h2-node or Implementing HTTP/2 Server with Node and Express

Slides 👓 📄 🖥️

Everything (PDF+Markdown): https://github.com/azat-co/h2-node

Better Apps—Better Life

Big idea: Web, HTTP and JavaScript are everywhere... and Node has some cool core features... What if the world can be a better place if more developers master Node?

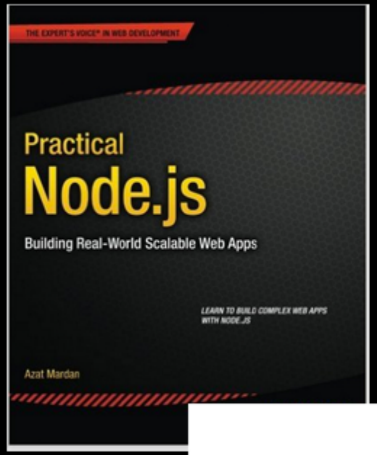
About Presenter

Capital One in Top 10 US Banks

5 HACKS TO GETTING THE JOB OF YOUR DREAMS TO LIVE A HAPPIER & HEALTHIER LIFE
BY AZAT MARDAN
2015

PROGWRITER [2.0: BEYOND BOOKS]
SECRETS of a successful online course creator and other INCOME strategies – that REALLY work
by Azat Mardan

PROGWRITER [PROGRAMMER + WRITER]
LESSONS learned on my path from ordinary developer to WRITER of multiple programming books – THAT SELL
by Azat Mardan

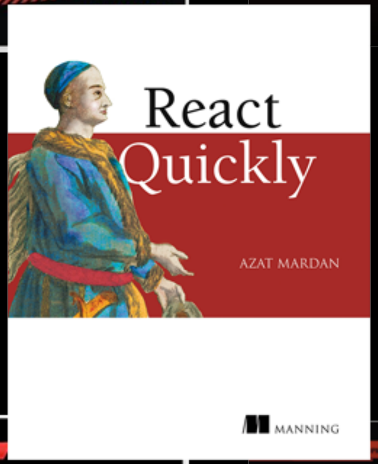

THE EXPERT'S VOICE® IN WEB DEVELOPMENT
Practical Node.js
Building Real-World Scalable Web Apps
LEARN TO BUILD COMPLEX WEB APPS WITH NODE.JS
Azat Mardan

THE EXPERT'S VOICE®
Full Stack JavaScript
Learn Backbone.js, Node.js and MongoDB
—
Azat Mardan
Apress®

INTRODUCTION TO OAUTH WITH NODE.JS

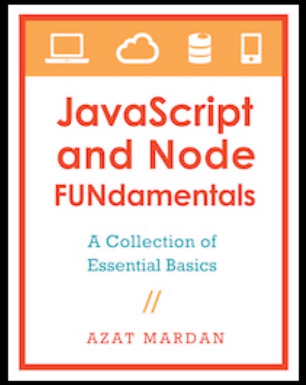
Start learning Backbone.js, Node.js and MongoDB
[AZAT MARDAN]
RAPID PROTOTYPING WITH JS
AGILE JAVASCRIPT DEVELOPMENT

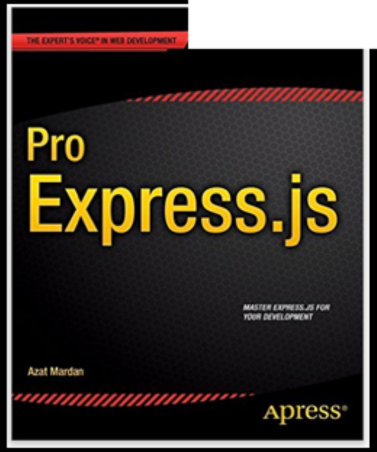
THE COMPREHENSIVE BOOK ON EXPRESS.JS
EXPRESS.JS GUIDE
AZAT MARDAN

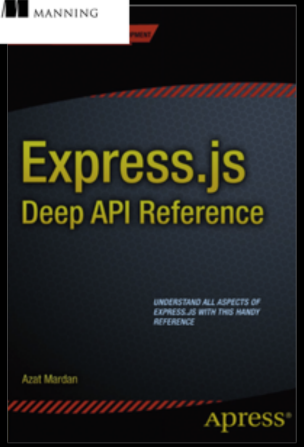

React Quickly
AZAT MARDAN
MANNING

OH MY JS
THE BEST JAVASCRIPT ARTICLES
</>

Начинай изучать Backbone.js, Node.js и MongoDB
[АЗАТ МАРДАН]
Перевод на русский Artod
БЫСТРОЕ ПРОТОТИПИРОВАНИЕ с JS
ГИБКАЯ РАЗРАБОТКА НА JAVASCRIPT


JavaScript and Node FUNdamentals
A Collection of Essential Basics
//
AZAT MARDAN


THE EXPERT'S VOICE® IN WEB DEVELOPMENT
Pro Express.js
MASTER EXPRESS.JS FOR YOUR DEVELOPMENT
Azat Mardan
Apress®


