HTTP/2 FOR ME & YOU

HTTP/2 FOR:

- SERVERS
- INVESTIGATORS
- YOUR APPS

TARA Z. MANICSIC

DEV ADVOCATE - PROGRESS KINVEY
GOOGLE DEVELOPER EXPERT
WOMEN WHO CODE CINCY
CINCY NODESCHOOL
@TZMANICS

HTTPS://WWW.CROWDCAST.IO/E/DSHAWAF8

SHOOOOOO

-MULTIPLEXING

SHOOOOOO

- MULTIPLEXING
- BINARY FRAMING LAYER

OOOOOOHS

- -MULTIPLEXING
- -BINARY FRAMING LAYER
- -HEADER COMPRESSION

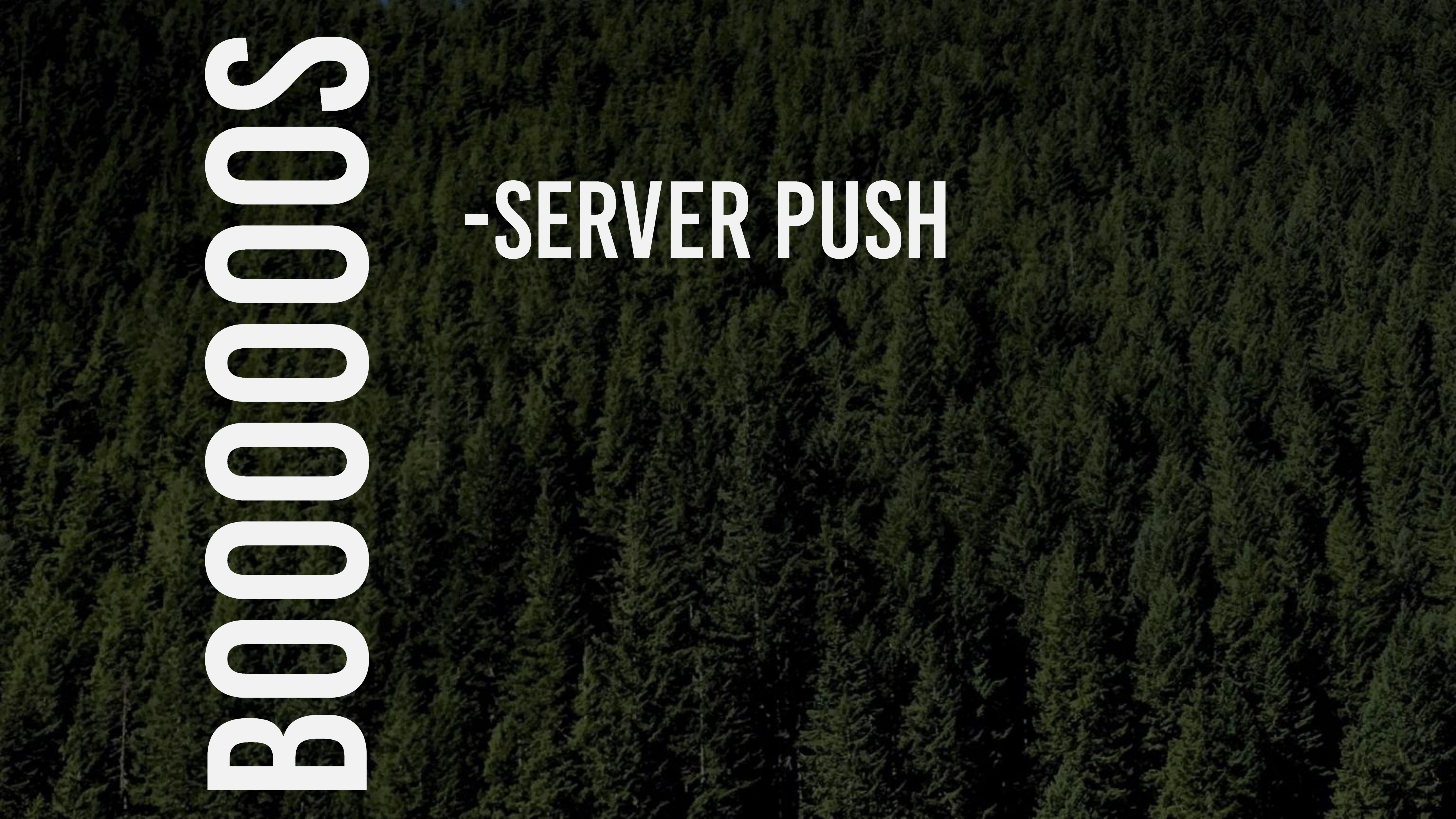
OOOOOOHS

-MULTIPLEXING

-BINARY FRAMING LAYER

-HEADER COMPRESSION

-STREAM PRIORITIZATION



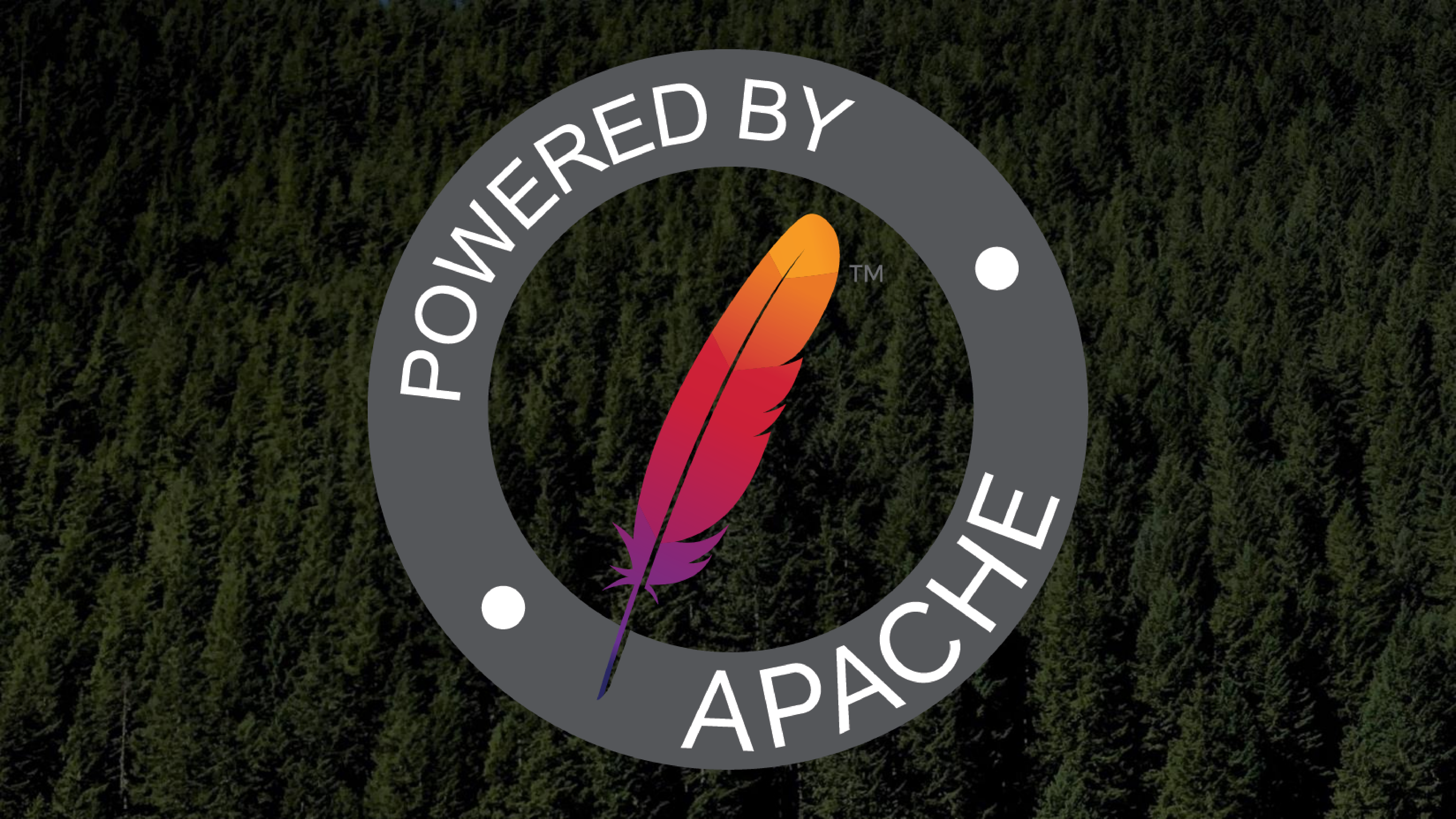
BOOOOOS

-SERVER PUSH

BOOOOOOS

- SERVER PUSH
- HTTP/1 COMPATIBILITY
- COMPLEXITY
- LACK OF GUIDANCE

HOW YOUR SERVERS ARE GOING TO HANDLE HTTP/2



