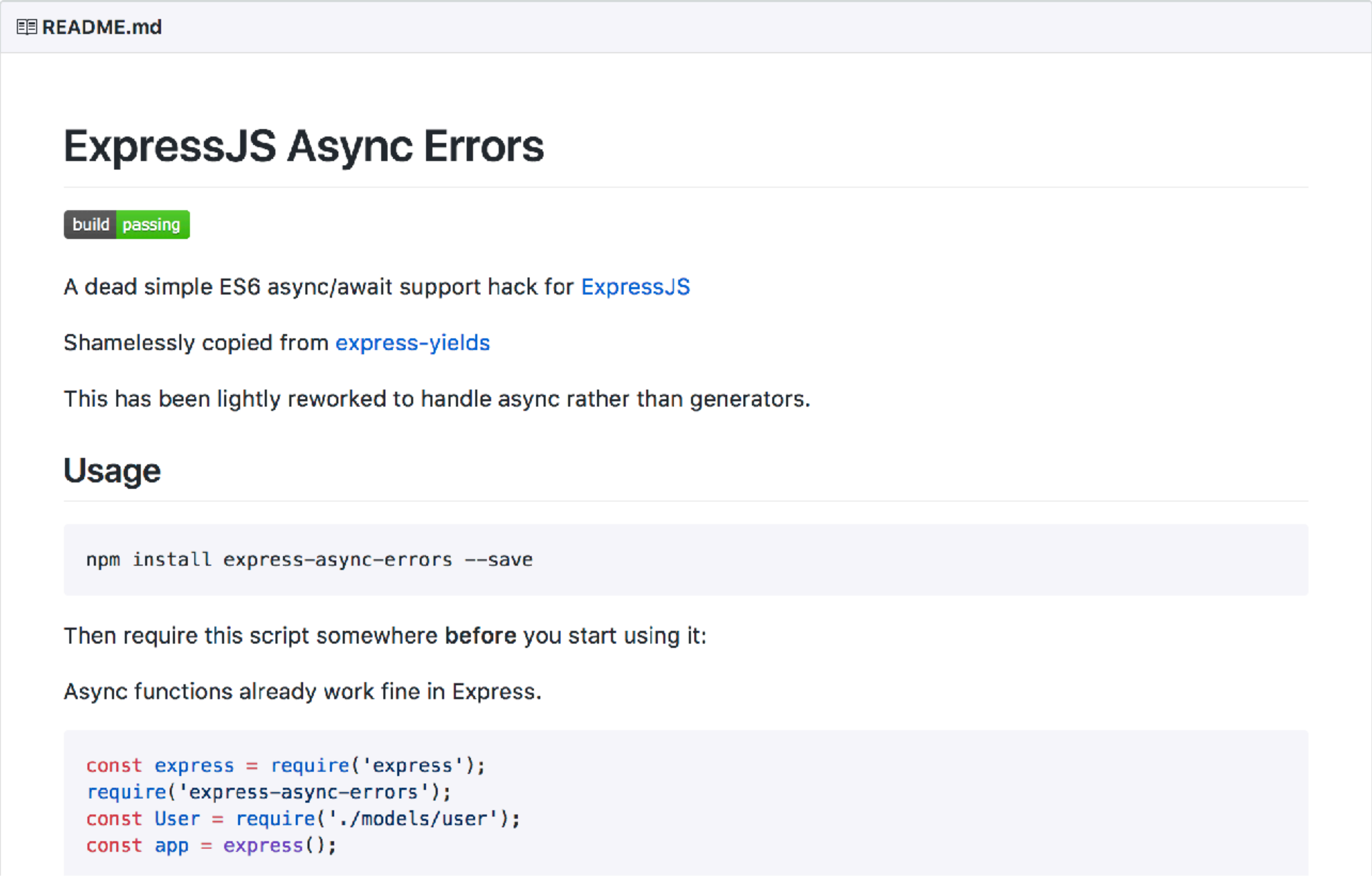
```
server.get('/users', authenticate, handleError(getUsers))
```



```
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 onreponse
(/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
```

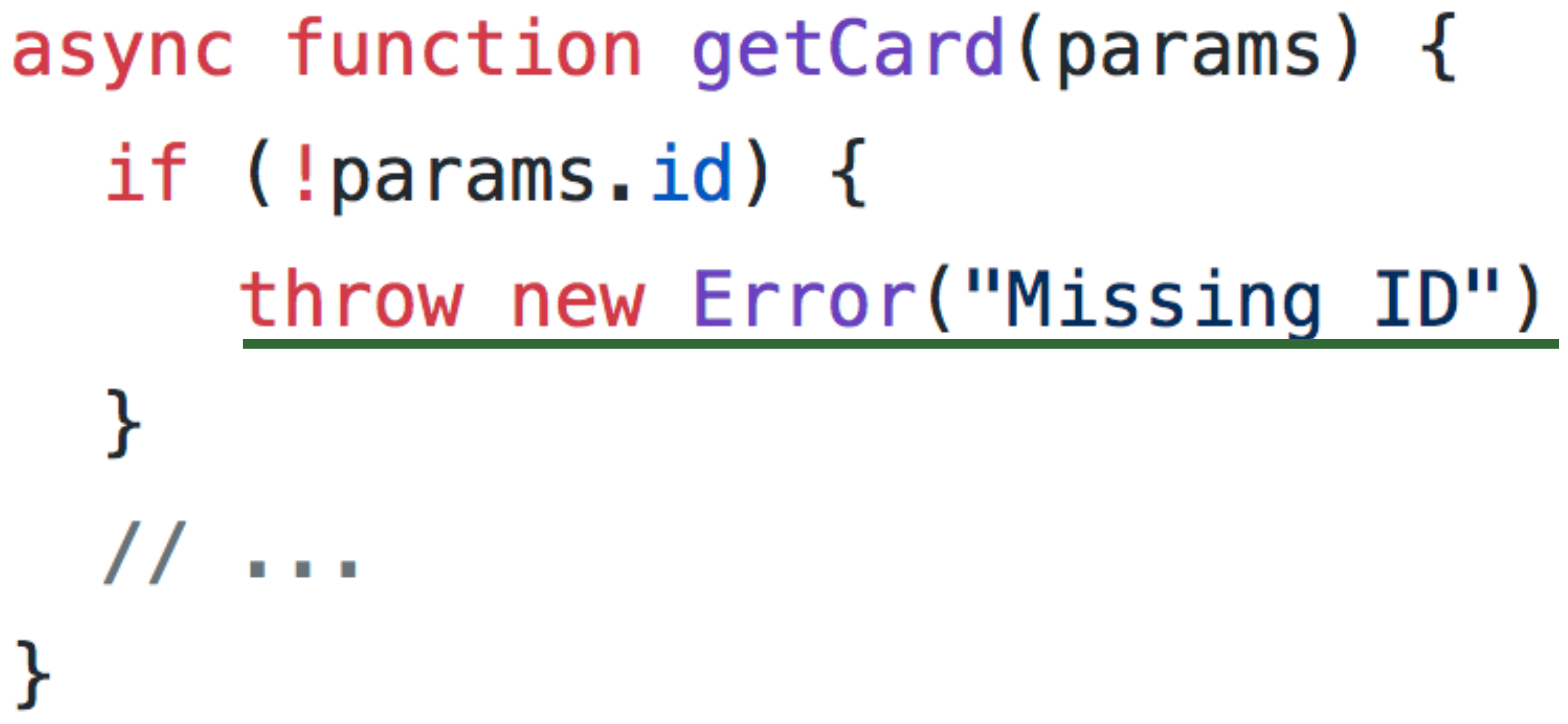
Then require this script somewhere **before** you start using it:

Async functions already work fine in Express.