Express.js Deep API Reference
UNDERSTAND ALL ASPECTS OF EXPRESS.JS WITH THIS HANDY REFERENCE
Azat Mardan
Apress®

About Presenter

- Work: Technology Fellow at Capital One (kind of a big deal)
- Experience: FDIC, NIH, DocuSign, HackReactor and Storify
- Books: React Quickly, Practical Node.js, Pro Express.js, Express.js API and 8 others

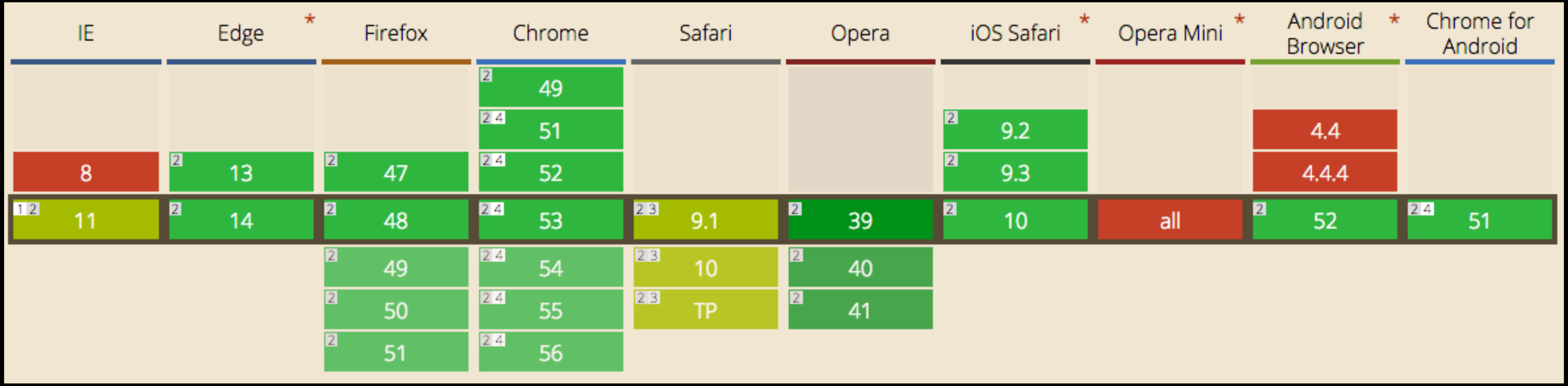
Azat Mardan

- Teach: NodeProgram.com
- Twitter: @azat_co
- Email: hi@azat.co
- Blog: webapplog.com

HTTP/2

It's here.

Really is here



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			🟩 [2] 49						
			🟩 [2] [4] 51			🟩 [2] 9.2		🟥 4.4	
🟥 8	🟩 [2] 13	🟩 [2] 47	🟩 [2] [4] 52			🟩 [2] 9.3		🟥 4.4.4	
🟨 [1] [2] 11	🟩 [2] 14	🟩 [2] 48	🟩 [2] [4] 53	🟨 [2] [3] 9.1	🟩 [2] 39	🟩 [2] 10	🟥 all	🟩 [2] 52	🟩 [2] [4] 51
		🟩 [2] 49	🟩 [2] [4] 54	🟨 [2] [3] 10	🟩 [2] 40				
		🟩 [2] 50	🟩 [2] [4] 55	🟨 [2] [3] TP	🟩 [2] 41				
		🟩 [2] 51	🟩 [2] [4] 56						

http://caniuse.com/#feat=http2

History of H2

Started at SPDY at Google

Benefits of H2

Multiplexing

Server push

Stream priority

Header compression

Encryption*

- De facto mandatory encryption

TL;DR

Old methods of HTTP/1 might not work and might even harm!

Let's Get Our Hands Dirty

SSL Key+Cert

```
$ mkdir http2-express
$ cd http2-express
$ openssl genrsa -des3 -passout pass:x -out server.pass.key 2048
...
$ openssl rsa -passin pass:x -in server.pass.key -out server.key
writing RSA key
$ rm server.pass.key
$ openssl req -new -key server.key -out server.csr
...
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
...
A challenge password []:
...
$ openssl x509 -req -sha256 -days 365 -in server.csr -signkey server.key -out server.crt
```

H/2 and Node

- spdy - like
- `http2` - not like
- core `http2` (based on `nghttp2`) - coming! (GitHub issue)

spdy

```
npm init
npm i express spdy -S
```

```
const port = 3000
const spdy = require('spdy')
const express = require('express')
const path = require('path')
const fs = require('fs')

const app = express()
```

