

Profiling JavaScript Performance

Welcome

Greetings!



Luke DeWitt

EAST COASTER / HUSBAND / FATHER OF 3 UNDER 5 (TIRED) /
DEVELOPER / DIRECTOR OF WEB SERVICES


RED
space™


MTV


SONY
PICTURES
TELEVISION


NBA
NBA


CN
CARTOON NETWORK


FOX


Paramount
A VIACOM COMPANY


IBM


NBC


PBS


nick

JavaScript

- Performance is important
- Everyone loves fast sites
- JS continually redefining itself
- Low barrier to entry
- Simple to learn
- Frameworks

JavaScript



Let’s Dive In

JavaScript

- The problems are the same, front or back end

JavaScript

- Single threaded
- One “chunk” at a time

But Luke!

JavaScript

- Single threaded
- One “chunk” at a time
- Avoid blocking code

JavaScript

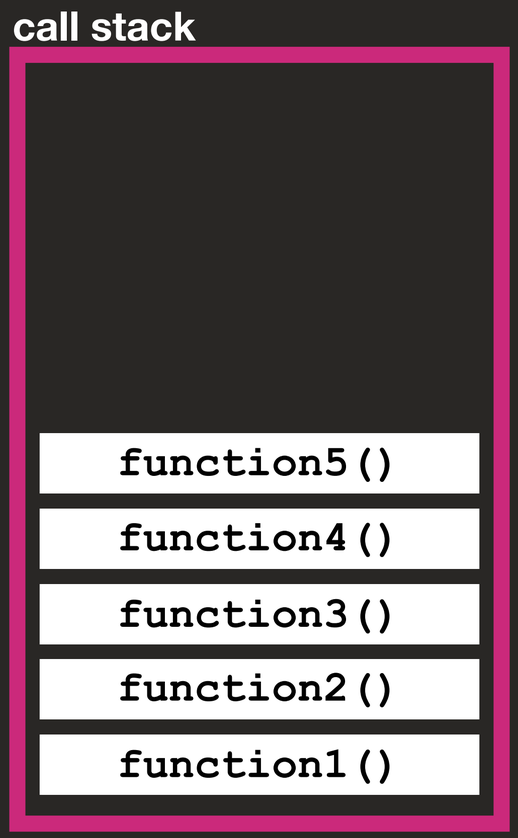
- Blocking Code
  - Render blocking
  - Unresponsive UIs
  - Accessibility Issues
  - Bad UX

Call Stack

Call Stack

- This is our “single thread”
- Last In, First Out
  - Code added when called (stack frame)
  - Popped off when “returned”
- Repeat until empty
- Do it again

The Call Stack



Call Stack Example

```
function andFinally() {
  console.log("done");
}

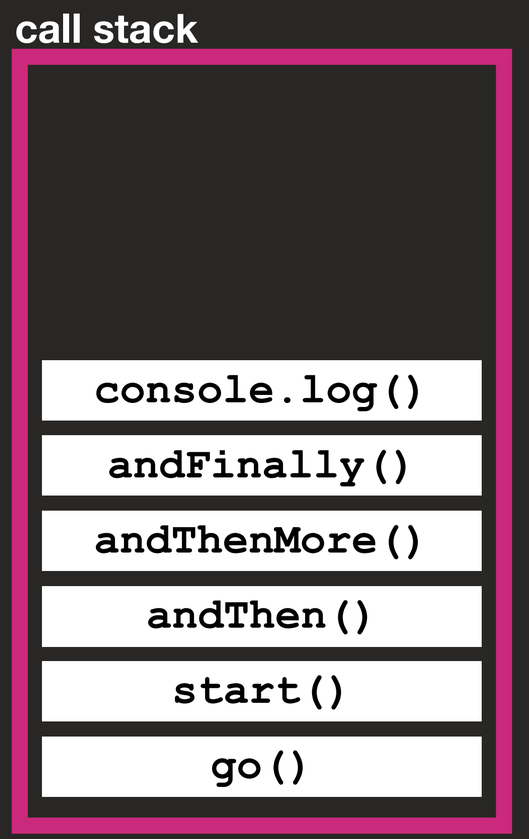
function andThenMore() {
  andFinally();
}

function andThen() {
  andThenMore();
}

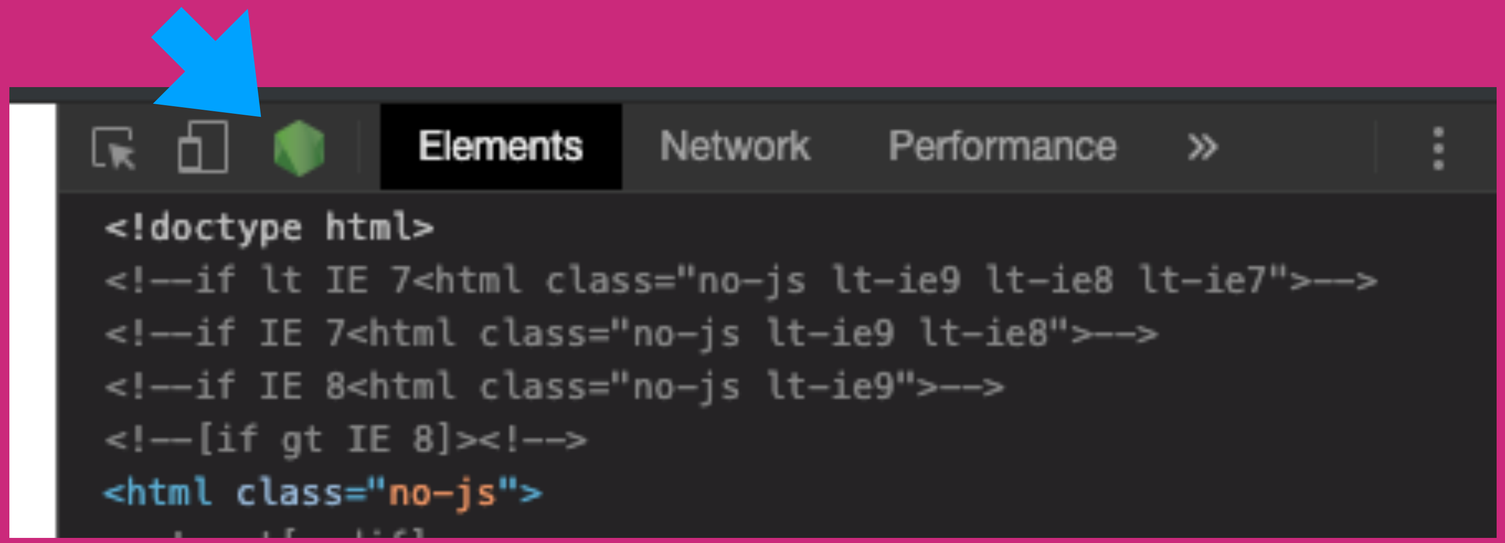
function start() {
  andThen();
}

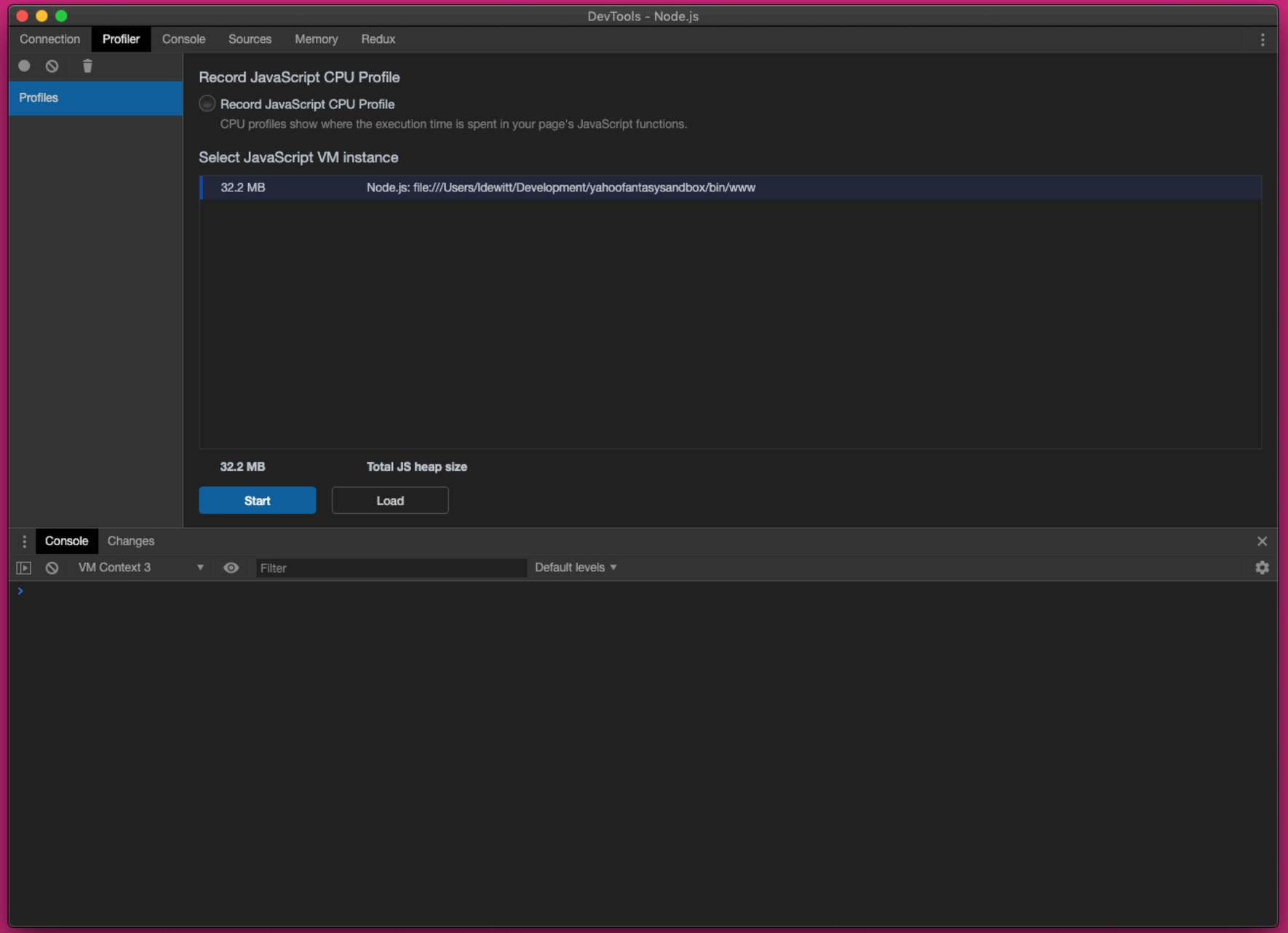
function go() {
  start();
}

go();
```

