Improving your applications with AbortController ✨

SIMON PLENDERLEITH

NODECONF REMOTE 2021

NodeConf Remote 2021

Oct. 19th, 2021

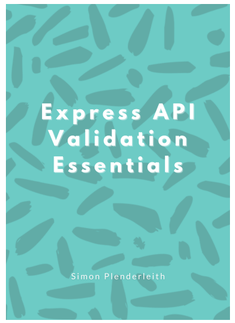
Slides and `code` at

simonplend.com/nodeconf2021

Hi, I'm Simon 👋

Independent **Node.js consultant and educator**

Help developers level up with Node.js at simonplend.com/blog



Author of **Express API Validation Essentials**
expressapivalidation.com

@simonplend

simonplend

simonplend.com

What’s on the menu?

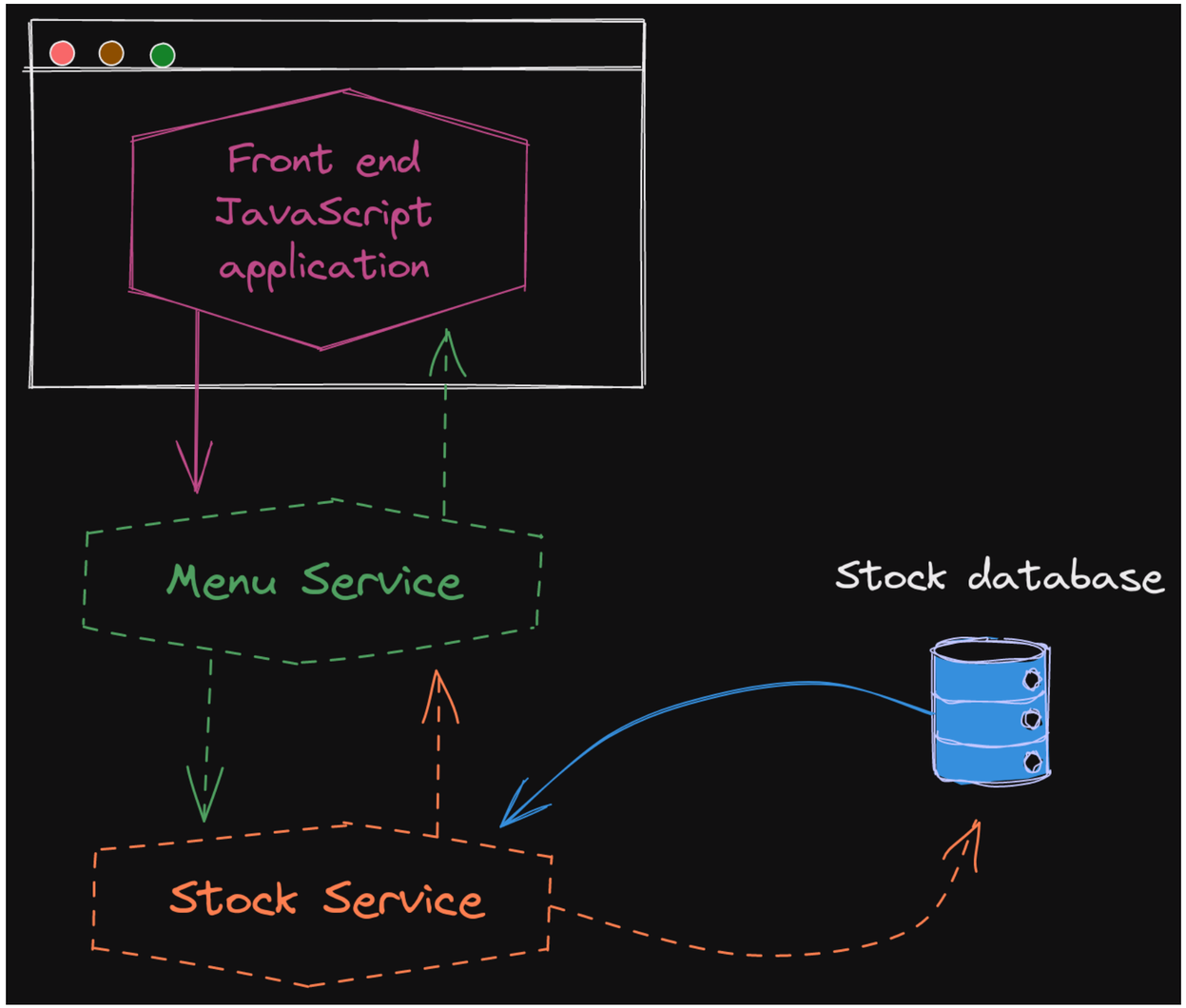
- Problems at Ice Cream Corp
- Performance testing
- Menu Service v2… ?
- The Abort API
- Menu Service v3
- Abort API support in Node.js