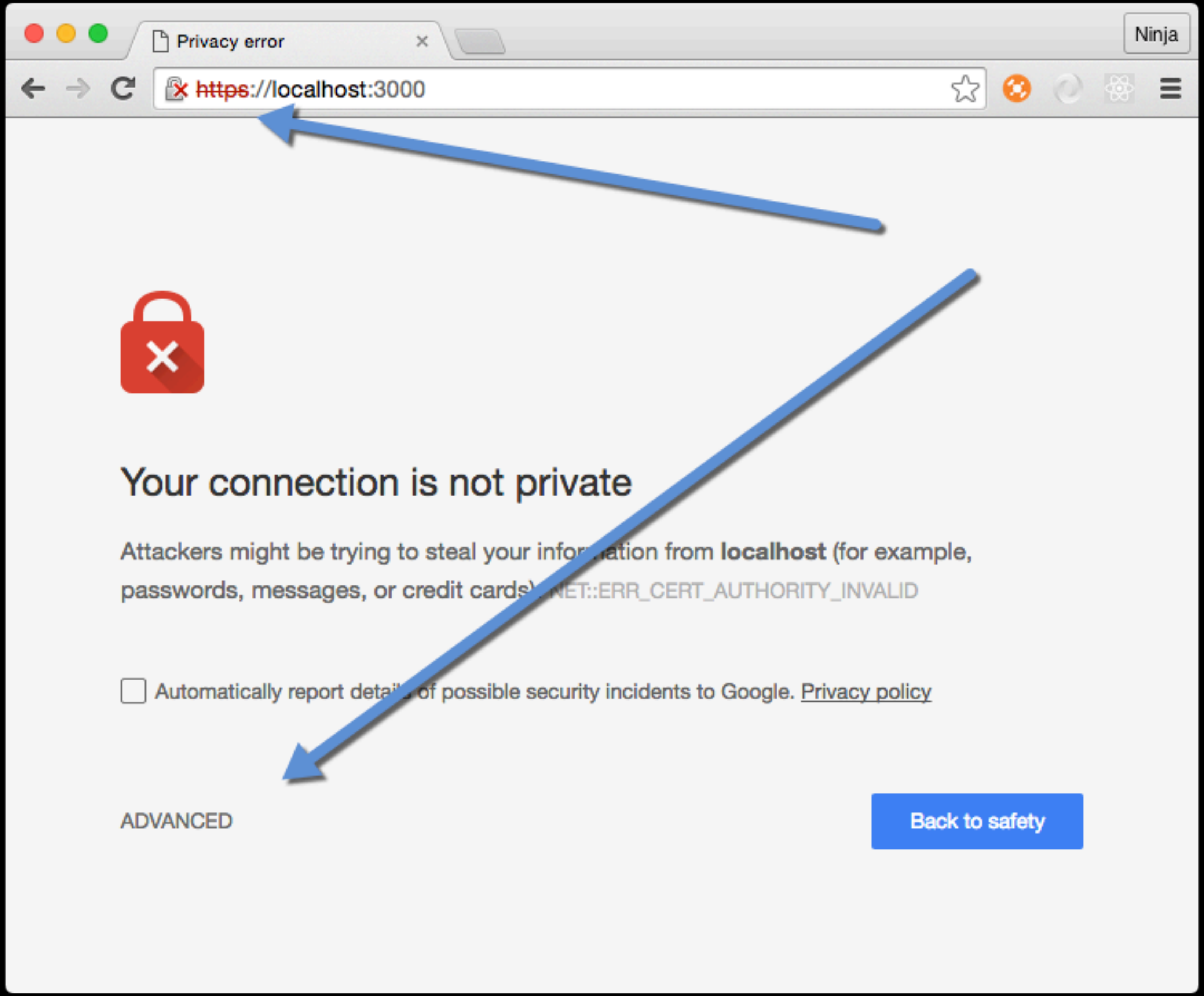
```
app.get('*', (req, res) => {
    res
      .status(200)
      .json({message: 'ok'})
})
```

```
const options = {
    key: fs.readFileSync(__dirname + '/server.key'),
    cert:  fs.readFileSync(__dirname + '/server.crt')
}
```

```
spdy
  .createServer(options, app)
  .listen(port, (error) => {
    if (error) {
      console.error(error)
      return process.exit(1)
    } else {
      console.log('Listening on port: ' + port + '.')
    }
  })
```