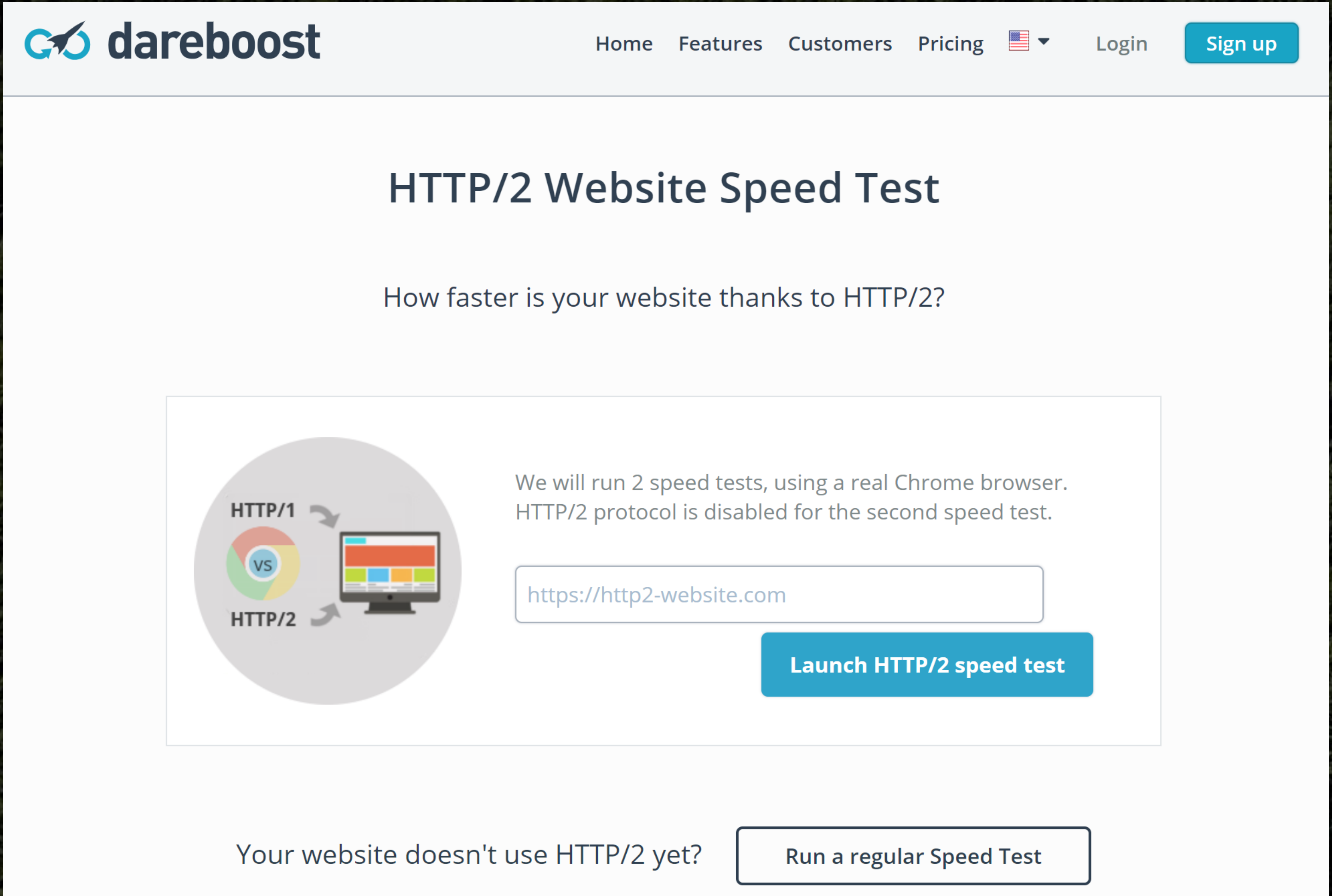
POWERED BY APACHE™

```
LoadModule http2_module modules/mod_http2.so
Protocols h2 http/1.1
Protocols h2 h2c http/1.1
Link </xxx.css>;rel=preload, </xxx.js>; rel=preload


     <Location /xxx.html>
        Header add Link "</xxx.css>;rel=preload"
        Header add Link "</xxx.js>;rel=preload"
     </Location>
```