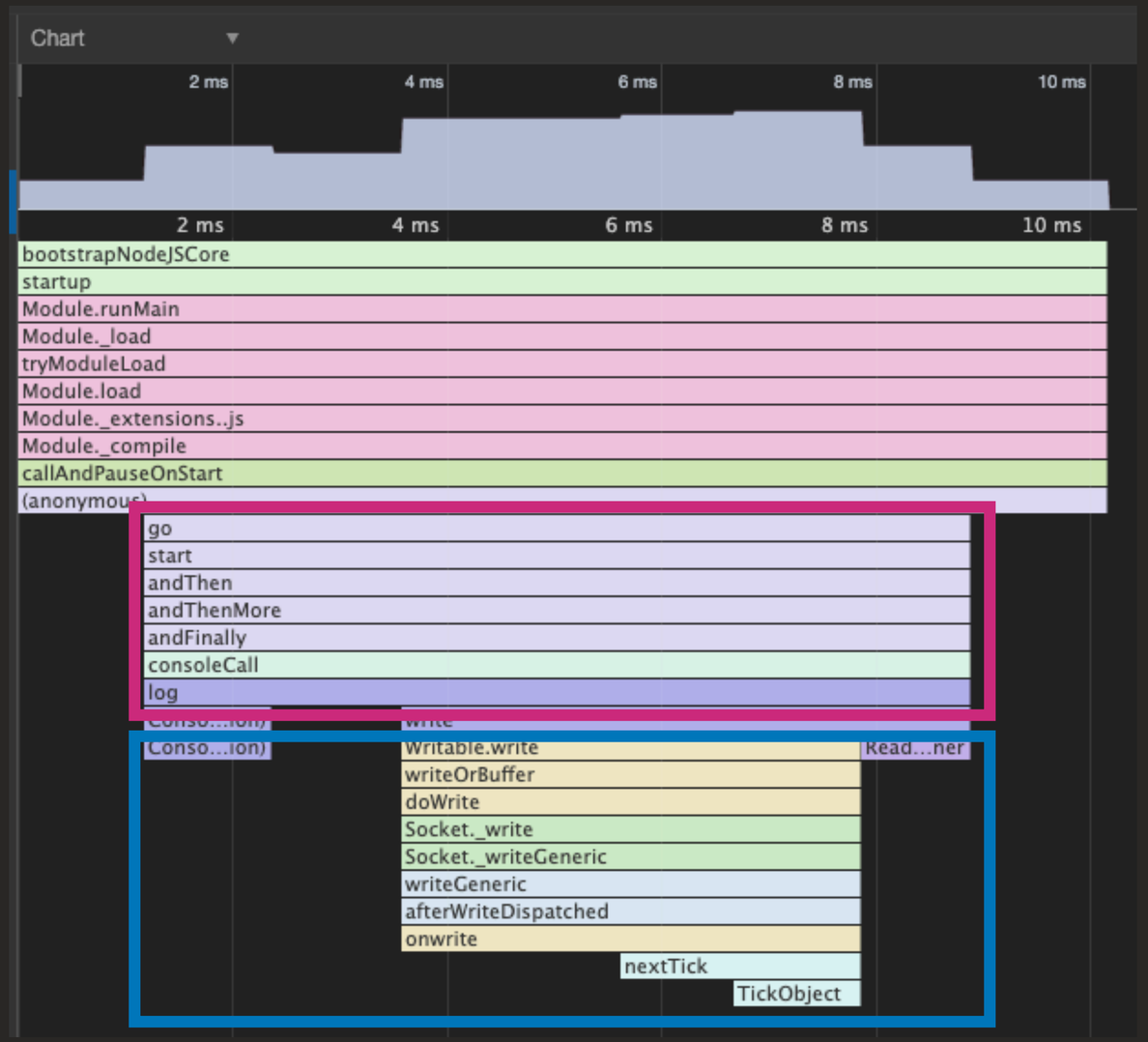


```
node --inspect app.js
node --inspect-brk app.js
```


Elements
Network
Performance
<!doctype html>
<!--if lt IE 7<html class="no-js lt-ie9 lt-ie8 lt-ie7">-->
<!--if IE 7<html class="no-js lt-ie9 lt-ie8">-->
<!--if IE 8<html class="no-js lt-ie9">-->
<!--[if gt IE 8]><!-->
<html class="no-js">


DevTools - Node.js
Connection
Profiler
Console
Sources
Memory
Redux
Profiles
Record JavaScript CPU Profile
Record JavaScript CPU Profile
CPU profiles show where the execution time is spent in your page's JavaScript functions.
Select JavaScript VM instance
32.2 MB
Node.js: file:///Users/ldewitt/Development/yahoofantasysandbox/bin/www
32.2 MB
Total JS heap size
Start
Load
Console
Changes
VM Context 3
Filter
Default levels

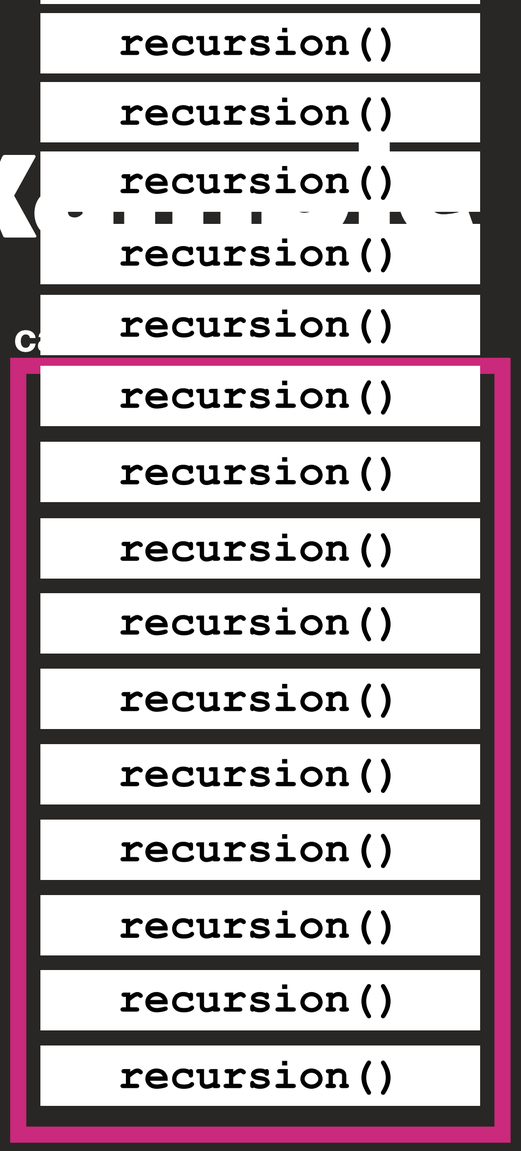
Profile Graph



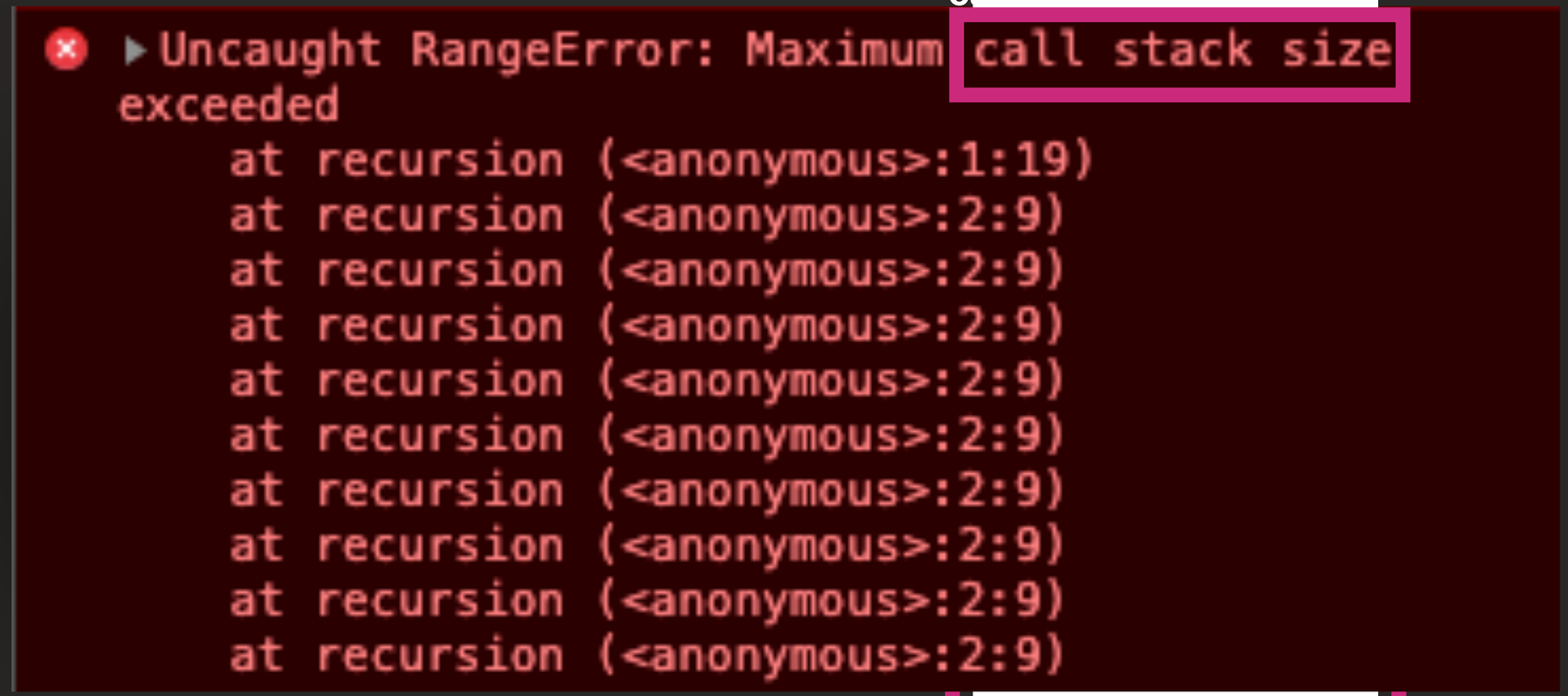
Recursion Ex

```
function recursion(n) {
    return recursion(n);
};

console.log(recursion(100));
```