Start

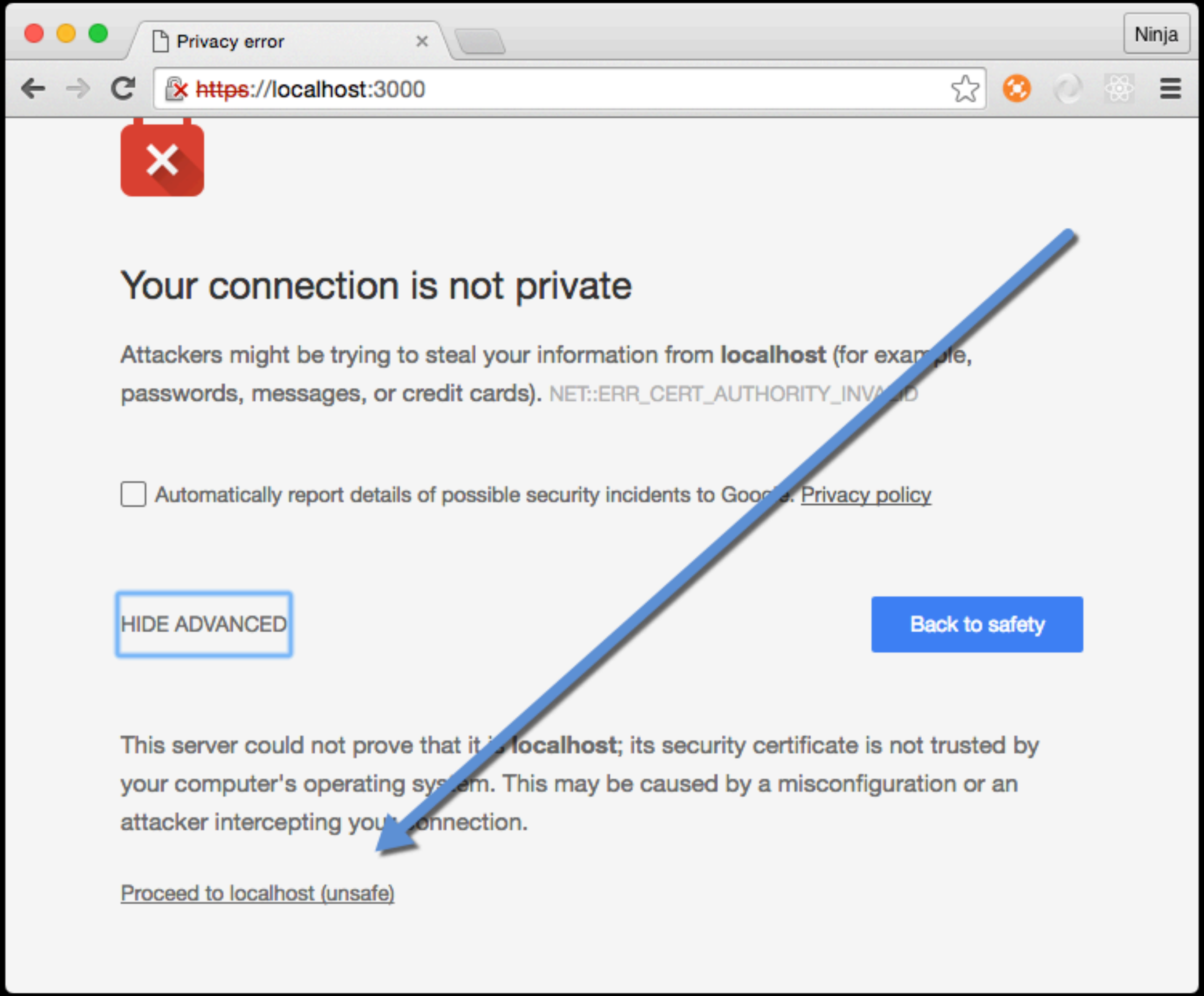
```
node server
```



Your connection is not private

Attackers might be trying to steal your information from **localhost** (for example, passwords, messages, or credit cards). NET::ERR_CERT_AUTHORITY_INVALID

Automatically report details of possible security incidents to Google. Privacy policy