Photo by Markus Spiske on Unsplash

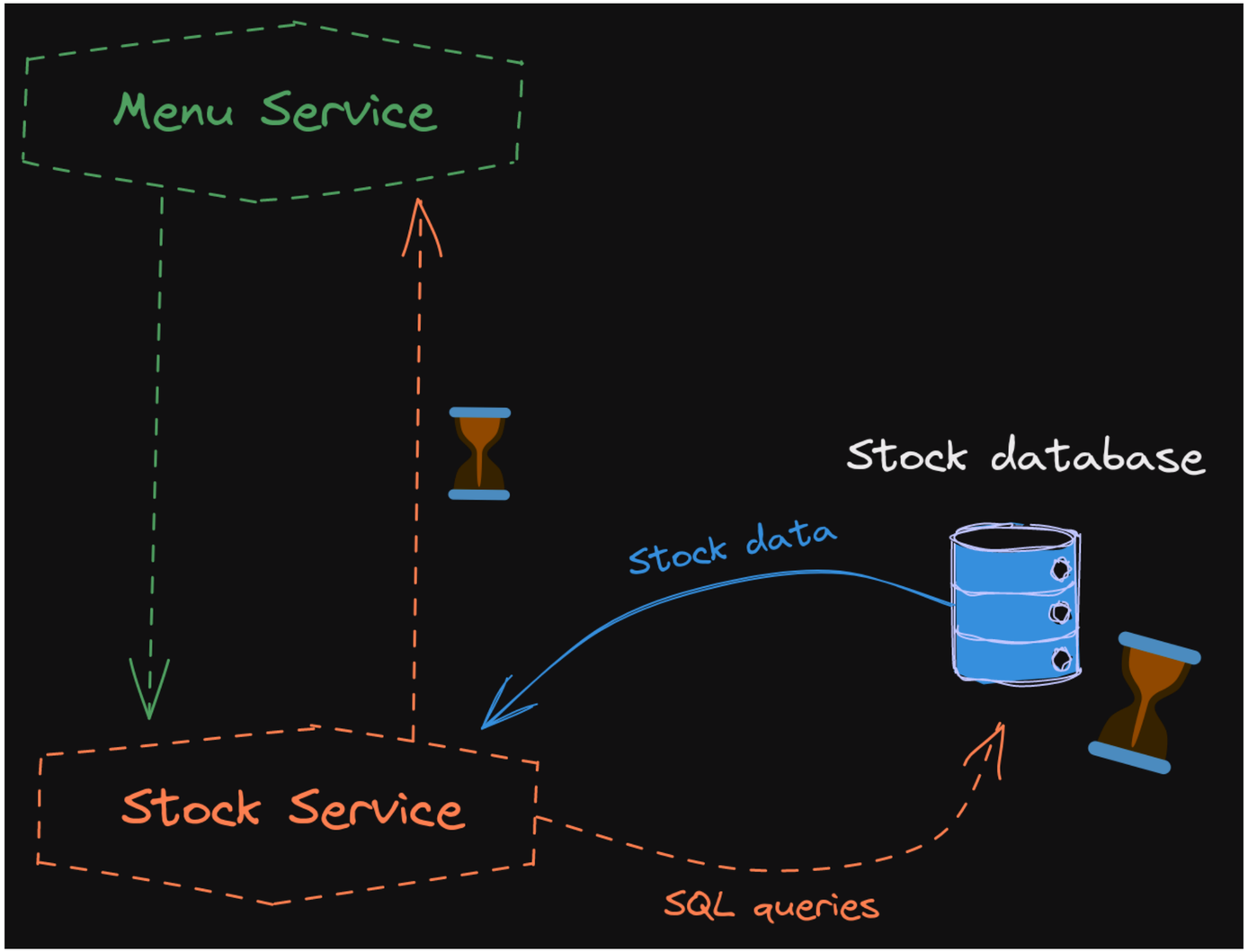
Problems at Ice Cream Corp 

Photo by Pawel Janiak on Unsplash

Ice Cream Corp Architecture



The Unpredictable Stock Service



The Unpredictable Stock Service: Code

```
import Fastify from "fastify";

const app = Fastify();

app.get("/stock", function (request, reply) {
  const stock = [
    { id: "cf93d53f-7f60-4c9d-a543-ef217a09269b", name: "Cookies 'n' Cream", stock: 8532 },
    { id: "41373bb6-e6fb-4606-8f01-1cd24e228939", name: "Neopolitan", stock: 0 },
    { id: "8e884a8d-ae04-4518-80b7-377ee5bcecc6", name: "Pistachio", stock: 6437 }
  ];

  const delay = Math.floor(Math.random() * 10000);