Recursion Ex



Example

```
const cube = n => {
  let result = n;

  while (result < n * n * n) {
    result += n;
  }

  return result;
};

console.log(cube(parseInt(process.argv.pop(), 10)));
```



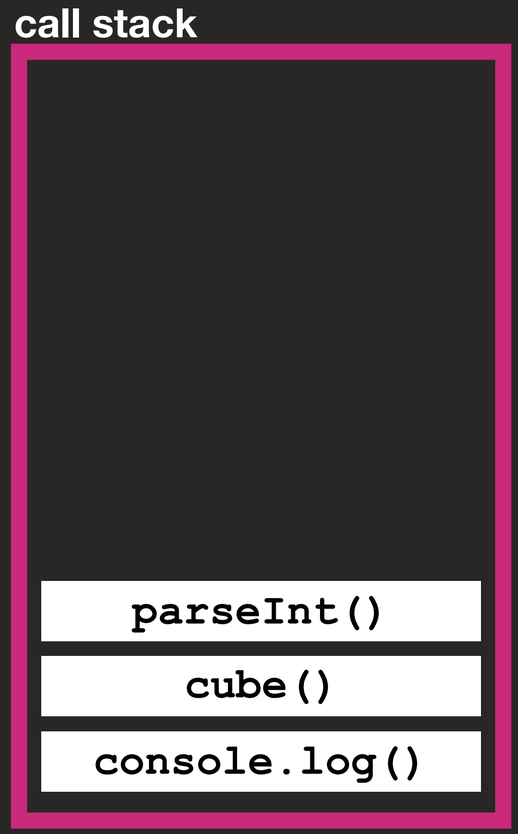
Call Stack Example

```
const cube = n => {
  let result = n;

  while (result < n * n * n) {
    result += n;
  }

  return result;
};

console.log(cube(parseInt(process.argv.pop(), 10)));
```



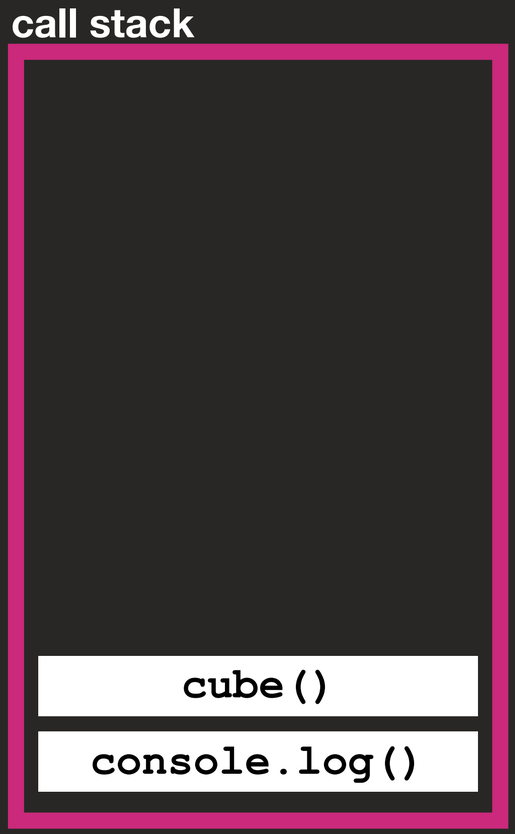
Call Stack Example

```
const cube = n => {
  let result = n;

  while (result < n * n * n) {
    result += n;
  }

  return result;
};

console.log(cube(parseInt(process.argv.pop(), 10)));
```



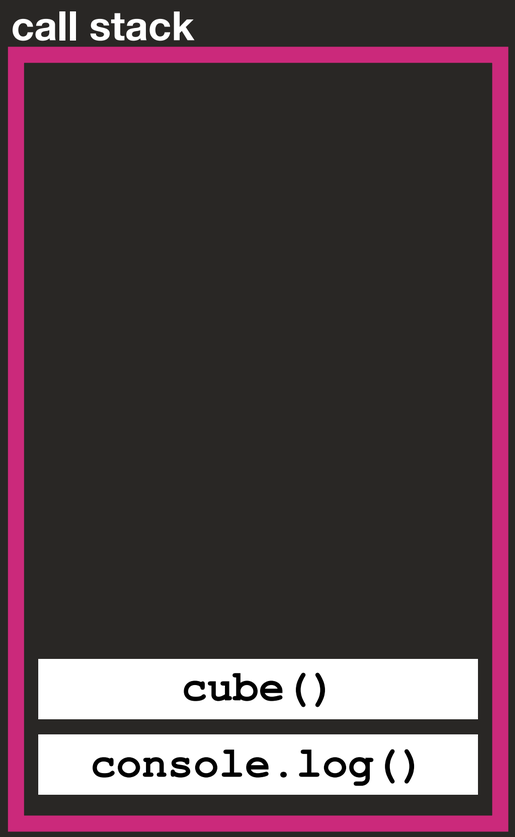
Call Stack Example

```
const cube = n => {
  let result = n;

  while (result < n * n * n) {
    result += n;
  }

  return result;
};

console.log(cube(parseInt(process.argv.pop(), 10)));
```



Profile Graph



JavaScript







JavaScript

- APIs are exposed
  - Timers
  - Callbacks
  - Promises
  - Async/Await
- Limits “Blocking” code
- setTimeout === window.setTimeout

Event Loop

- Responsible for pushing items onto the Call Stack
- Responsible for executing the top most stack frame
- Runs when the stack is empty
- Also responsible for pulling stack frames from the queue(s) onto the Call Stack for execution

External APIs

- Implemented by the browsers / node runtime
  - “Web APIs” for browsers
  - C++ for node
- Essentially act as threading mechanisms
- Better code execution
- Multiple Queues