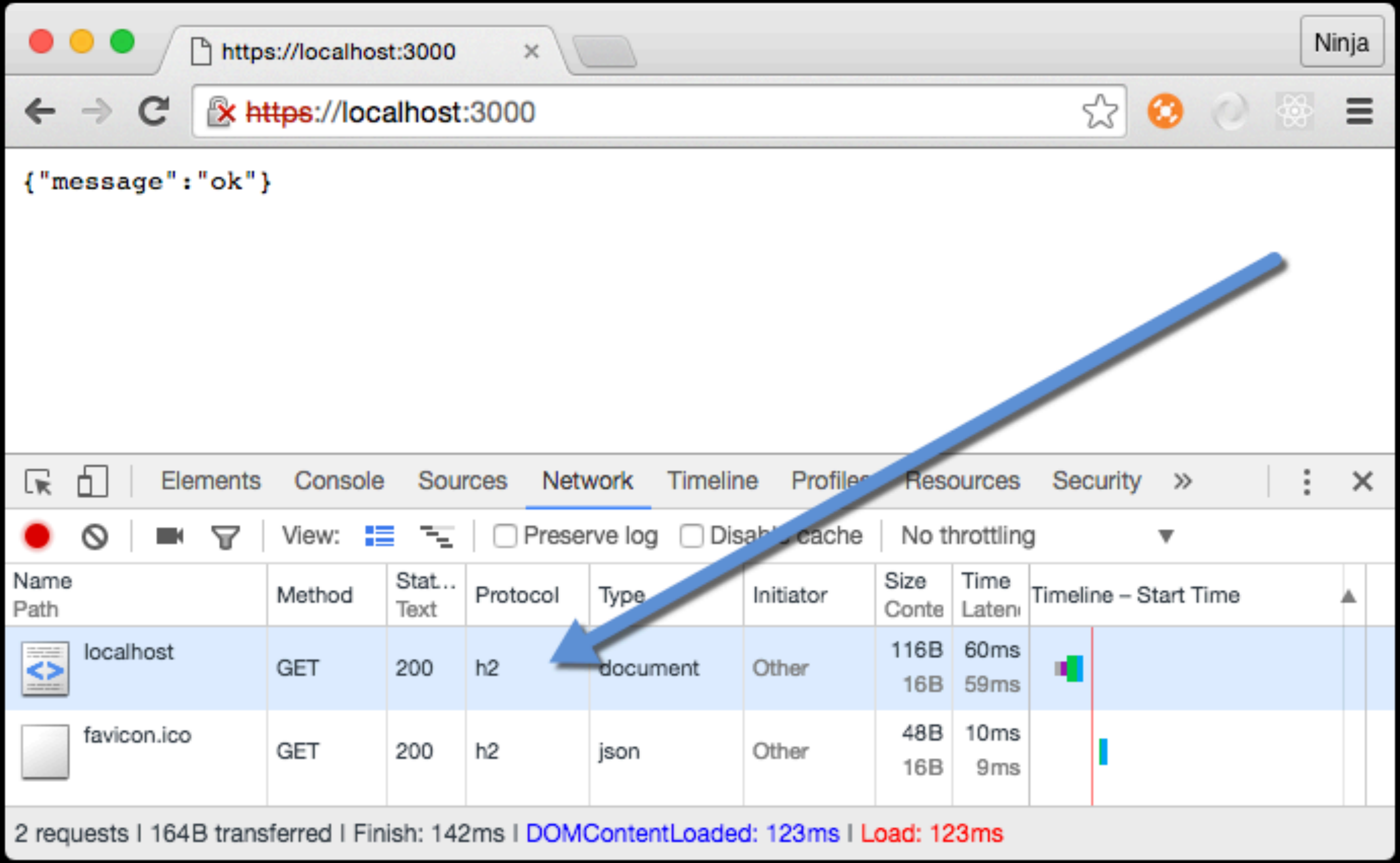
Your connection is not private

Attackers might be trying to steal your information from **localhost** (for example, passwords, messages, or credit cards). NET::ERR_CERT_AUTHORITY_INVALID

Automatically report details of possible security incidents to Google. Privacy policy

This server could not prove that it is **localhost**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

Proceed to localhost (unsafe)



```
{"message":"ok"}
```

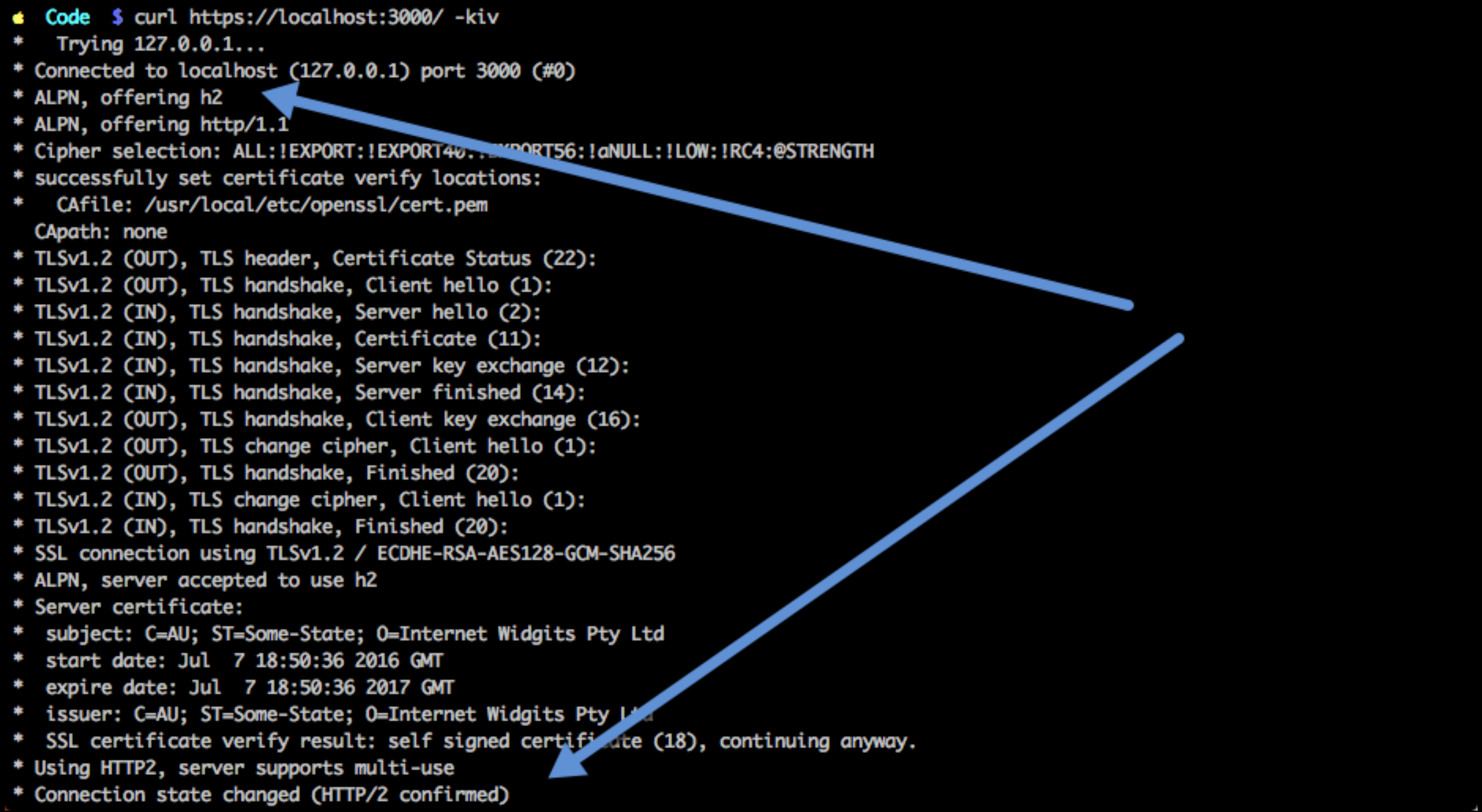
Name Path	Method	Stat... Text	Protocol	Type	Initiator	Size Conte	Time Laten	Timeline – Start Time
localhost	GET	200	h2	document	Other	116B 16B	60ms 59ms	
favicon.ico	GET	200	h2	json	Other	48B 16B	10ms 9ms	