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

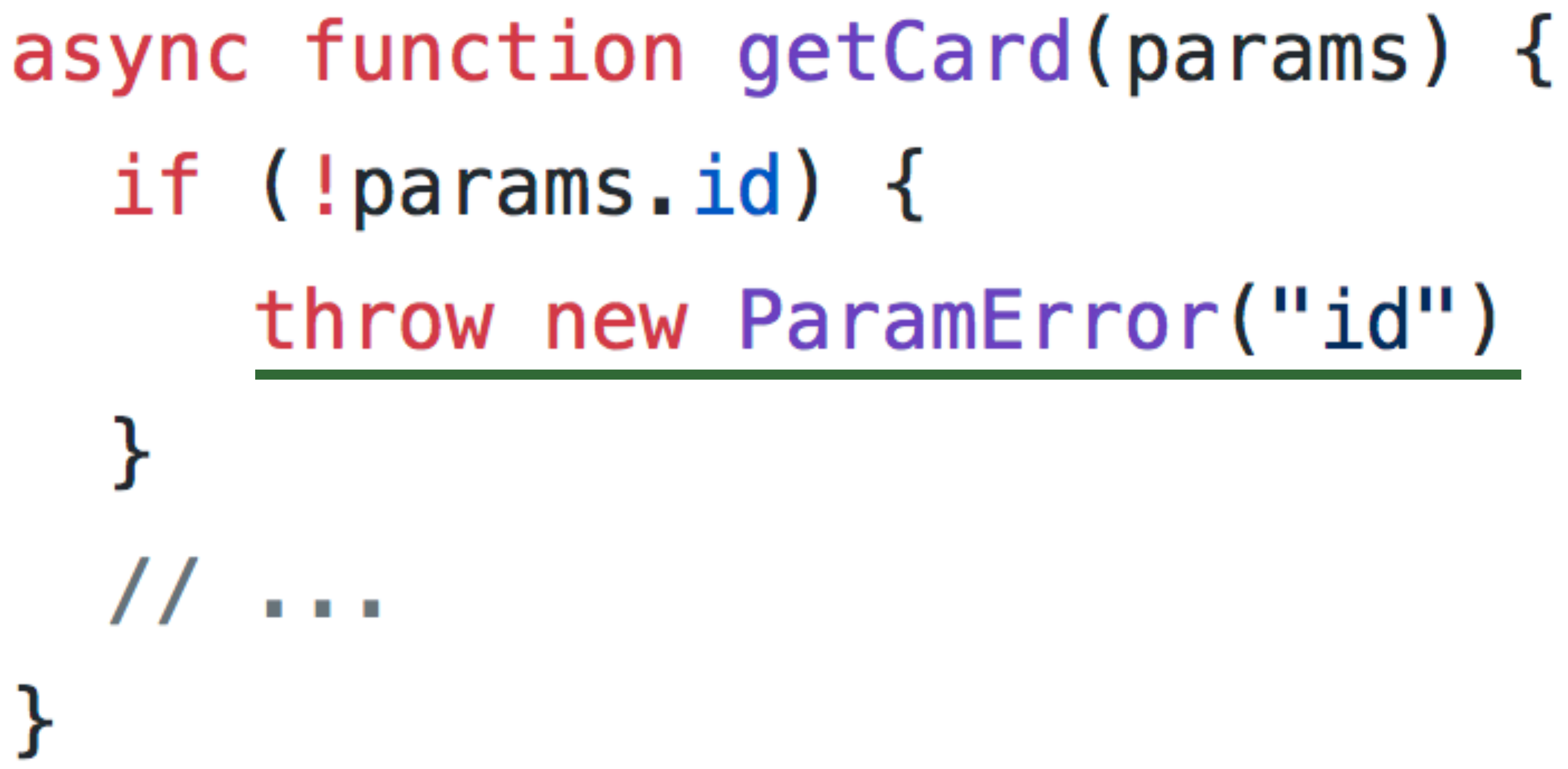
Custom Error Classes

CUSTOM ERRORS



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

CUSTOM ERRORS

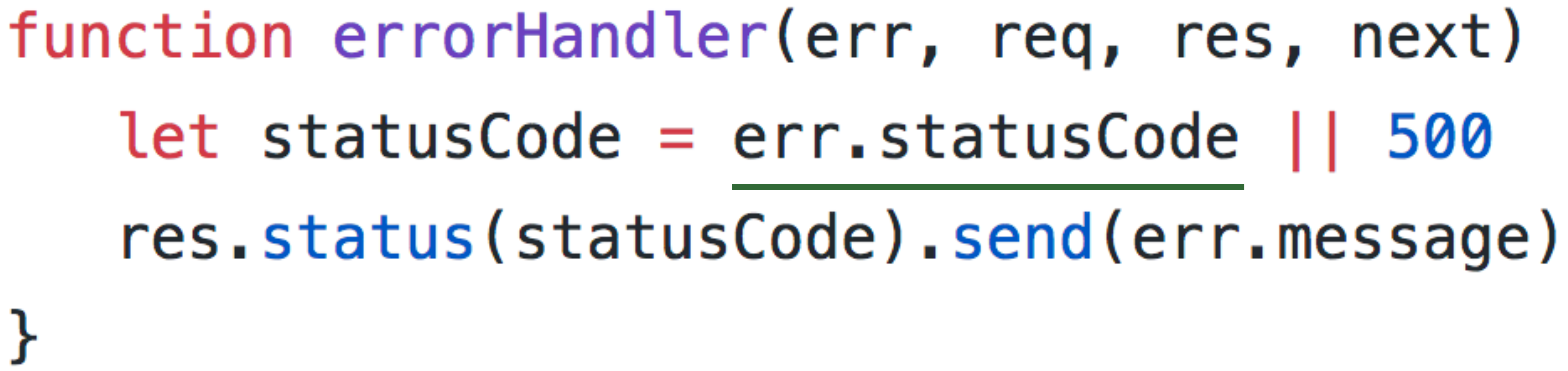


```
async function getCard(params) {
  if (!params.id) {
    throw new ParamError("id")
  }
  // ...
}
```

CUSTOM ERRORS

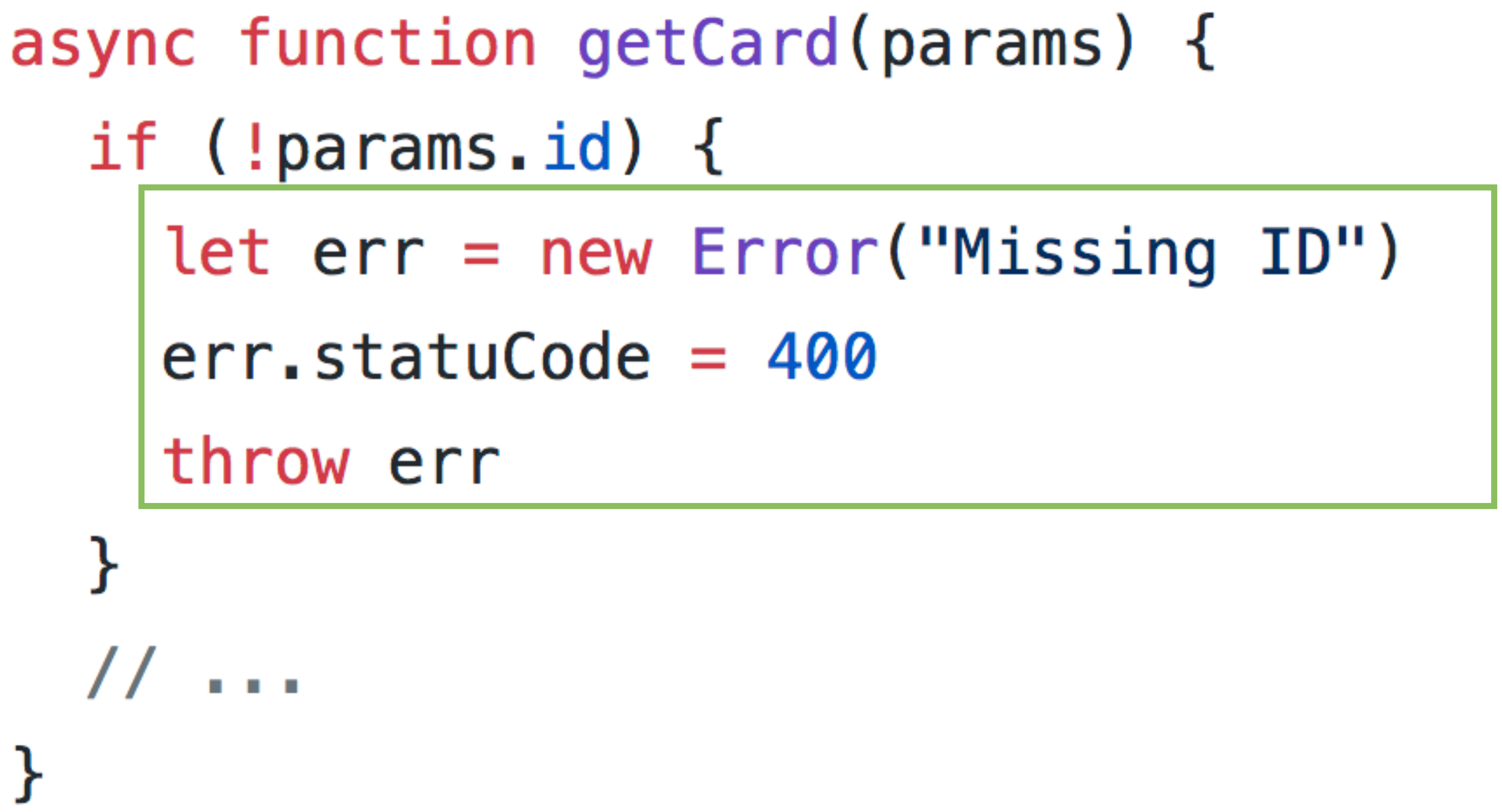
```
class ParamError extends Error {
    constructor(field) {
    1. super(`Missing param: ${field}`)
    2. Error.captureStackTrace(this, ParamError)
    3. this.statusCode = 400
    }
}
```