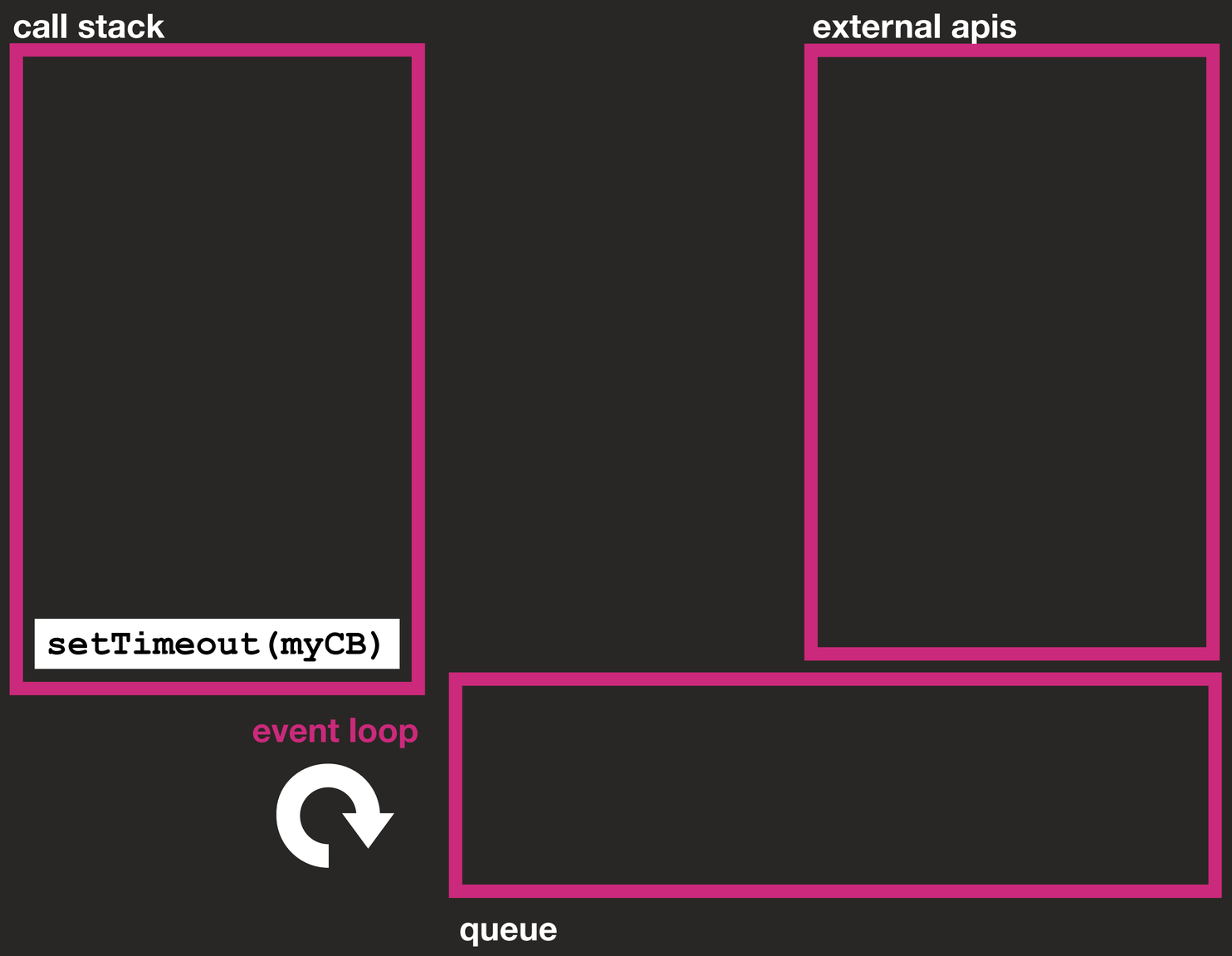
JavaScript


call stack
external apis
function3()
function2()
function1()
event loop
queue

JavaScript

```
const myCB () => {
  console.log("done!");
}

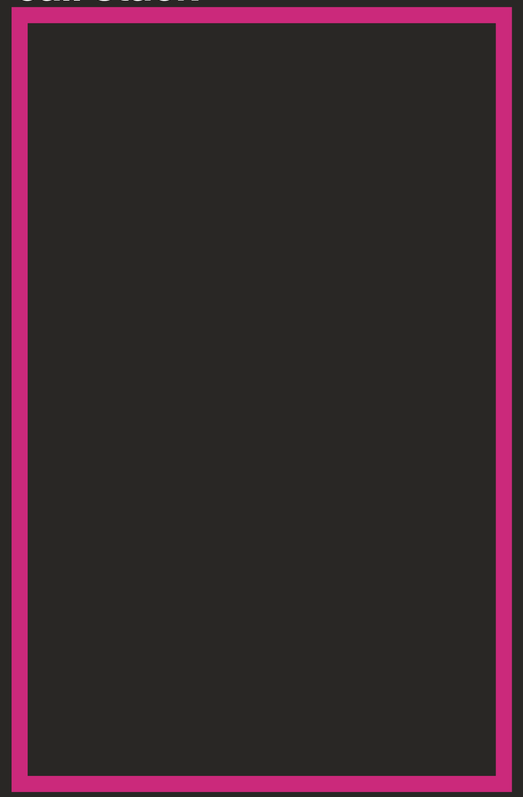
setTimeout(myCB, 5000);
```

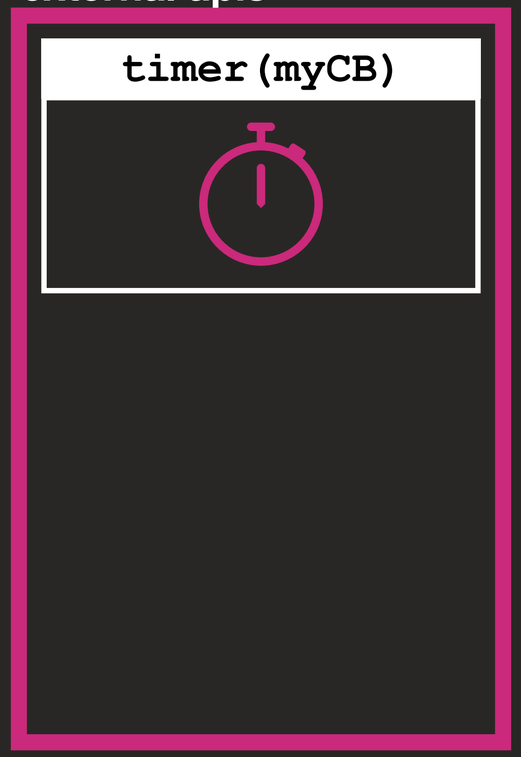


JavaScript

```
const myCB () => {
  console.log("done!");
}

setTimeout(myCB, 5000);
```





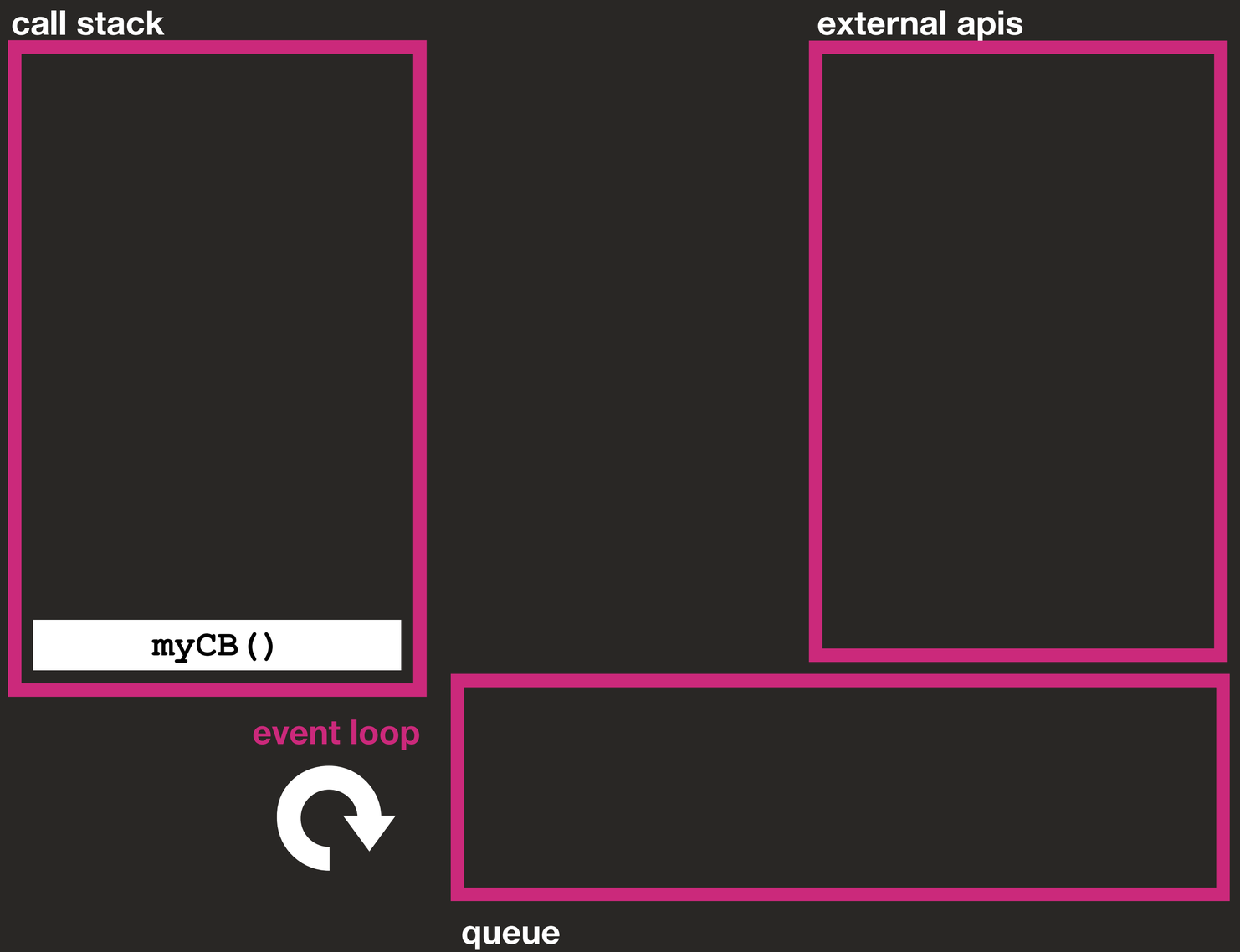
event loop



JavaScript

```
const myCB () => {
  console.log("done!");
}

setTimeout(myCB, 5000);
```




chrome!
Firefox!
GLUE

XHR & Promise Example

```
const axios = require("axios");

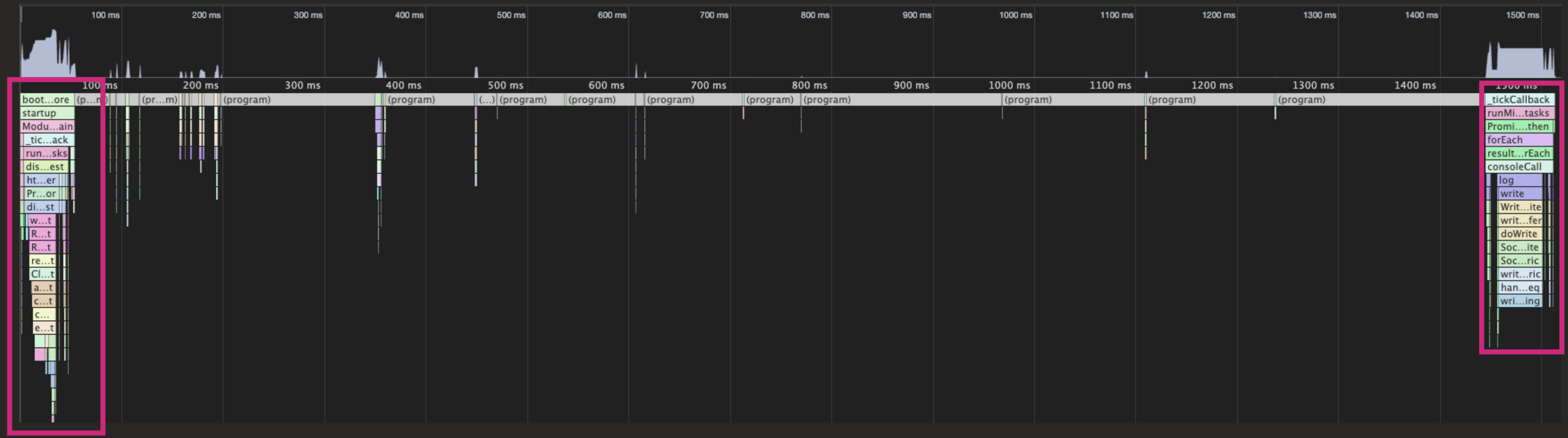
const getCharacter = i => {
  return axios.get(`https://swapi.co/api/people/${i}/`);
};

const getCharacters = count => {
  const calls = [];
  for (let i = 1; i <= count; i++) {
    calls.push(getCharacter(i));
  }

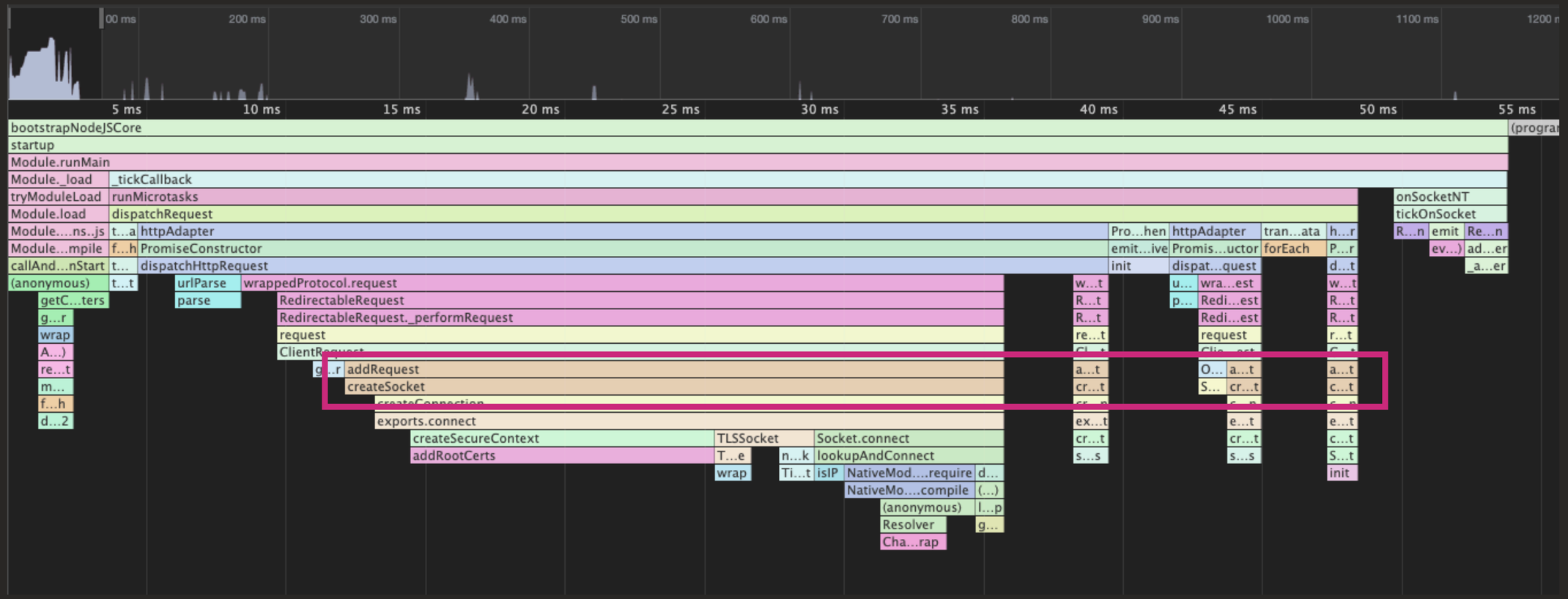
  Promise.all(calls)
    .then(([...results]) => results.forEach(({ data }) => console.log(data)))
};

getCharacters(15);
```