2 requests | 164B transferred | Finish: 142ms | DOMContentLoaded: 123ms | Load: 123ms

Using CURL

```
curl https://localhost:3000/ -k
```

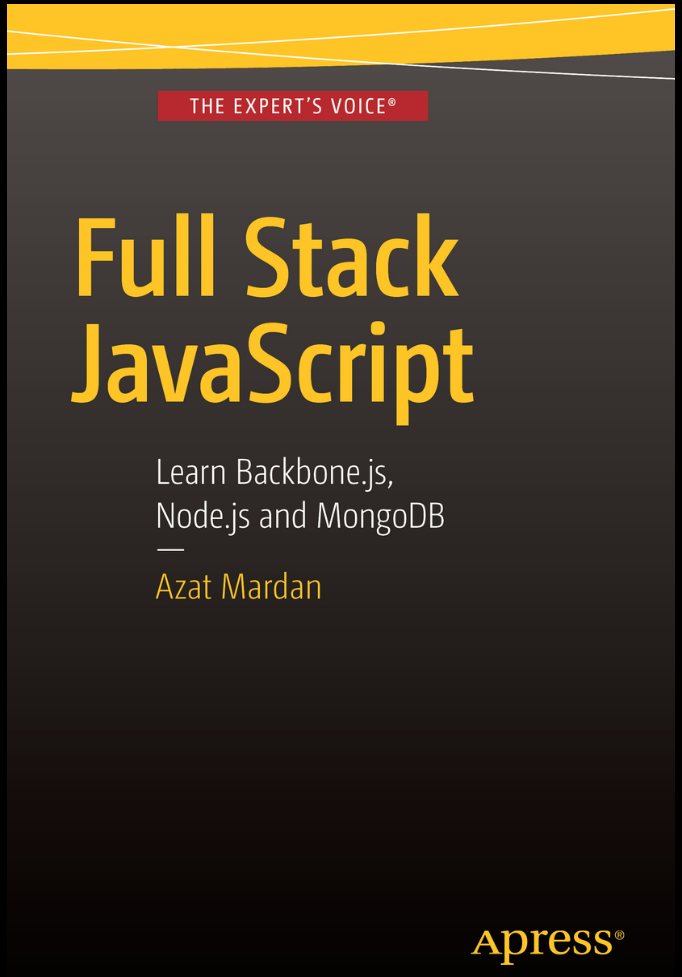
Make sure you got the latest version 7.46 with nghttp2)



```
 Code $ curl https://localhost:3000/ -kiv
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 3000 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /usr/local/etc/openssl/cert.pem
  CApath: none
* TLSv1.2 (OUT), TLS header, Certificate Status (22):
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES128-GCM-SHA256
* ALPN, server accepted to use h2
* Server certificate:
*  subject: C=AU; ST=Some-State; O=Internet Widgits Pty Ltd
*  start date: Jul  7 18:50:36 2016 GMT
*  expire date: Jul  7 18:50:36 2017 GMT
*  issuer: C=AU; ST=Some-State; O=Internet Widgits Pty Ltd
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Using HTTP2, server supports multi-use
* Connection state changed (HTTP/2 confirmed)
```

Trivia Time

Winner Gets Azat's Full Stack JavaScript


THE EXPERT'S VOICE®

Full Stack JavaScript

Learn Backbone.js, Node.js and MongoDB

—

Azat Mardan

Apress®

❓

Server Push—Yeah!

```
const http2 = require('spdy')
const logger = require('morgan')
const express = require('express')
const app = express()
const fs = require('fs')
```

```
app.use(logger('dev'))
app.get('/', (req, res) => {
  res.send(`hello, http2!
go to /pushy`)
})
```