  setTimeout(() => {
    reply.send({ stock, delay });
  }, delay);
});

await app.listen(4000);
```

Menu Service v1: Code

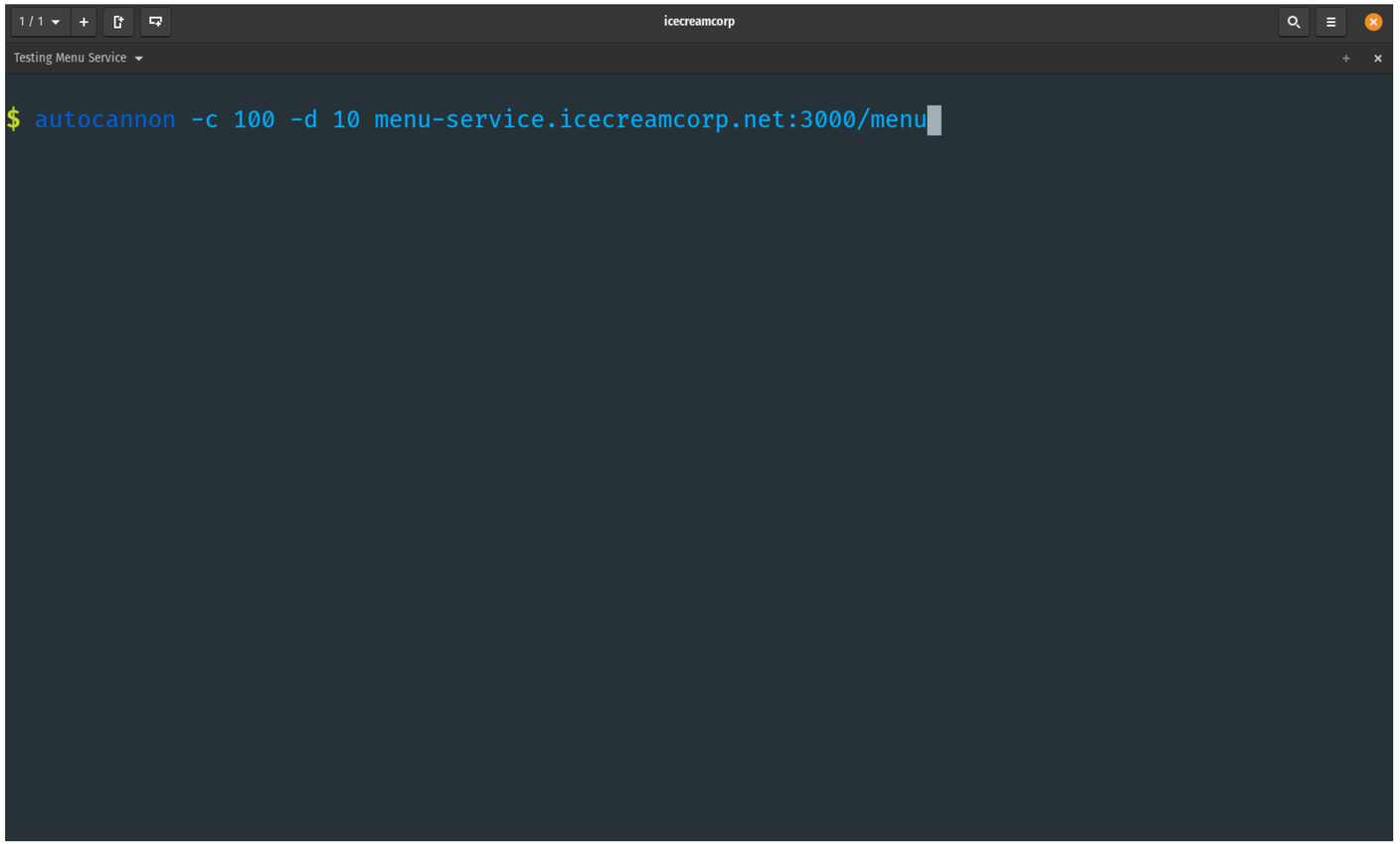
```
import fetch from "node-fetch";
import Fastify from "fastify";

async function makeRequest(url) {
  const response = await fetch(url);

  const responseData = await response.json();

  return responseData;
}

const app = Fastify();

app.get("/menu", async function (request, reply) {
  const stockServiceEndpoint = "http://stock-service.icecreamcorp.net:4000/stock";

  const stockServiceResponse = await makeRequest(stockServiceEndpoint);

  reply.send({ stockServiceResponse });
});

await app.listen(3000);
```