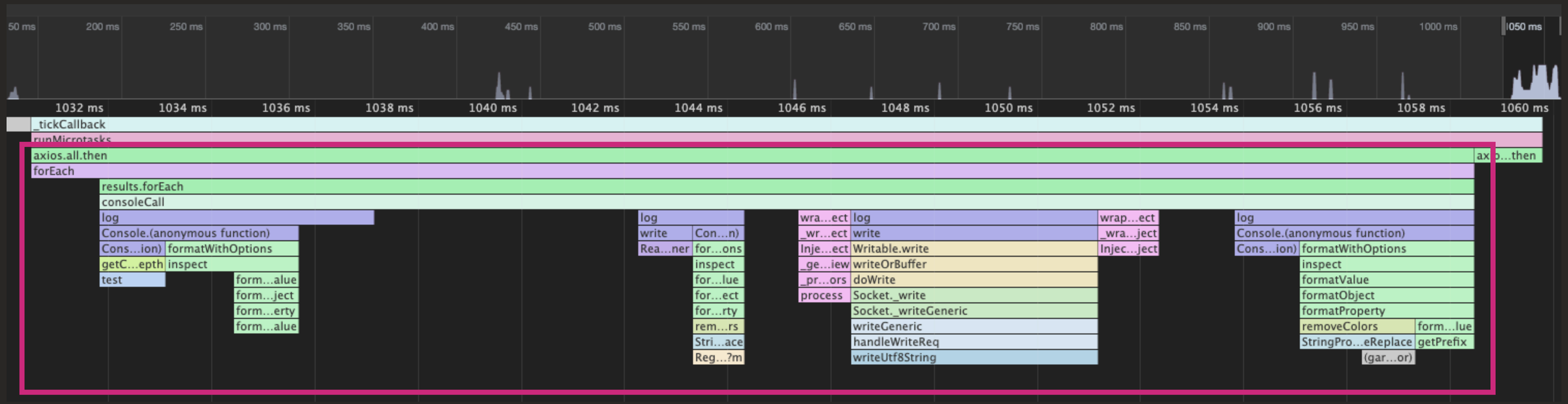
XHR & Promise Graph



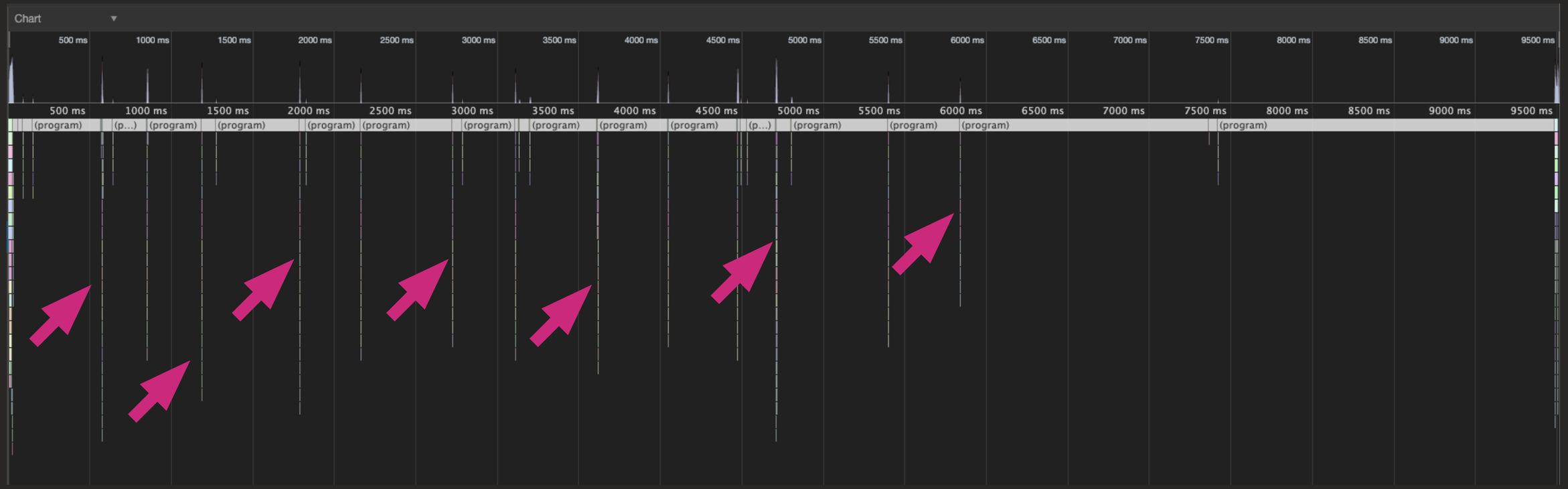
XHR & Promise Graph (Request)



XHR & Promise Graph (Response)



XHR & Async/Await Example



... tooling...


Debug All tests in test: blog
Debugger
Console
Interactive console
Frames
Variables
Thread 2 [slee...
articles_controller_test.rb:23
This is a debugger.
Yup.
It's extremely powerful.
Yup.
Coding will be much easier if you use it!
That makes sense to me.
Then use it!
console.log("beepboop")

```
console.profile
console.profileEnd
```

Example

```
function andFinally() {
  for (let i = 0; i < 1000000; i++) {
    // whut?!
  }

  console.log("done");
}

function andThenMore() {
  andFinally();
}

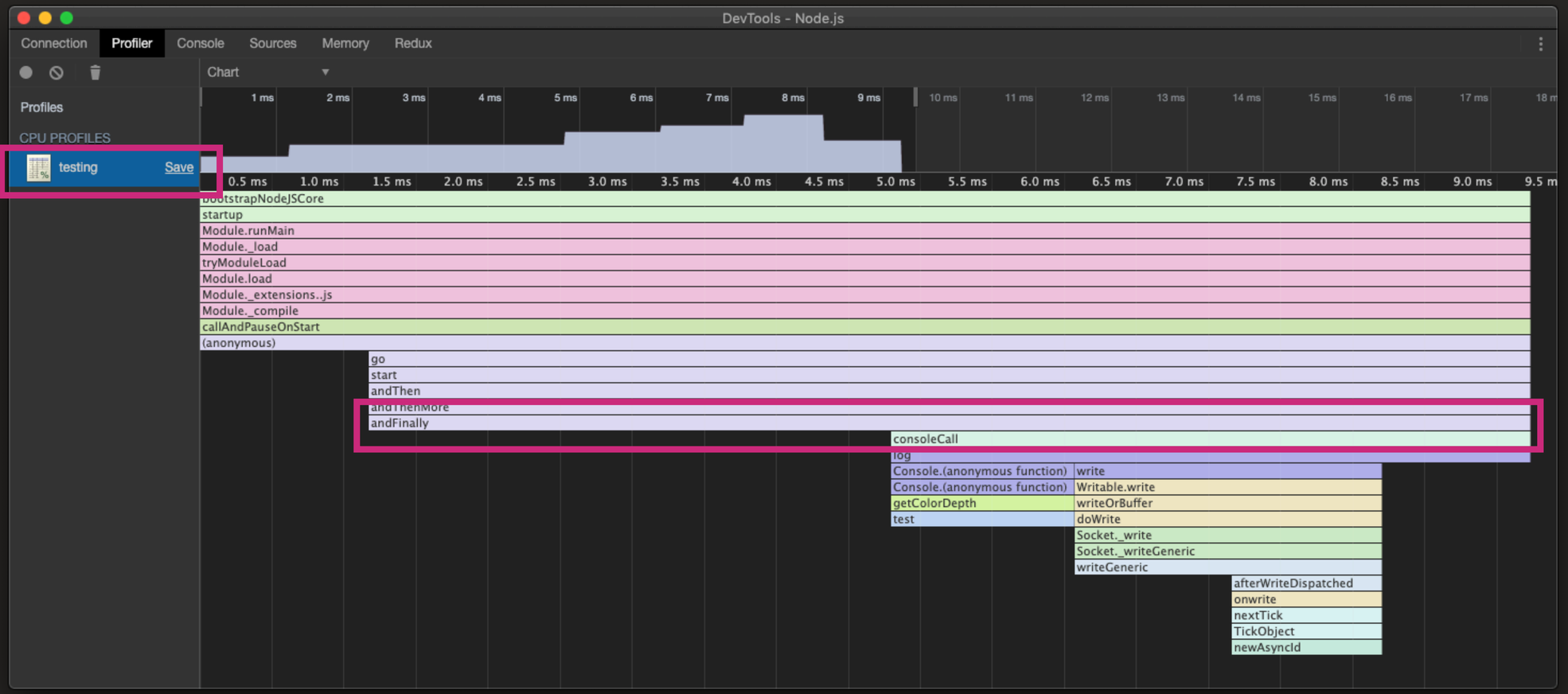
function andThen() {
  andThenMore();
}

function start() {
  andThen();
}

function go() {
  start();
}

console.profile("testing");
go();
console.profileEnd("testing");
```