CUSTOM ERRORS



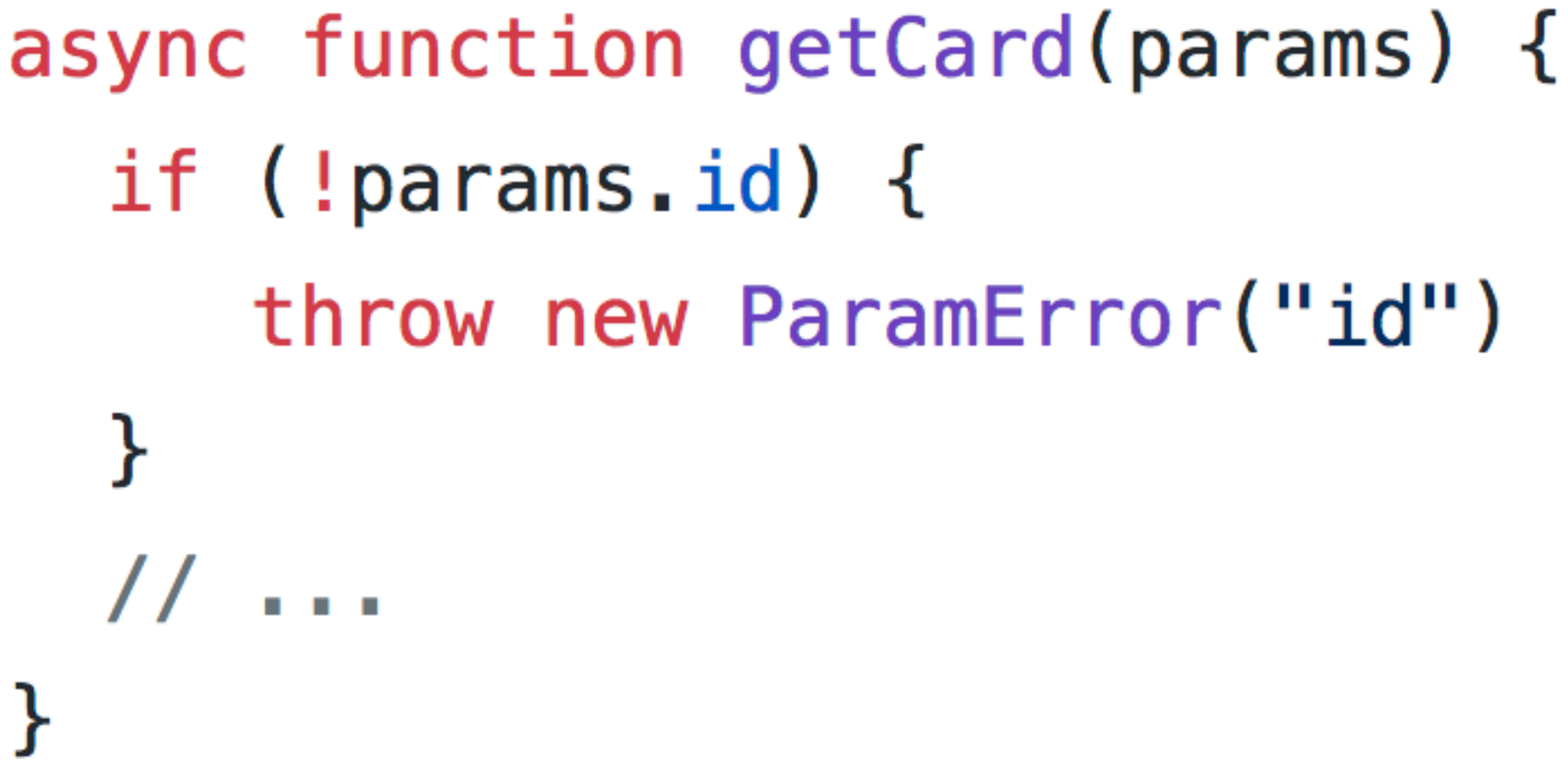
```
function errorHandler(err, req, res, next) {
   let statusCode = err.statusCode || 500
   res.status(statusCode).send(err.message)
}
```

CUSTOM ERRORS



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

CUSTOM ERRORS



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

SUMMARY

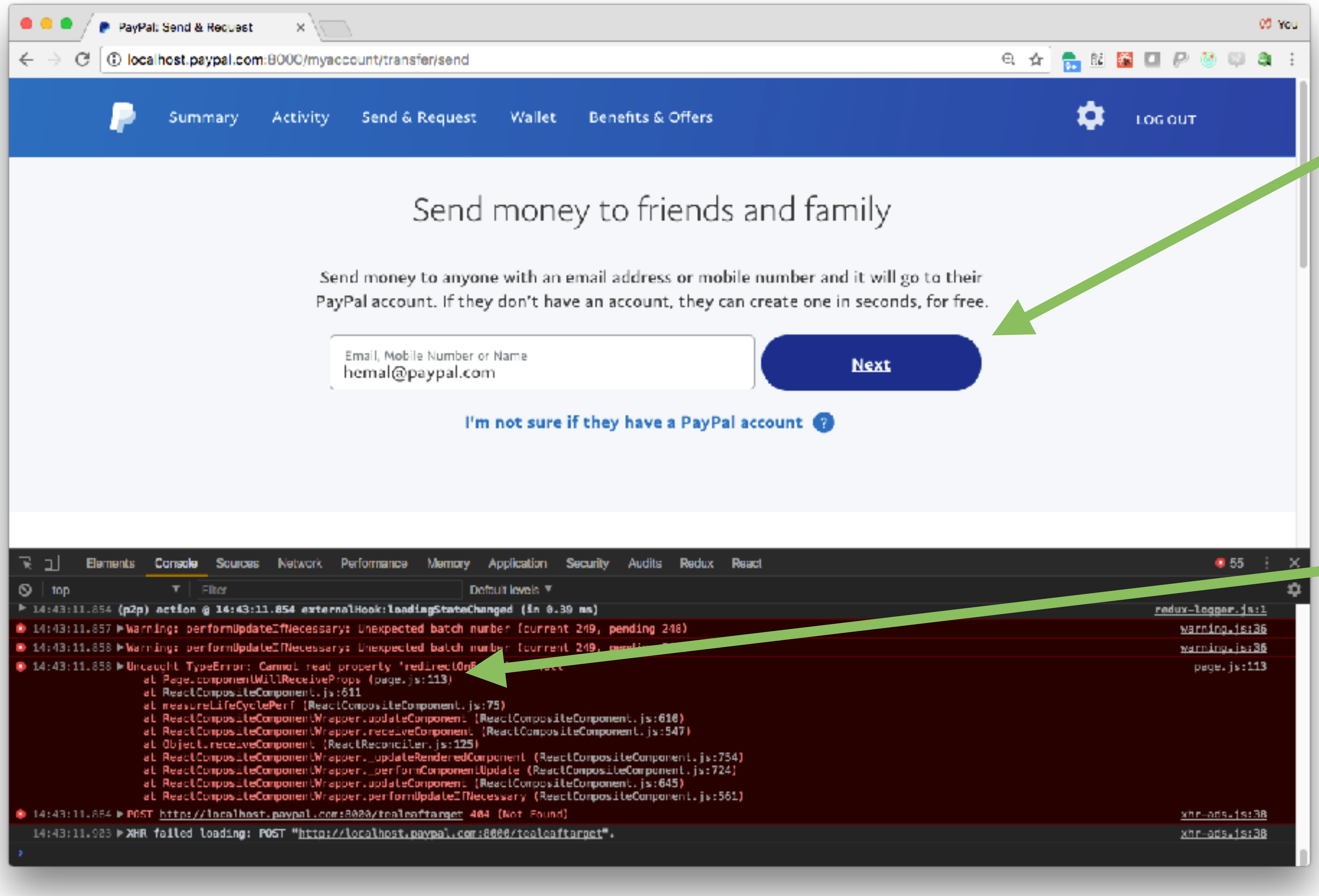
- Don’t use object literals or strings for errors (**missing stack trace**)
- Use the **Error** built-in object
- Subclass **Error** to add **statusCodes** or **to convert error codes into user-friendly error messages** for localization, etc
- We basically have one error class per micro-service to handle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