Performance testing

Load testing and profiling the Menu Service

AutoCannon

Clinic.js



Menu Service v1: Behaviour

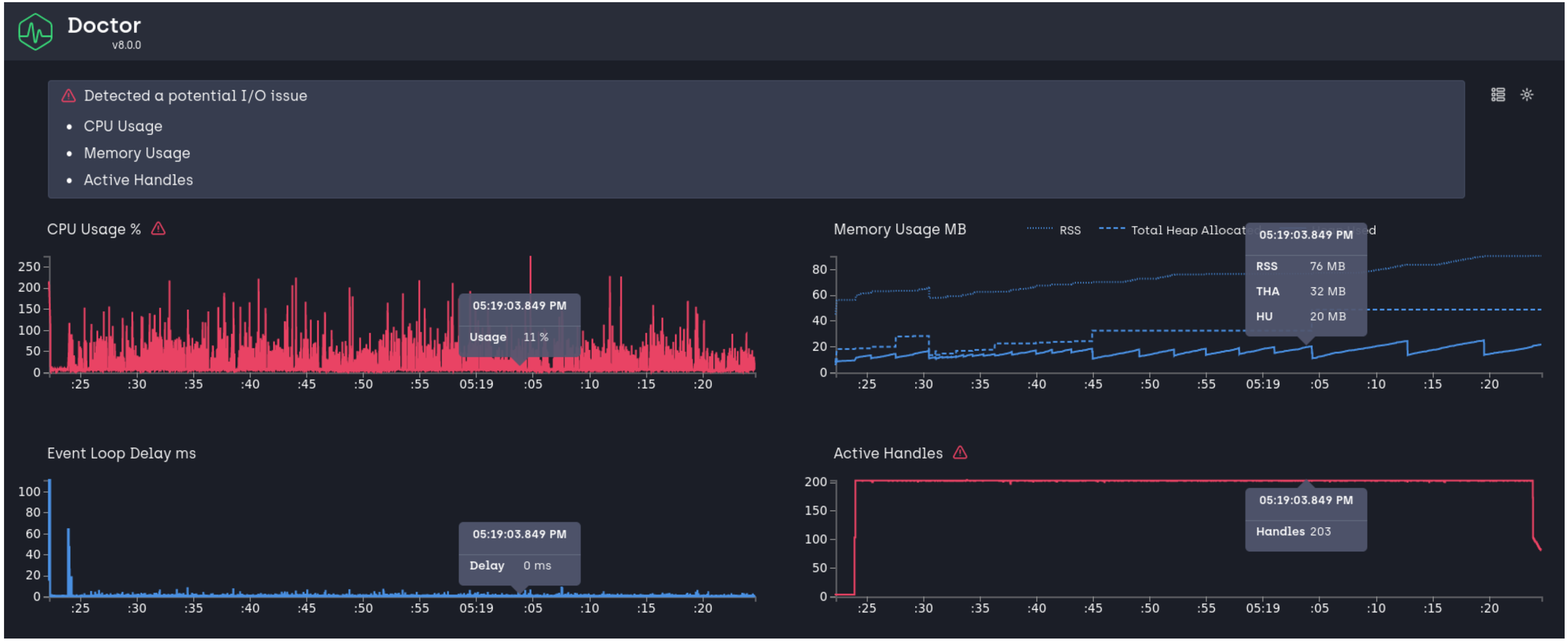
```
Running 60s test @ http://menu-service.icecreamcorp.net:3000/menu
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg *	Stdev	Max
Latency	244 ms	4862 ms	9718 ms	9900 ms	4898.42 ms	2941.22 ms	10000 ms

Stat	1%	2.5%	50%	97.5%	Avg *	Stdev	Min
Req/Sec	6	8	19	27	19.15	4.63	6