Example Graph



Example

```
function andFinally() {
  for (let i = 0; i < 1000000; i++) {
    // whut?!
  }

  console.log("done");
}

function andThenMore() {
  andFinally();
}

function andThen() {
  andThenMore();
}

function start() {
  andThen();
}

function go() {
  start();
}

console.profile("testing");
go();
console.profileEnd("testing");
```

Moment of Truth

Open Source

Search or jump to... / Pull requests Issues Marketplace Explore



👀 building baller ass shit

Luke DeWitt
whatadewitt

Edit profile

Developer in Halifax, livin' the dream, buildin' cool(?) shit

@theREDspace

Halifax, NS

Overview | Repositories 41 | Projects 0 | Stars 117 | Followers 29 | Following 1

Popular repositories — Customize your pins

yahoo-fantasy-sports-api
NodeJS wrapper for the Yahoo! Fantasy Sports API
JavaScript ★ 95 23

yahoofantasysandbox
Fantasy Sandbox Source
HTML ★ 11 5

fantasyapp
Example Application
JavaScript ★ 6 1

symmetric-pandemic
JS based deck manager for symmetric pandemic games
★ 3 2

passport-yahoo-oauth
Forked from jaredhanson/passport-yahoo-oauth
Custom (updated?) Yahoo! (OAuth) authentication strategy for Passport and Node.js.
JavaScript ★ 3 3

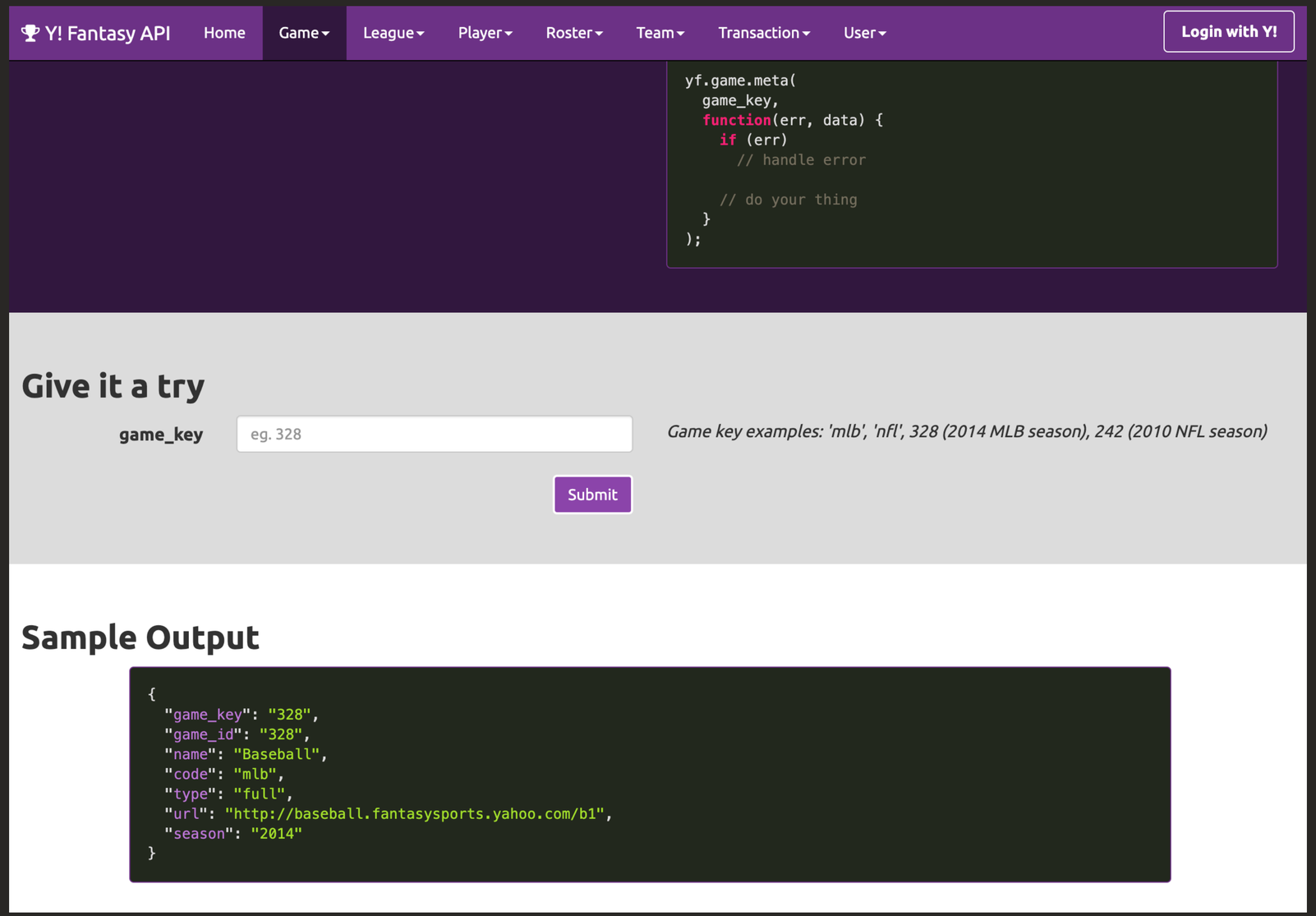
Luke-s-Node-Fun
simple node/socket.io fun
★ 1

https://dewey.codes

Project at a Glance

- Started building it in 2014
- Domain knowledge / interest existed
- Learning project
- Hard deadline
  - … rushed

The Sandbox


Y! Fantasy API
Home
Game
League
Player
Roster
Team
Transaction
User
Login with Y!
yf.game.meta(
game_key,
function(err, data) {
if (err)
// handle error
// do your thing
}
);
Give it a try
game_key
eg. 328
Game key examples: 'mlb', 'nfl', 328 (2014 MLB season), 242 (2010 NFL season)
Submit
Sample Output
{
"game_key": "328",
"game_id": "328",
"name": "Baseball",
"code": "mlb",
"type": "full",
"url": "http://baseball.fantasysports.yahoo.com/b1",
"season": "2014"
}