The Mystery of the Client-Side Errors

CLIENT-SIDE MONITORING

BUTTON DOESN'T WORK

REAL ISSUE USUALLY IN DEV TOOLS



Send money to friends and family

Send money to anyone with an email address or mobile number and it will go to their PayPal account. If they don't have an account, they can create one in seconds, for free.

Email, Mobile Number or Name: hemal@paypal.com

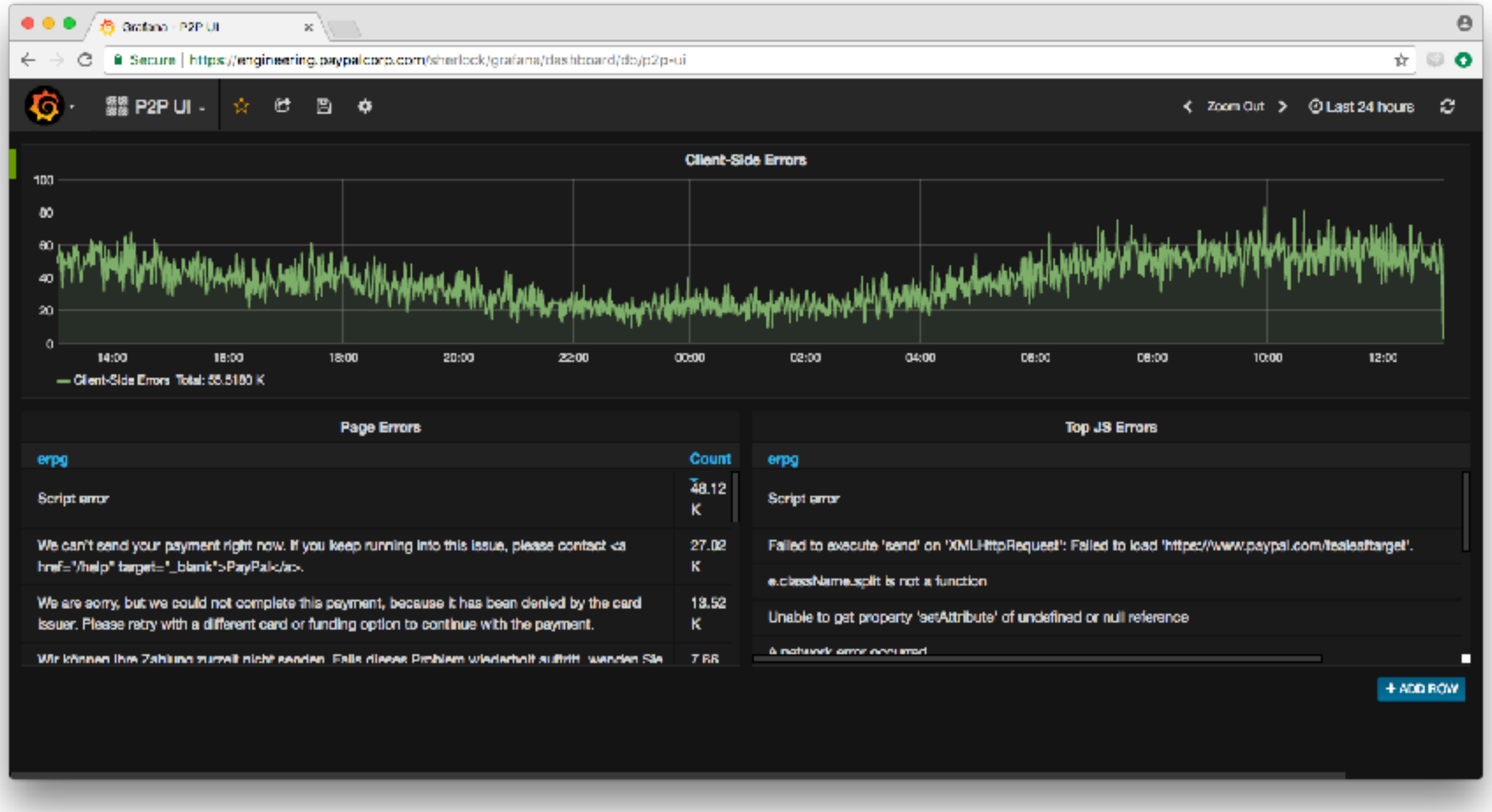
Next

I'm not sure if they have a PayPal account

```
▶ 14:43:11.854 (p2p) action @ 14:43:11.854 externalHook:loadingStateChanged (in 0.39 ms)      redux-logger.js:1
● 14:43:11.857 ▶Warning: performUpdateIfNecessary: Unexpected batch number (current 249, pending 248)      warning.js:36
● 14:43:11.858 ▶Warning: performUpdateIfNecessary: Unexpected batch number (current 249,      warning.js:36
● 14:43:11.858 ▶Uncaught TypeError: Cannot read property 'redirectOn      page.js:113
                    at Page.componentWillReceiveProps (page.js:113)
                    at ReactCompositeComponent.js:611
                    at measureLifeCyclePerf (ReactCompositeComponent.js:75)
                    at ReactCompositeComponentWrapper.updateComponent (ReactCompositeComponent.js:618)
                    at ReactCompositeComponentWrapper.receiveComponent (ReactCompositeComponent.js:547)
                    at Object.receiveComponent (ReactReconciler.js:125)
                    at ReactCompositeComponentWrapper._updateRenderedComponent (ReactCompositeComponent.js:754)
                    at ReactCompositeComponentWrapper._performComponentUpdate (ReactCompositeComponent.js:724)
                    at ReactCompositeComponentWrapper.updateComponent (ReactCompositeComponent.js:645)
                    at ReactCompositeComponentWrapper.performUpdateIfNecessary (ReactCompositeComponent.js:561)
● 14:43:11.864 ▶POST http://localhost.paypal.com:8000/tealeaftarget 404 (Not Found)      xhr-ads.js:38
  14:43:11.923 ▶XHR failed loading: POST "http://localhost.paypal.com:8000/tealeaftarget".      xhr-ads.js:38
```