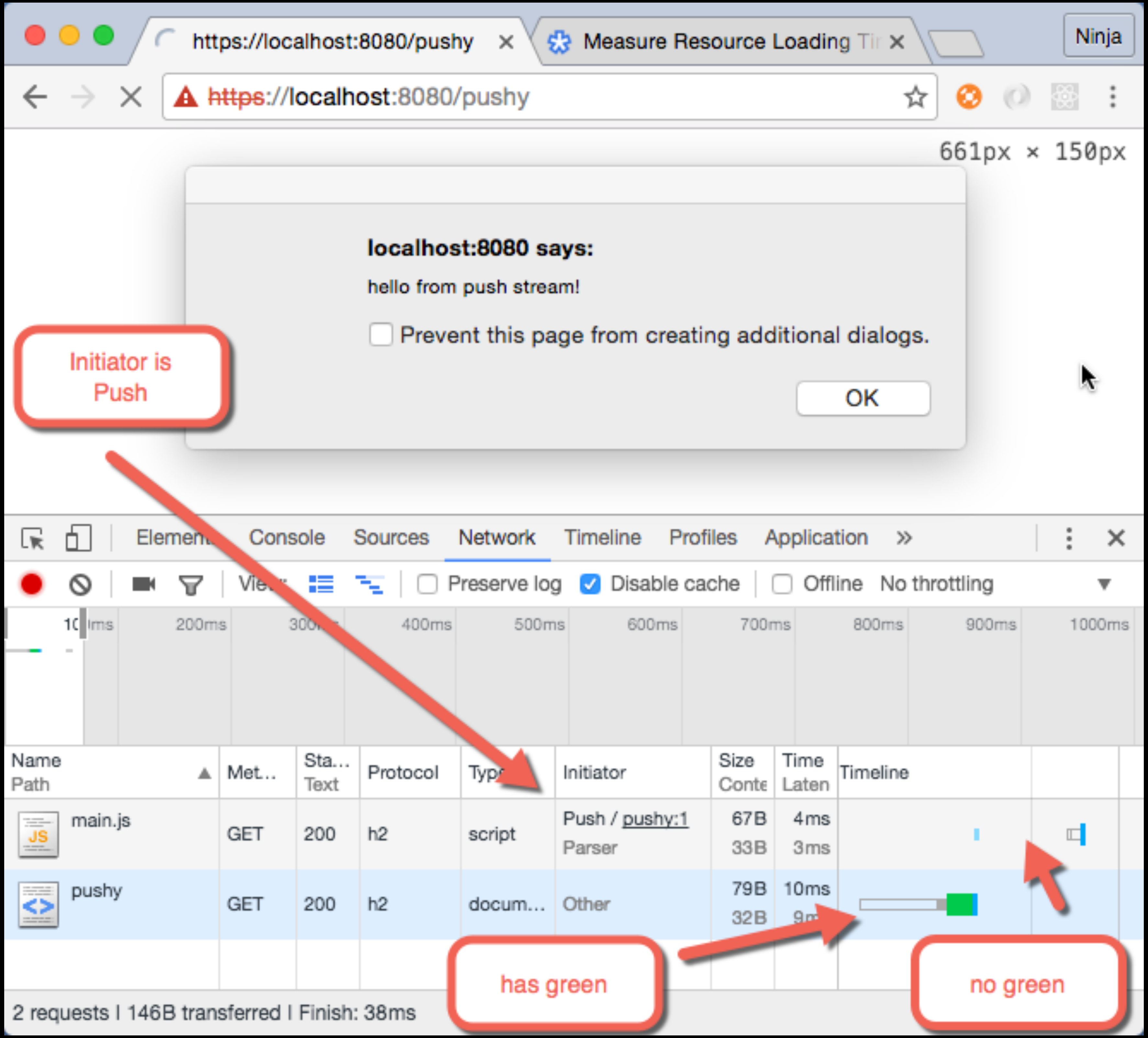
```
app.get('/pushy', (req, res) => {
  var stream = res.push('/main.js', {
    status: 200, // optional
    method: 'GET', // optional
    request: {
      accept: '*/*'
    },
    response: {
      'content-type': 'application/javascript'
    }
  })
  stream.on('error', () => {
  })
  stream.end('alert("hello from push stream!");')
  res.end('<script src="/main.js"></script>')
})
```

```
var options = {
  key: fs.readFileSync('./server.key'),
  cert: fs.readFileSync('./server.crt')
}

http2
  .createServer(options, app)
  .listen(8080, ()=>{
    console.log(`Server is listening on https://localhost:8080.
You can open the URL in the browser.`)
  }
)
```

Result

```
GET /pushy 200 4.918 ms - -
```


https://localhost:8080/pushy
Measure Resource Loading Ti
Ninja
https://localhost:8080/pushy
661px × 150px

localhost:8080 says:

hello from push stream!

☐ Prevent this page from creating additional dialogs.

Name / Path	Met…	Sta… / Text	Protocol	Type	Initiator	Size / Conte	Time / Laten	Timeline
main.js	GET	200	h2	script	Push / pushy:1 Parser	67B 33B	4ms 3ms	
pushy	GET	200	h2	docum…	Other	79B 32B	10ms 9ms	

2 requests | 146B transferred | Finish: 38ms

Code

- https://github.com/azat-co/http2-express
- https://github.com/azat-co/http2-node-server-push

Want to work with Node but your boss won't let you?