Current Code (API Call)

```
exports.getData = function(req, res) {
  var func;
  var yf = req.app.yf,
    resource = req.params.resource,
    subresource = req.params.subresource,
    query = req.query;

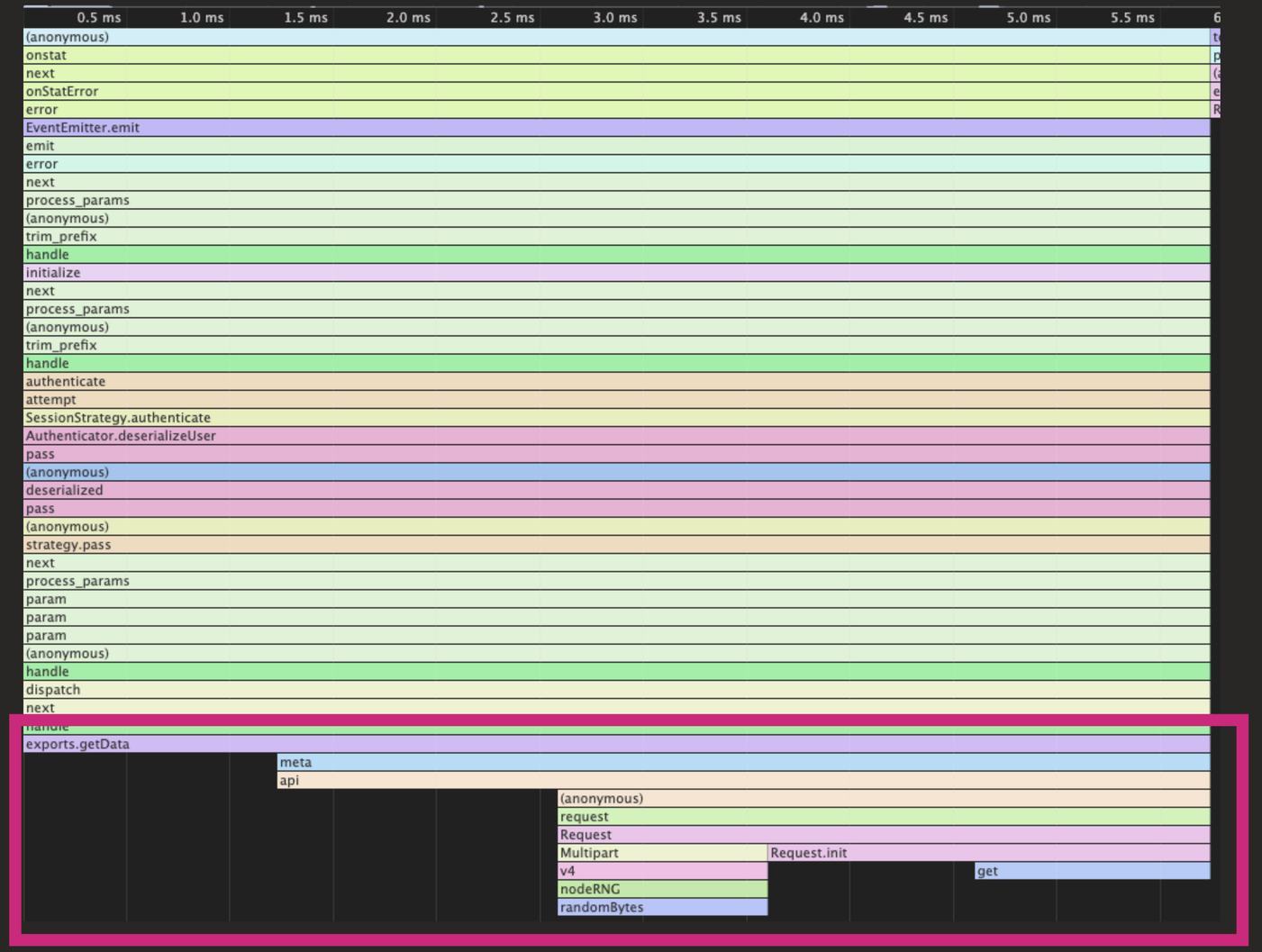
  /*
   * - make sure that user token is set (in case of server restart)
   * - map data for proper request
   */
  var args = Object.map(query);

  var callback = function callback(err, data) {
    console.profileEnd(`${resource}-${subresource}`);
    // handle success or error here...
  };

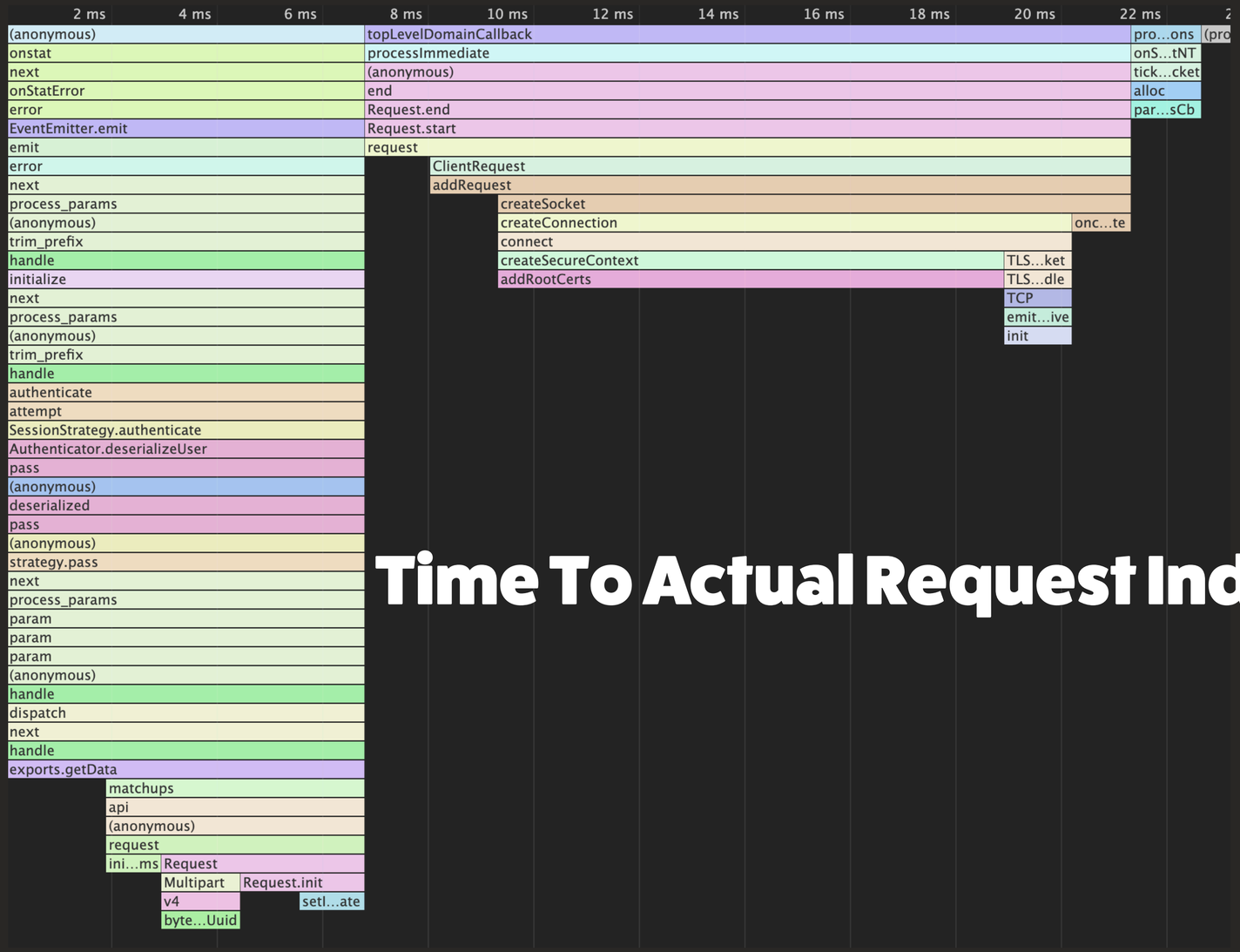
  // some work here
  args.push(callback);

  console.log(resource, subresource, args);
  console.profile(`${resource}-${subresource}`);
  yf[resource][subresource].apply(yf[resource], args);
};
```

Profile Graph (API Call)

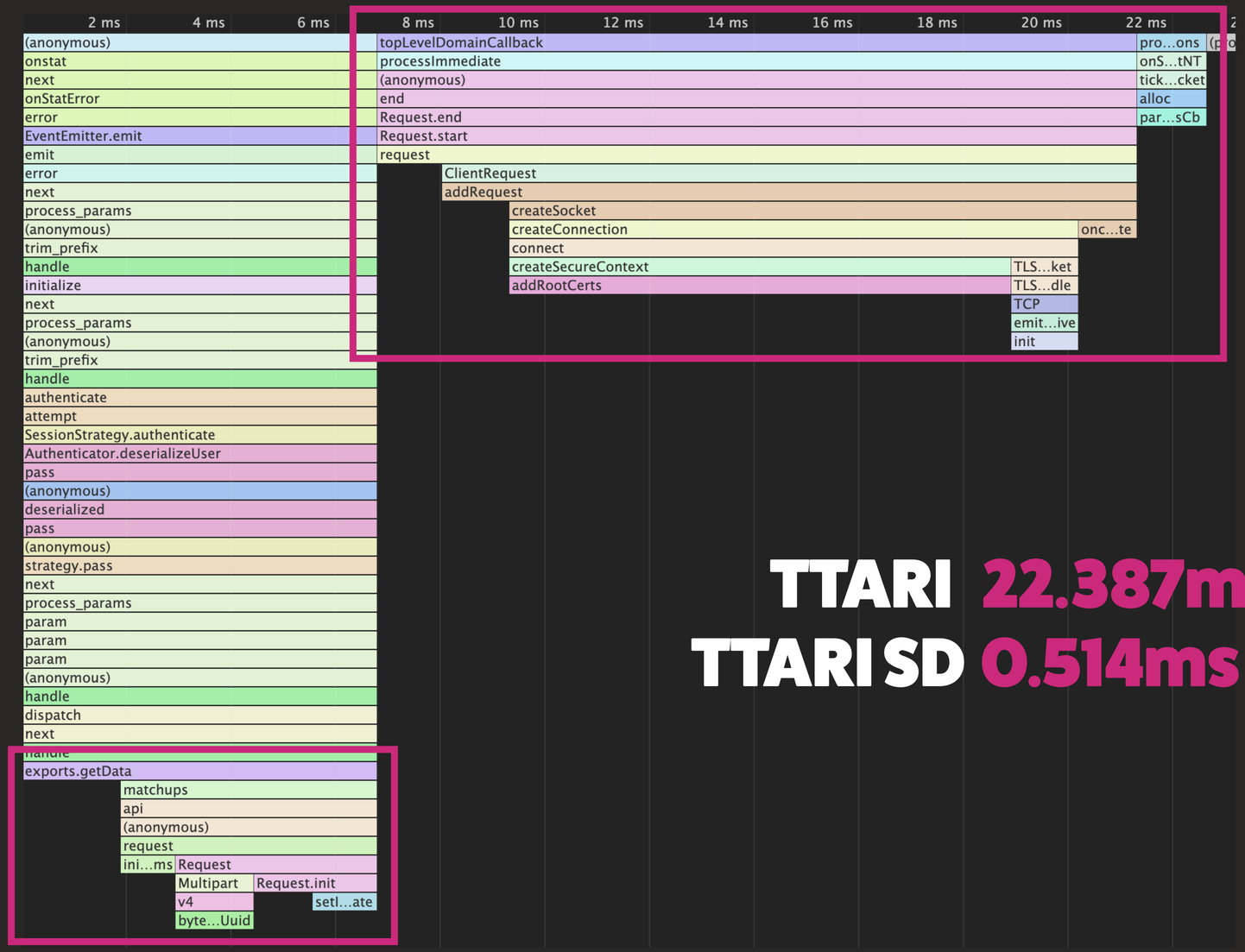


Profile Graph (API Call)



Time To Actual Request Index

Profile Graph (API Call)



TTARI 22.387ms

TTARI SD 0.514ms

Current Code (API Call)