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
}
```

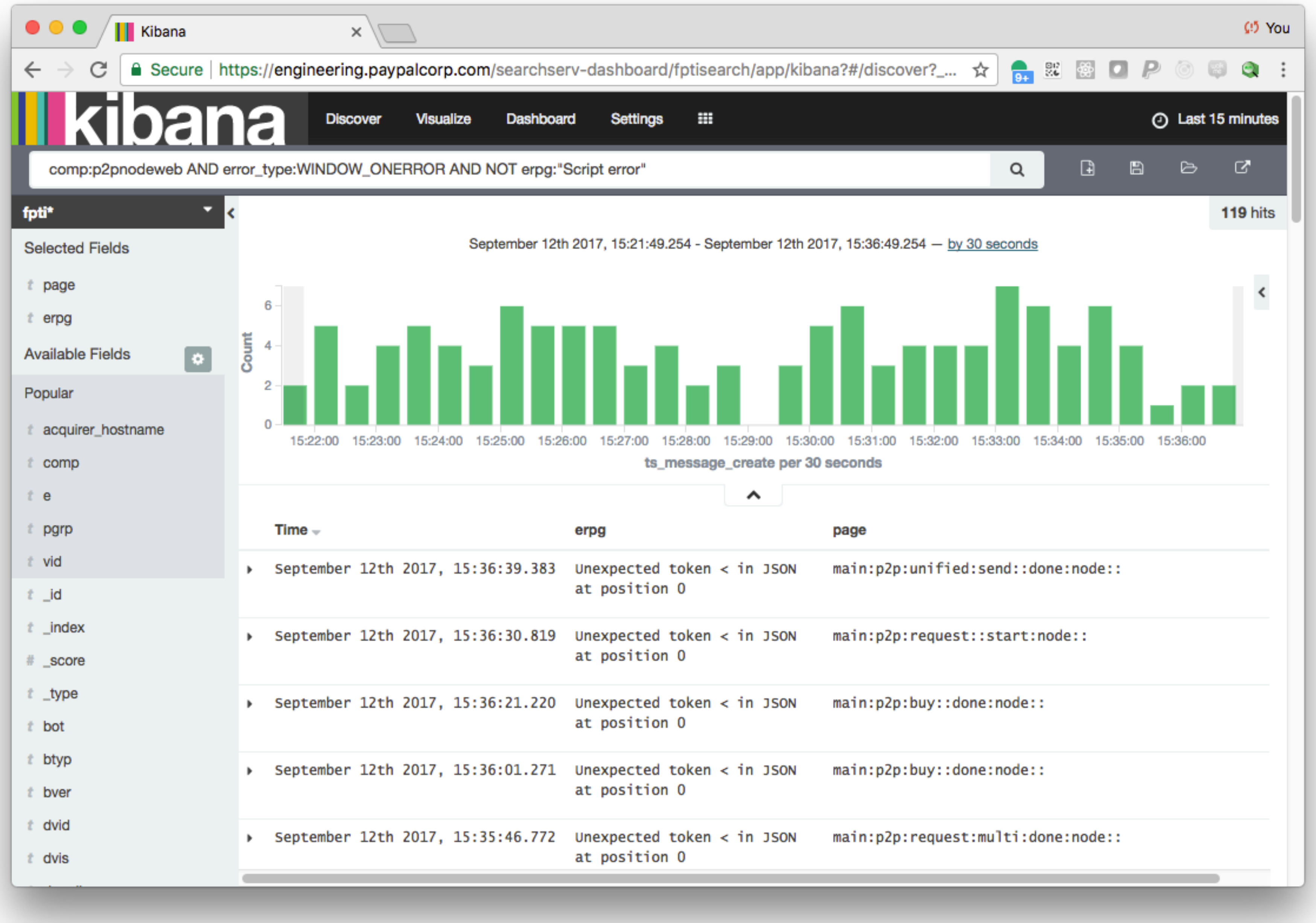


Page Errors

erpg	Count
Script error	48.12 K
We can't send your payment right now. If you keep running into this issue, please contact <a href="/help" target="_blank">PayPal</a>.	27.02 K
We are sorry, but we could not complete this payment, because it has been denied by the card issuer. Please retry with a different card or funding option to continue with the payment.	13.52 K
Wir können Ihre Zahlung zurzeit nicht senden. Falls dieses Problem wiederholt auftritt, wenden Sie	7.66

Top JS Errors

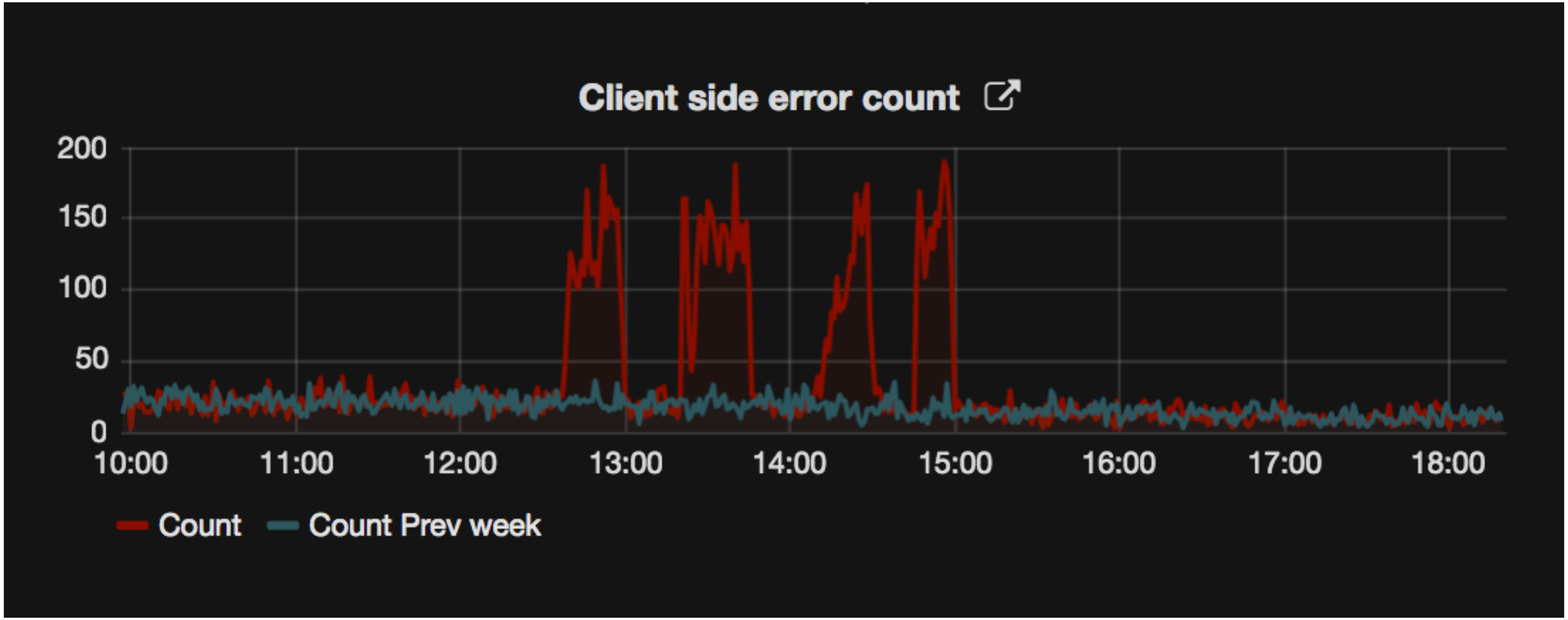
erpg
Script error
Failed to execute 'send' on 'XMLHttpRequest': Failed to load 'https://www.paypal.com/tealeaftarget'.
e.className.split is not a function
Unable to get property 'setAttribute' of undefined or null reference
A network error occurred