Capital One is hiring ~2,000 more software engineers in UK, Canada and US.

https://jobs.capitalone.com

We use Node and other cutting-edge open source tech a lot! (React, Kotlin, Clojure, Angular 2, TypeScript, Go, etc.)

Learn More

Node at Capital One by Azat Mardan at Node Interactive 2015

https://www.youtube.com/watch?v=BJPeLJhv1Ic

Tip

Email me to be referred for a job at Capital One.

My Contacts

Twitter: @azat_co
Email: hi@azat.co

30-Second Summary

1. spdy or http2 core (soon)
2. Express rocks
3. Server Push - yeah!

Slides 👓 📄 🖥️

Everything (PDF+Markdown): https://github.com/azat-co/h2-node

Want to learn more about Node.js?

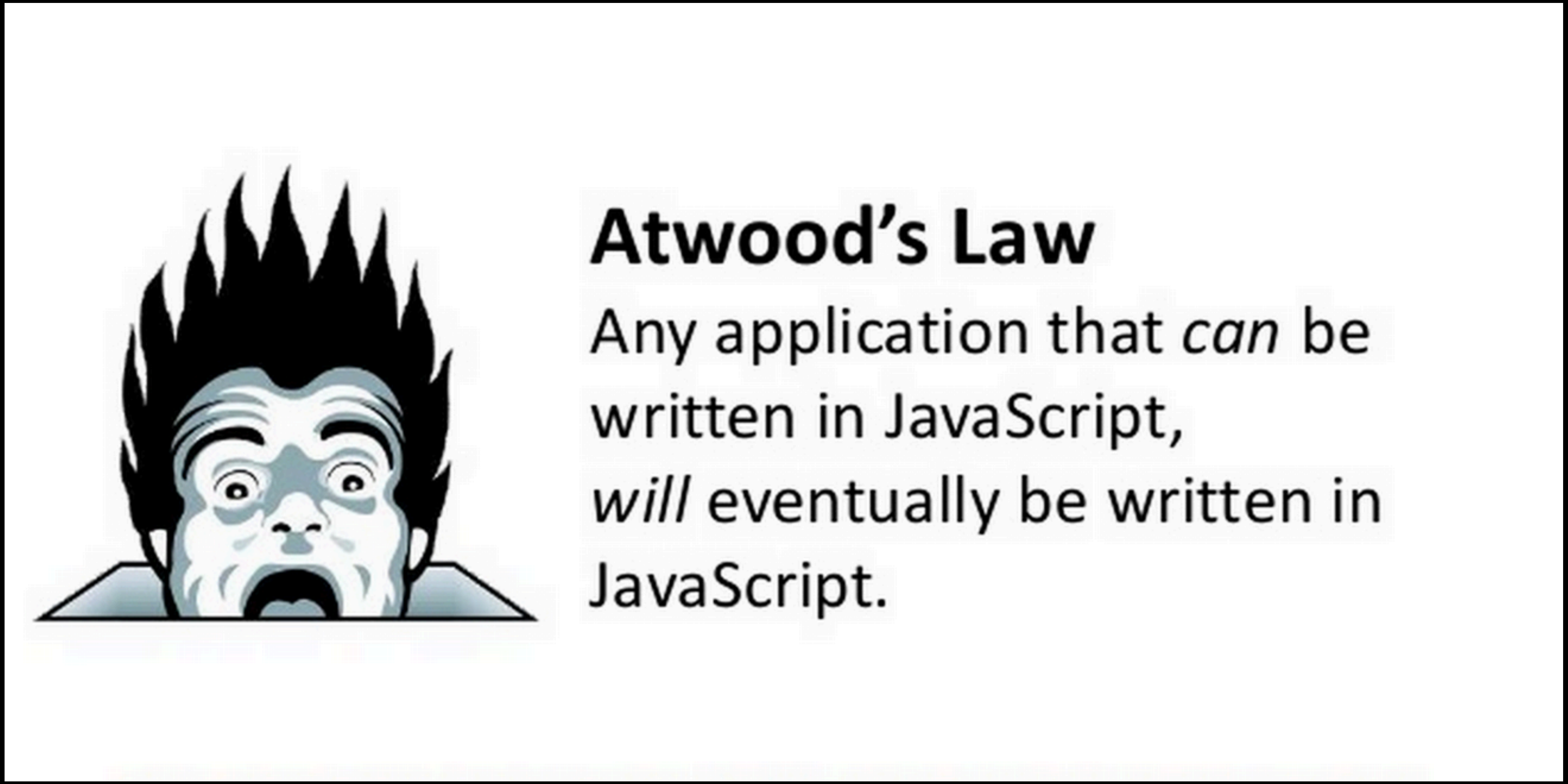
Check out <u>Node.University</u>, <u>Webapplog.com</u> and <u>NodeProgram.com</u> for the best online and in-person education!

Node University

http://node.university

One Last Thing 👉

CodingHorror.com



Atwood's Law

Any application that *can* be written in JavaScript, *will* eventually be written in JavaScript.