```
Req/Bytes counts sampled once per second.

1k requests in 60.03s, 539 kB read
3 errors (3 timeouts)
```

Menu Service v1: Behaviour



How can we improve the Menu Service?

Menu Service v2... ?

Menu Service v2: Behaviour

```
Running 60s test @ http://menu-service.icecreamcorp.net:3000/menu
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg *	Stdev	Max
Latency	253 ms	1004 ms	1024 ms	1146 ms	957.63 ms	176.57 ms	1217 ms

Stat	1%	2.5%	50%	97.5%	Avg *	Stdev	Min
Req/Sec	13	100	105	112	103.47	12.1	13

```
Req/Bytes counts sampled once per second.

612 2xx responses, 5596 non 2xx responses
6k requests in 60.03s, 1.97 MB read
```

Menu Service v2: Behaviour

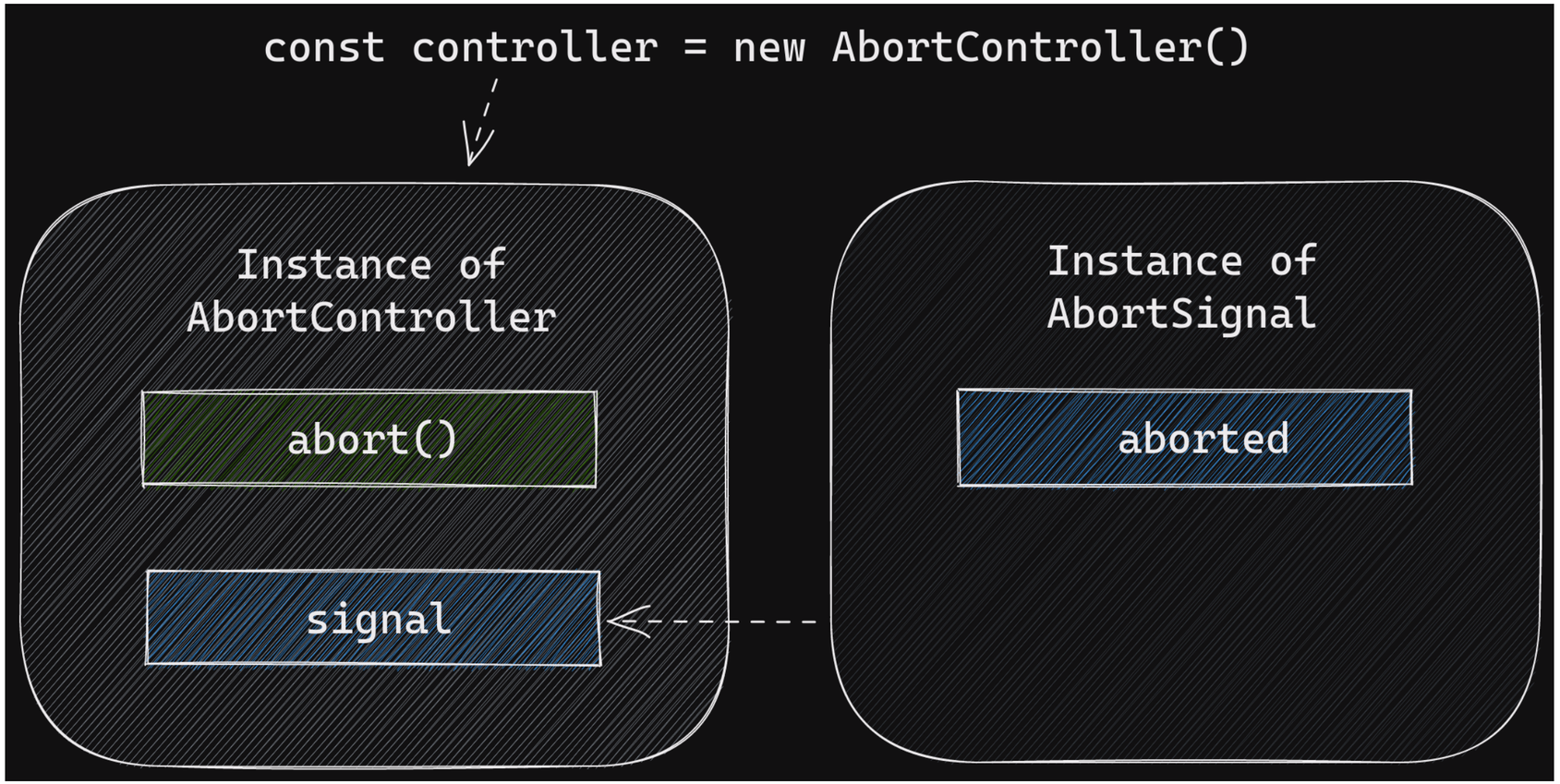