comp:p2pnodeweb AND error_type:WINDOW_ONERROR AND NOT erpg:"Script error"

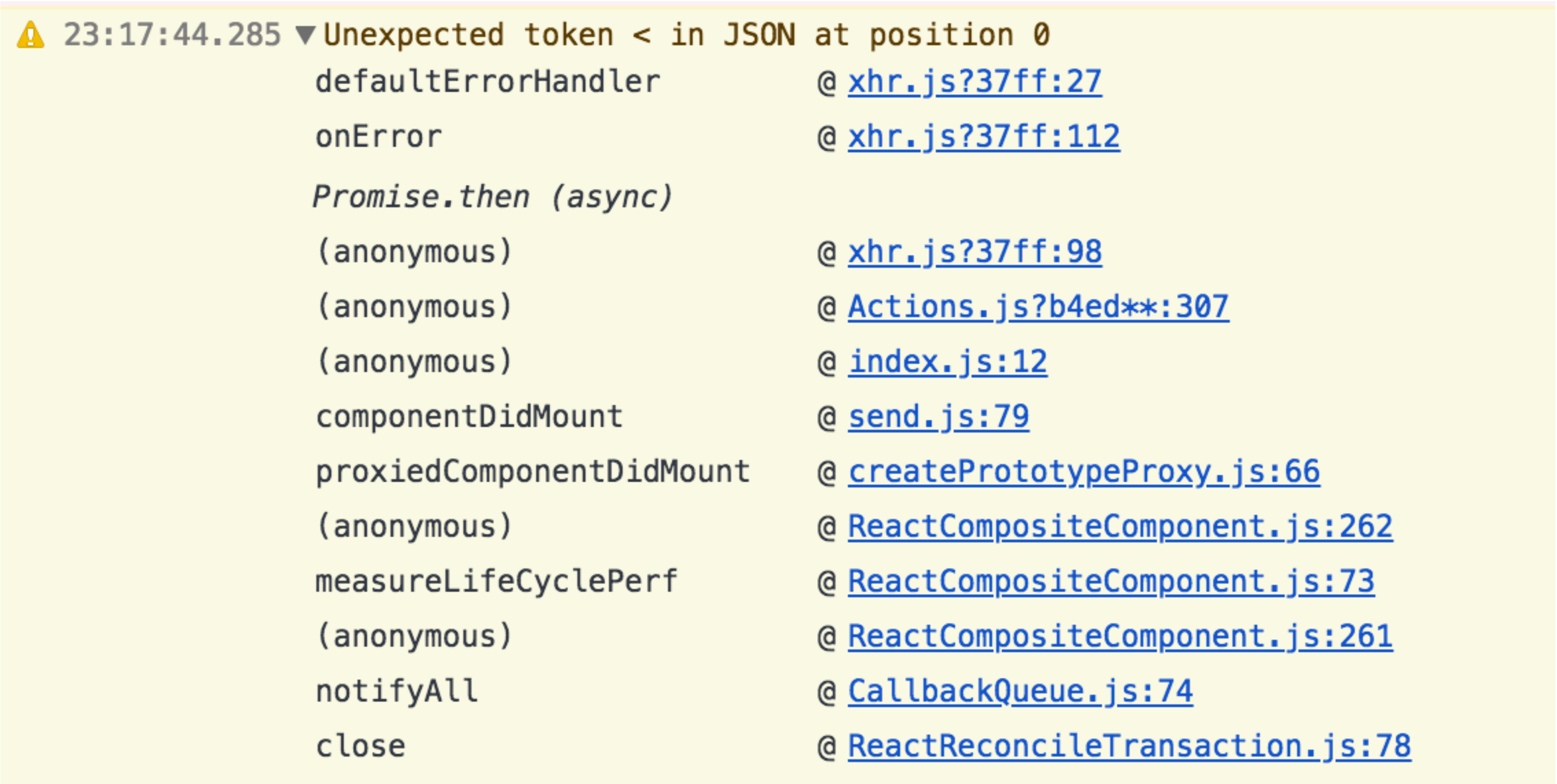
Time	erpg	page
September 12th 2017, 15:36:39.383	Unexpected token < in JSON at position 0	main:p2p:unified:send::done:node::
September 12th 2017, 15:36:30.819	Unexpected token < in JSON at position 0	main:p2p:request::start:node::
September 12th 2017, 15:36:21.220	Unexpected token < in JSON at position 0	main:p2p:buy::done:node::
September 12th 2017, 15:36:01.271	Unexpected token < in JSON at position 0	main:p2p:buy::done:node::
September 12th 2017, 15:35:46.772	Unexpected token < in JSON at position 0	main:p2p:request:multi:done:node::

WE NOTICED A SPIKE DURING DEPLOY



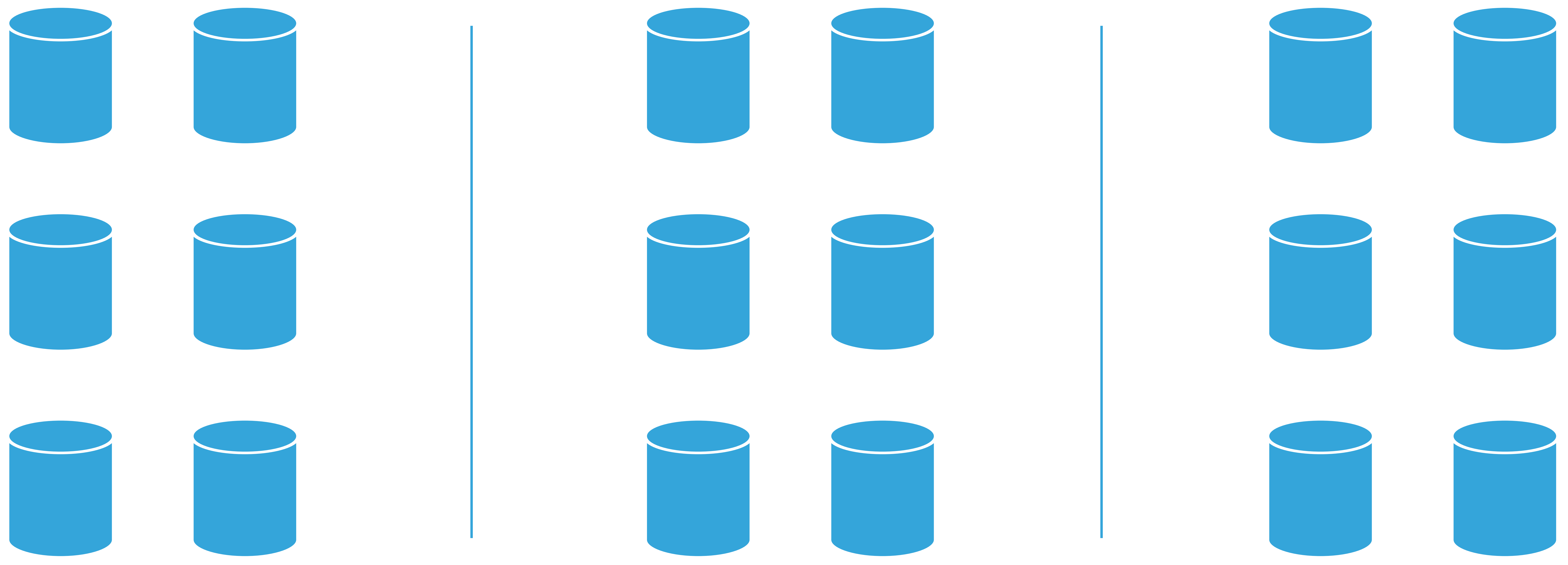
WE CONGRATULATED OURSELVES…THEN ACTUALLY LOOKED INTO THE BUG

THE MYSTERY OF THE CLIENT-SIDE ERRORS

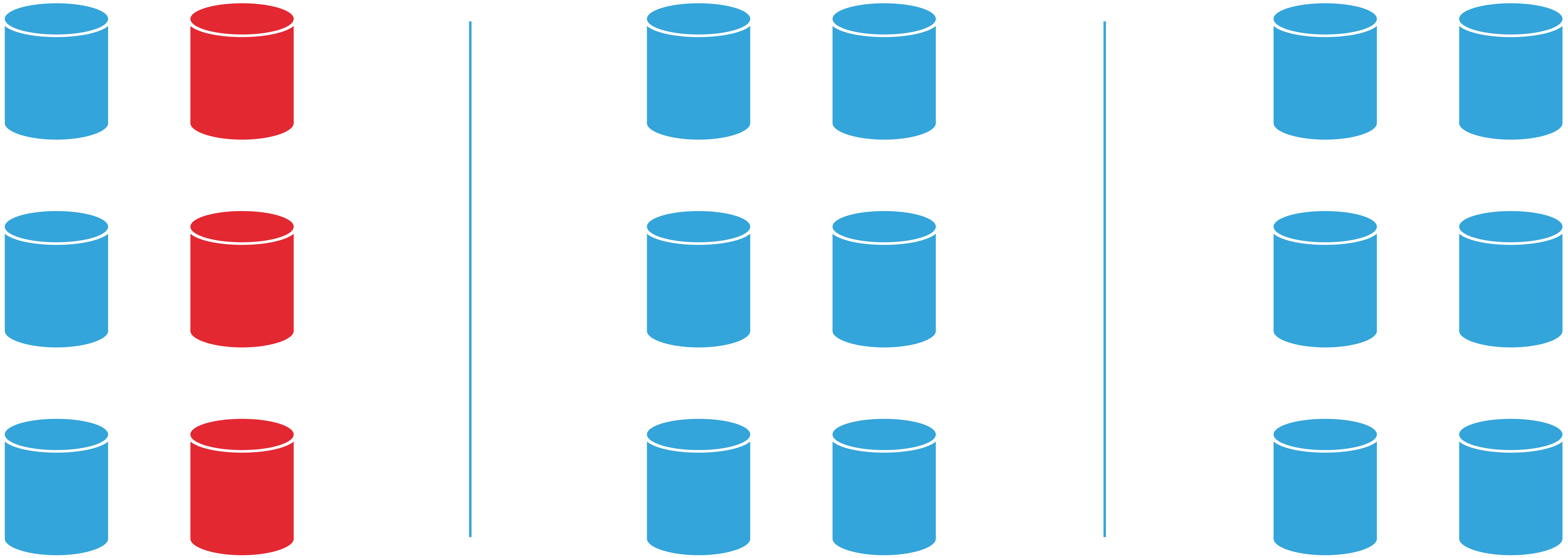