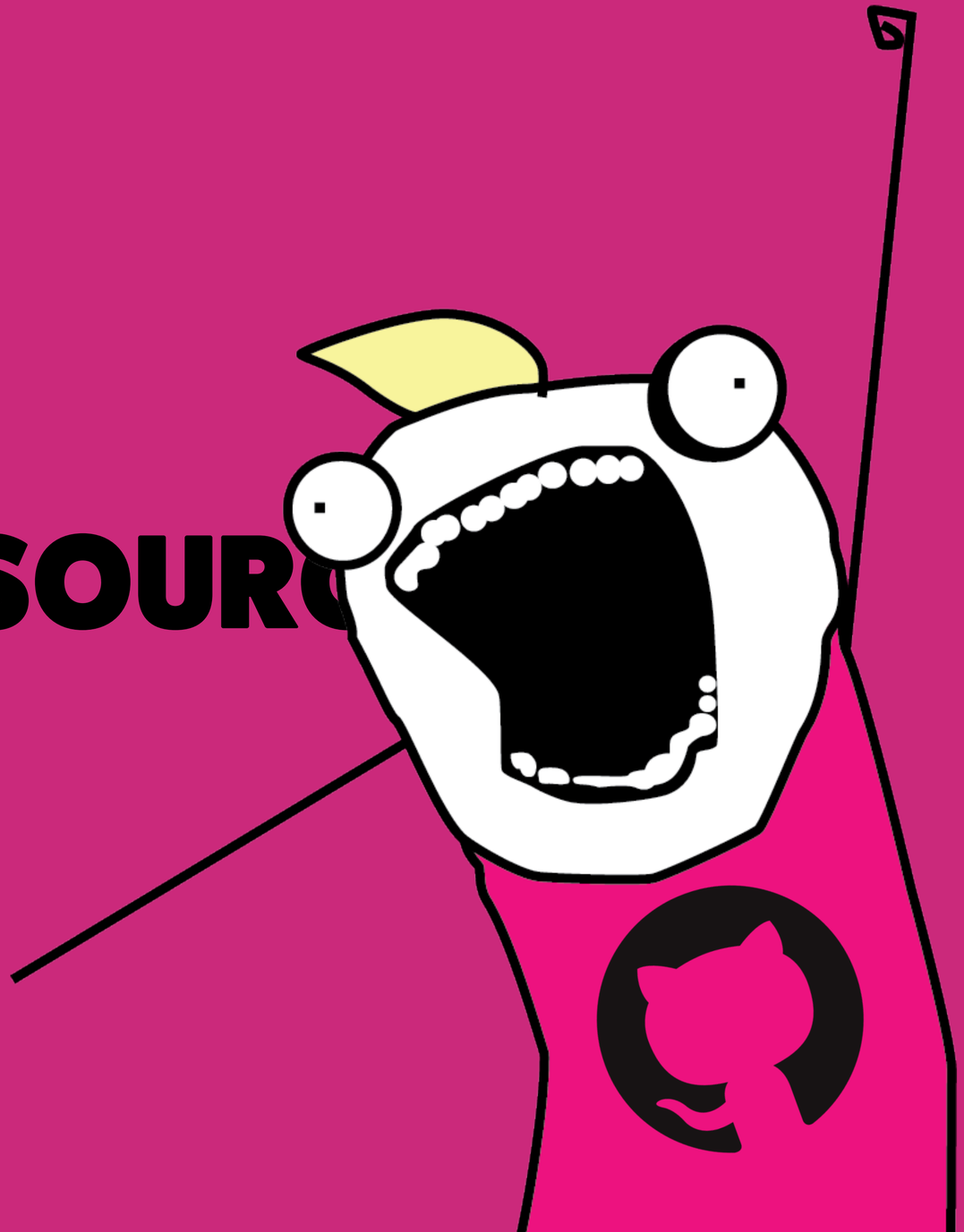
```
api(...args) {
  const method = args.shift();
  const url = args.shift();
  const cb = extractCallback(args);
  let postData = false;

  if (args.length) {
    postData = args.pop();
  }

  const options = {
    url: url,
    method: method,
    json: true,
    auth: {
      bearer: this.yahooUserToken
    }
  };

  return new Promise((resolve, reject) => {
    request(options, (e, body, data) => {
      const err = e || data.error;
      if (err) {
        reject(err);
        return cb(err);
      }
      resolve(data);
      return cb(null, data);
    });
  });
}
```

MOAR OPEN SOUR



Updated Code (API Call)

```
api(...args) {
  const method = args.shift();
  const url = args.shift();
  const cb = extractCallback(args);
  let postData = false;

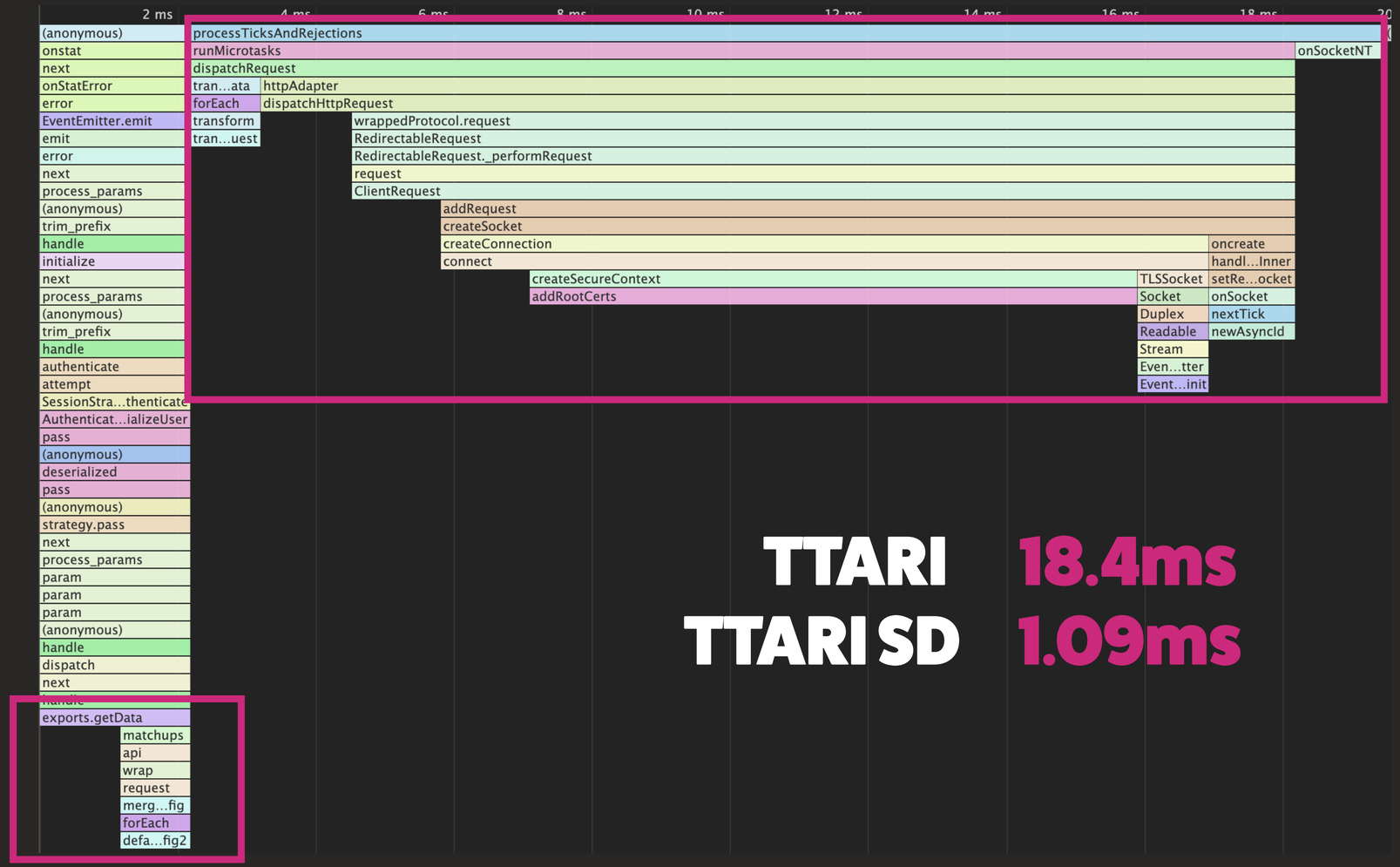
  if (args.length) {
    postData = args.pop();
  }

  const options = {
    url: url,
    method: method,
    headers: { Authorization: `bearer ${this.yahooUserToken}` }
  };

  return axios(options)
    .then(resp => {
      if (resp.data.err) {
        throw resp.data.err;
      }

      cb(null, resp.data);
      return resp.data;
    })
    .catch(e => {
      cb(e);
    });
}
```

Profile Graph (API Call w/ Axios)



TTARI 18.4ms

TTARI SD 1.09ms

TTARI

Request 22.387 ms

Axios 18.4 ms

18% Improvement

We can do better...

Vanilla Code (API Call)

```
api(...args) {
    const method = args.shift();
    const url = args.shift();
    const cb = extractCallback(args);
    let postData = false;

    if (args.length) {
      postData = args.pop();
    }

    const options = {
      hostname: "fantasysports.yahooapis.com",
      path: url.replace("https://fantasysports.yahooapis.com", ""),
      method: method,
      headers: {
        Authorization: ` Bearer ${this.yahooUserToken}`
      }
    };

    return new Promise((resolve, reject) => {
      https
        .request(options, resp => {
          let data = "";

          resp.on("data", chunk => {
            data += chunk;
          });

          resp.on("end", () => {
            data = JSON.parse(data);

            if (data.err) {
              cb(data.err);
              return reject(data.err);
            }

            cb(null, data);
            return resolve(data);
          });
        })
        .on("error", err => {
          cb(err.message);
          return reject(err.message);
        })
        .end();
    });
  }
```

Profile Graph (API Call w/ Vanilla JS)



TTARI 15.9ms

TTARI SD 0.535ms

TTARI

Request 22.387 ms

Axios 18.4 ms

Vanilla 15.9 ms

29% Improvement

TWENTY-NINE



Thank you for coming!

Thank You!

RED space™

redspace.com / **T** (902) 444.3490

LUKE DEWITT @whatadewitt

FACEBOOK REDspace
TWITTER @theREDspace
LINKEDIN The REDspace

Thank You!

Oh, by the way, we're hiring! https://www.redspace.com/jobs