What now?!

The Abort API

The Abort API

A JavaScript API which consists of two classes:

`AbortController` and `AbortSignal`



The Abort API in action

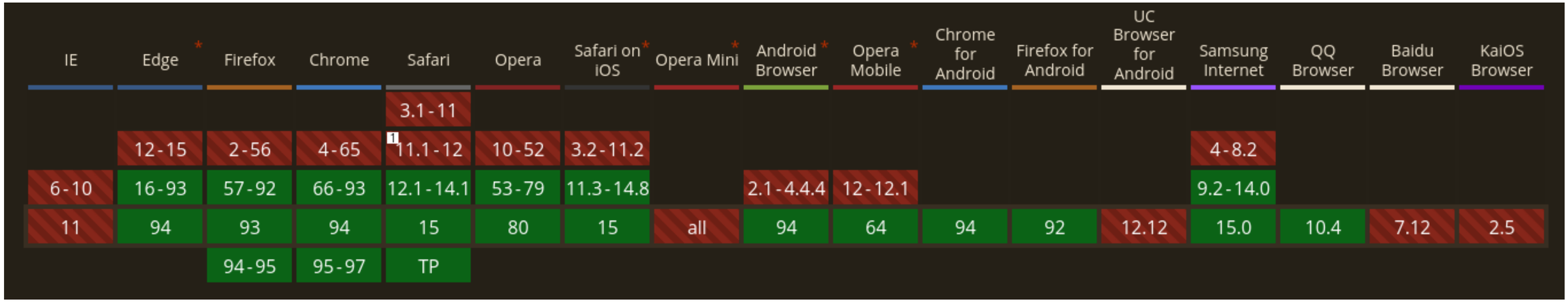
```
const controller = new AbortController();
const signal = controller.signal;

signal.addEventListener("abort", () => {
  console.log("The abort signal was triggered");
}, { once: true });

controller.abort();
```

Support for the Abort API

The Abort API originated in the Web Platform

Microsoft Edge 16 was the first browser to implement the Abort API in October 2017.

Now all major browsers support it.



Stable **support in Node.js since v15.4.0**, released in December 2020.