```
⚠ 23:17:44.285 ▼Unexpected token < in JSON at position 0
                  defaultErrorHandler         @ xhr.js?37ff:27
                  onError                     @ xhr.js?37ff:112
                  Promise.then (async)
                  (anonymous)                 @ xhr.js?37ff:98
                  (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
```

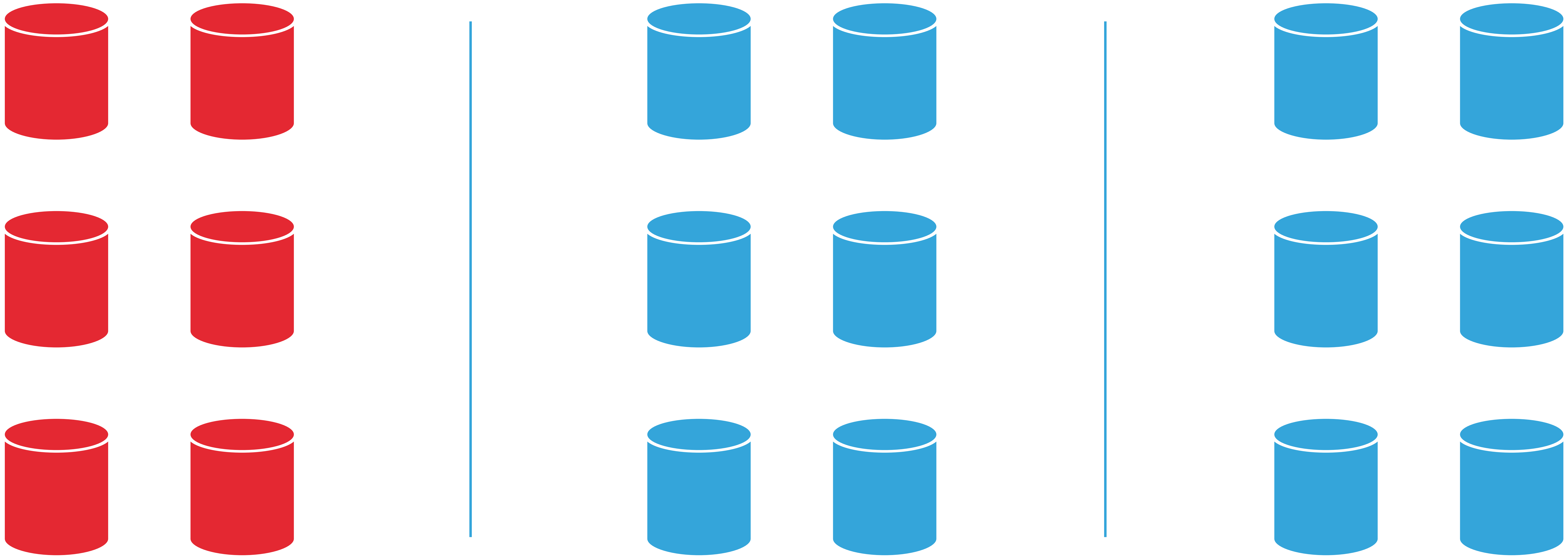
WE HAVE A LOT OF SERVERS



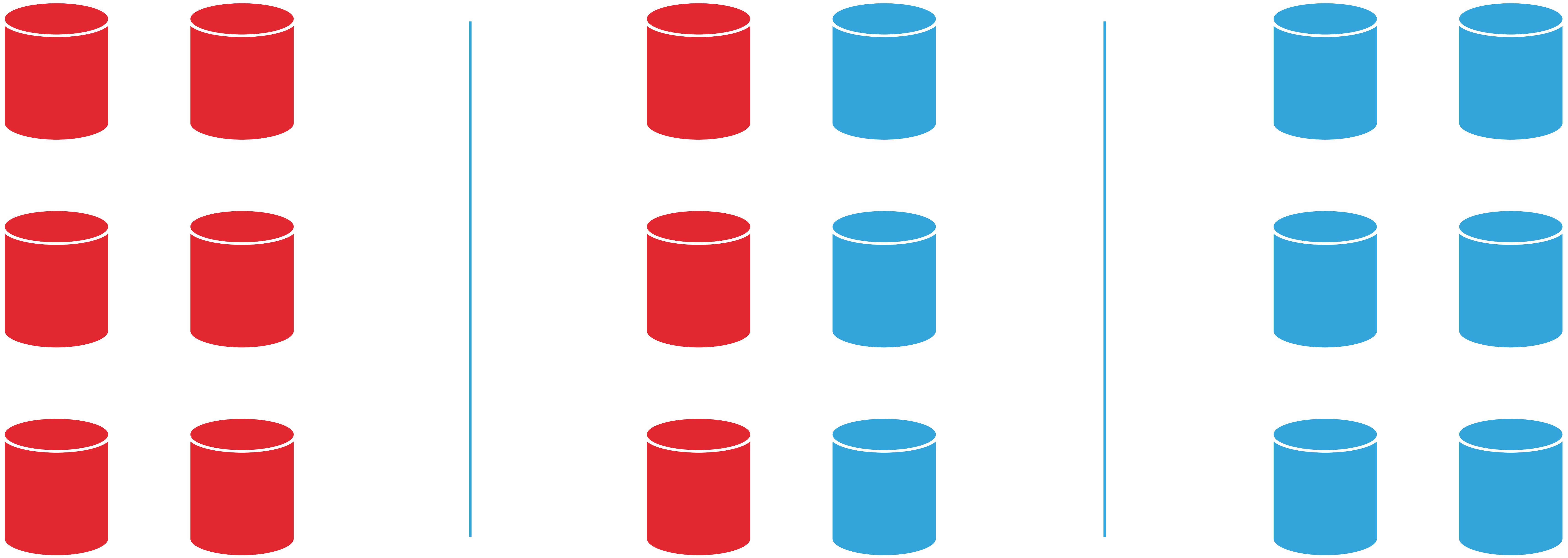
WE HAVE A LOT OF SERVERS



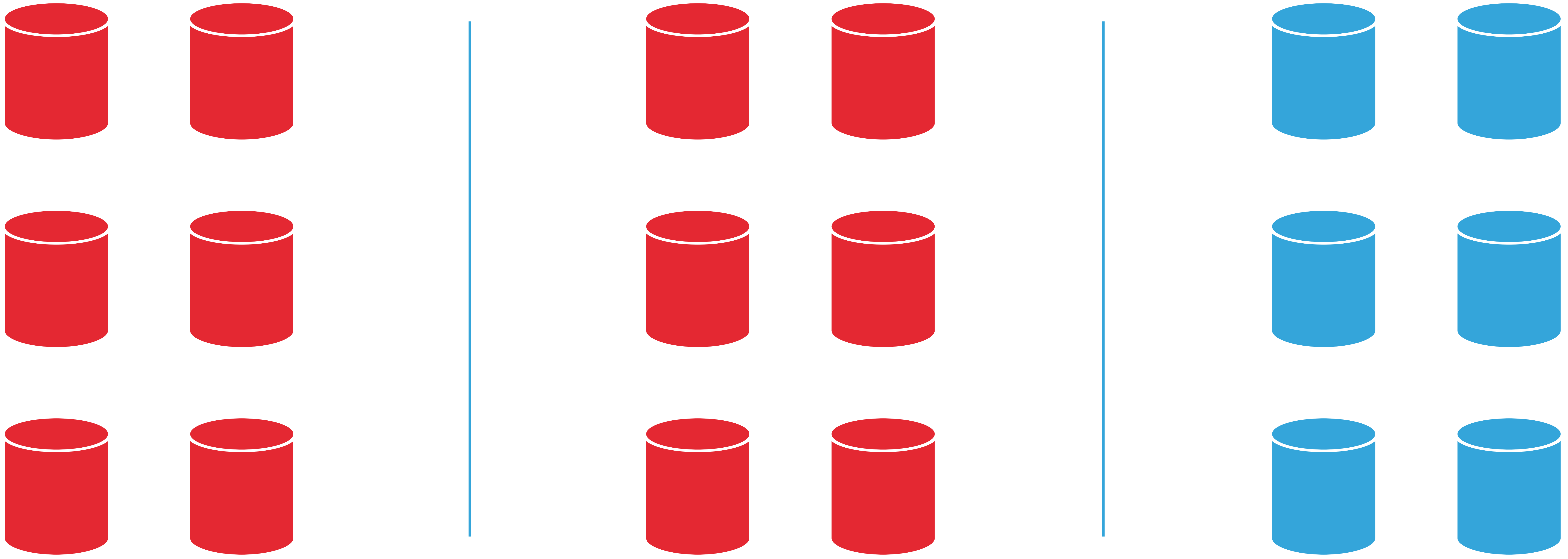
WE HAVE A LOT OF SERVERS



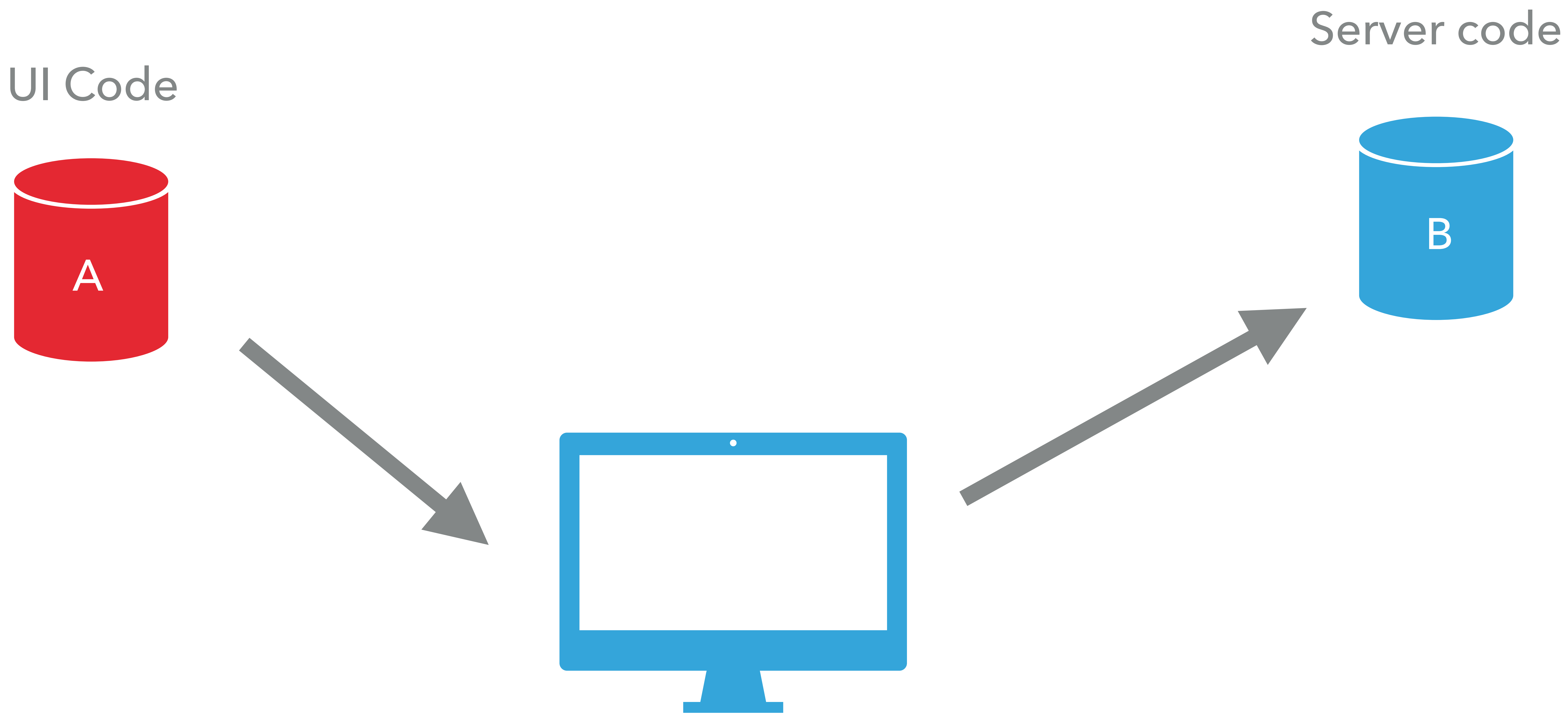
WE HAVE A LOT OF SERVERS



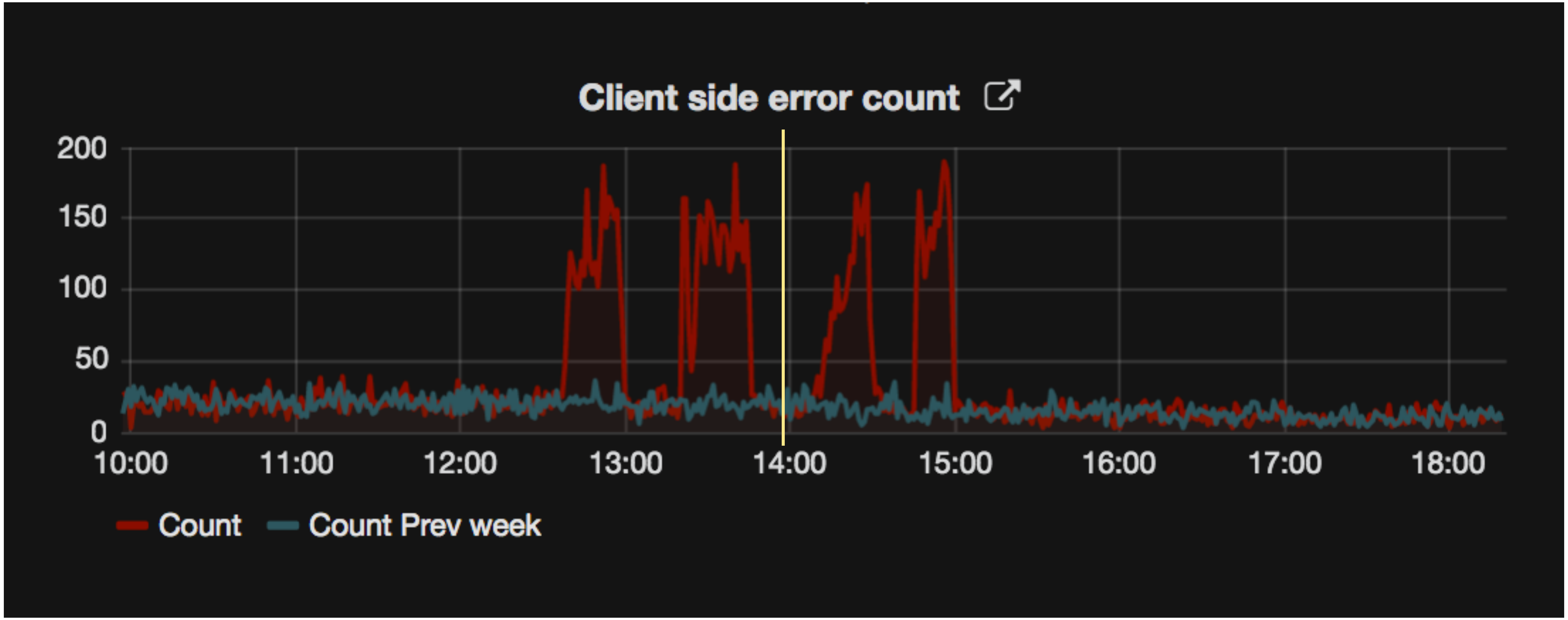
WE HAVE A LOT OF SERVERS



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