Source: caniuse.com

What benefits can we get from cancelling async operations?

- Greater control over the behaviour of our application
- Free up system or network resources the application is using
- Ability to "fail fast"
- Potential improvement to application performance

Photo by David Becker on Unsplash

Cancelling an HTTP request with an `AbortSignal`

```
import fetch from "node-fetch";

const cancelRequest = new AbortController();

fetch("https://jsonplaceholder.typicode.com/posts", {
  signal: cancelRequest.signal
})
  .then(response => response.json())
  .then(responseData => console.log(responseData))
  .catch(error => console.error(error));

cancelRequest.abort();
```

```
$ node node-fetch-with-abortsignal.js

file:///dev/node_modules/node-fetch/src/index.js:56
            const error = new AbortError('The operation was aborted.');
                          ^


AbortError: The operation was aborted.
    at abort (file:///dev/node_modules/node-fetch/src/index.js:56:18)
    at EventTarget.abortAndFinalize (file:///dev/node_modules/node-fetch/src/index.j
    at EventTarget.[nodejs.internal.kHybridDispatch] (node:internal/event_target:562
    at EventTarget.dispatchEvent (node:internal/event_target:504:26)
    at abortSignal (node:internal/abort_controller:97:10)
    at AbortController.abort (node:internal/abort_controller:122:5)
    at file:///dev/abort-api/node-fetch-with-abortsignal.js:12:16
    at ModuleJob.run (node:internal/modules/esm/module_job:183:25)
    at async Loader.import (node:internal/modules/esm/loader:178:24)
    at async Object.loadESM (node:internal/process/esm_loader:68:5) {
  type: 'aborted'
}
```

Menu Service v3

Menu Service v3: Behaviour

```
Running 60s test @ http://menu-service.icecreamcorp.net:3000/menu
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg *	Stdev	Max
Latency	282 ms	1006 ms	1116 ms	1167 ms	967.9 ms	172.98 ms	1251 ms

Stat	1%	2.5%	50%	97.5%	Avg *	Stdev	Min
Req/Sec	4	78	104	111	102.57	13.55	4

```
Req/Bytes counts sampled once per second.

614 2xx responses, 5540 non 2xx responses
6k requests in 60.03s, 1.94 MB read
```

Menu Service v3: Behaviour

⏭️ Next steps

- Capture metrics of how many requests we’re aborting
- Back off making requests to the Stock Service
- Prioritise improving the Stock Service

Abort API support in Node.js

Compatibility in Node.js versions

	v16.x	>= v14.17.0	Older versions
`AbortController` + `AbortSignal` classes	✅	Experimental	Use abort-controller package
Node.js core APIs which accept `AbortSignal`	✅	✅	❌

Node.js >= v14.17.0

Enable abort classes with flag e.g. `node --experimental-abortcontroller server.js`

OR `npm install abort-controller` and:

```
import AbortController from "abort-controller";
```

Support for `AbortSignal` in Node.js core API methods

child_process

- `child_process.exec`
- `child_process.execFile`
- `child_process.fork`
- `child_process.spawn`

dgram

- `dgram.createSocket`

events

- `events.on`
- `events.once`

fs

- `fs.readFile`
- `fs.watch`
- `fs.writeFile`

http

- `http.request`
- `https.request`

http2session

- `http2Session.request`

timers/promises

- `timers/promises.setImmediate`
- `timers/promises.setTimeout`

readline

- `readline.Interface`
- `readline.createInterface`

📦 Libraries with support for `AbortSignal`

HTTP libraries

- Node Fetch
- Undici
- Axios - v0.22.0 (released on Oct 1st 2021) - Details in README

Other popular libraries

- Piscina - The Node.js worker pool
- AWS SDK for JavaScript - The official AWS SDK
- dockerode - Node.js module for Docker’s Remote API

With thanks to... 🙏

- James Snell for sharing the `Promise.race()` setTimeout pattern
- For review and feedback on this talk:
  - Luciano Mammino
  - Liam Keaton
  - Kevin Cunningham
  - Nick Ramsbottom
- For sharing libraries with support for `AbortSignal`:
  - Tim Perry
  - James Snell
  - Trivikram Kamat

Thank you! 💫

🍨 Slides and `code` at simonplend.com/nodeconf2021

@simonplend

simonplend

simonplend.com

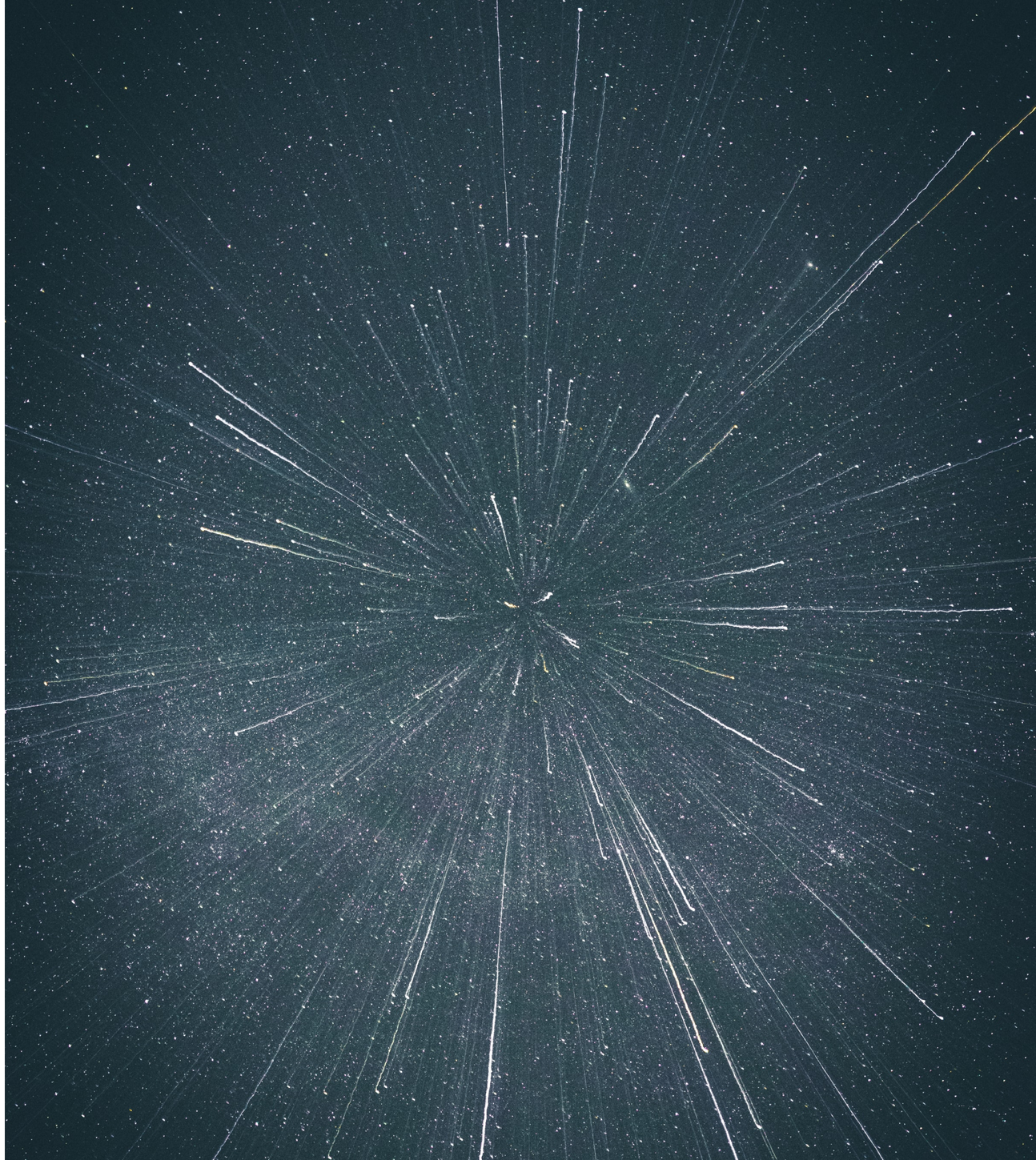


Photo by Casey Horner on Unsplash