NGINX

HOW WE CAN VISUALIZE HTTP/2 WORKING

HTTPS://WWW.DAREBOOST.COM/EN/WEBSITE-SPEED-TEST-HTTP2-VS-HTTP1



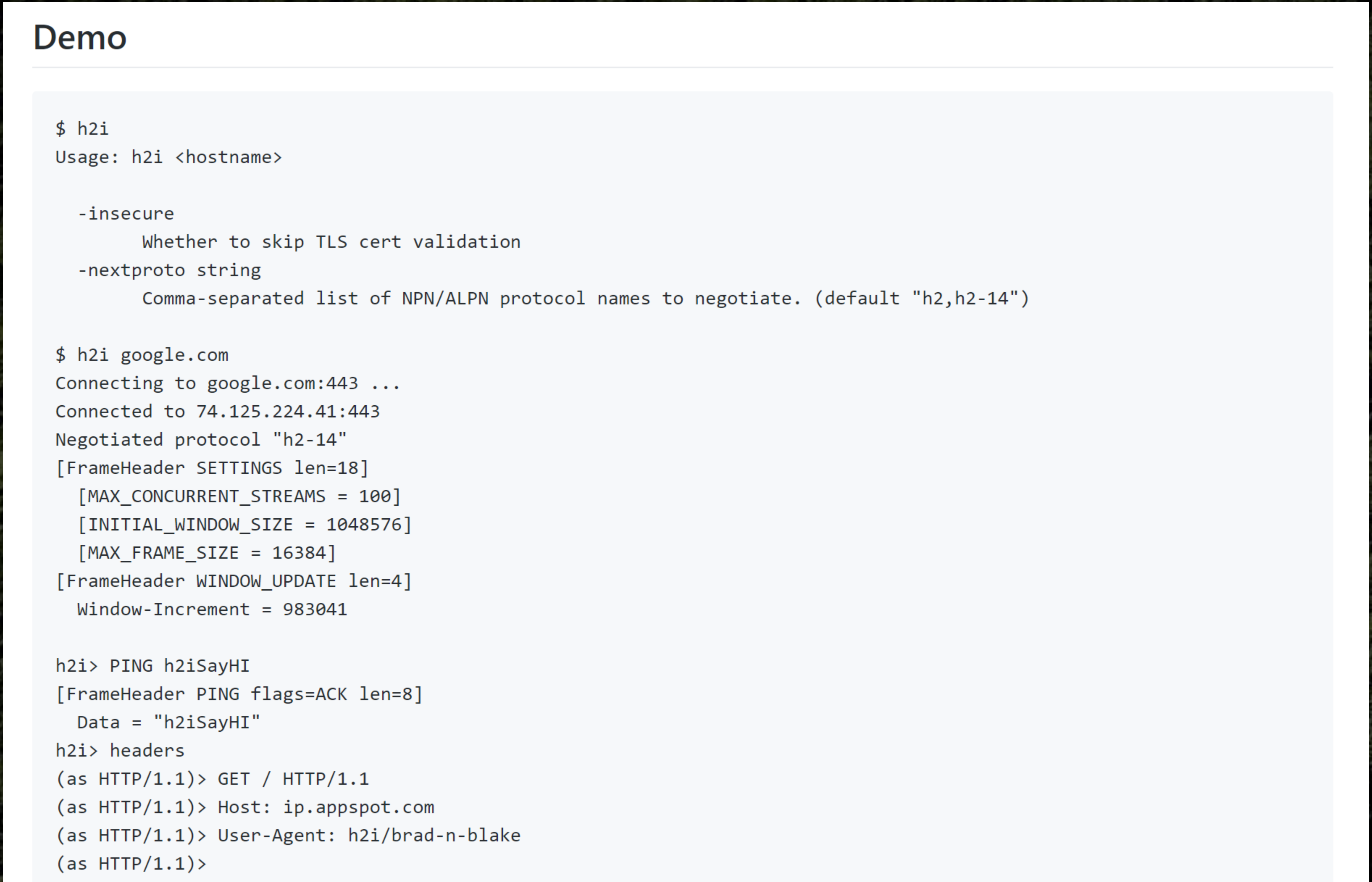
HTTP/2 Website Speed Test

How faster is your website thanks to HTTP/2?

We will run 2 speed tests, using a real Chrome browser. HTTP/2 protocol is disabled for the second speed test.

Your website doesn't use HTTP/2 yet?

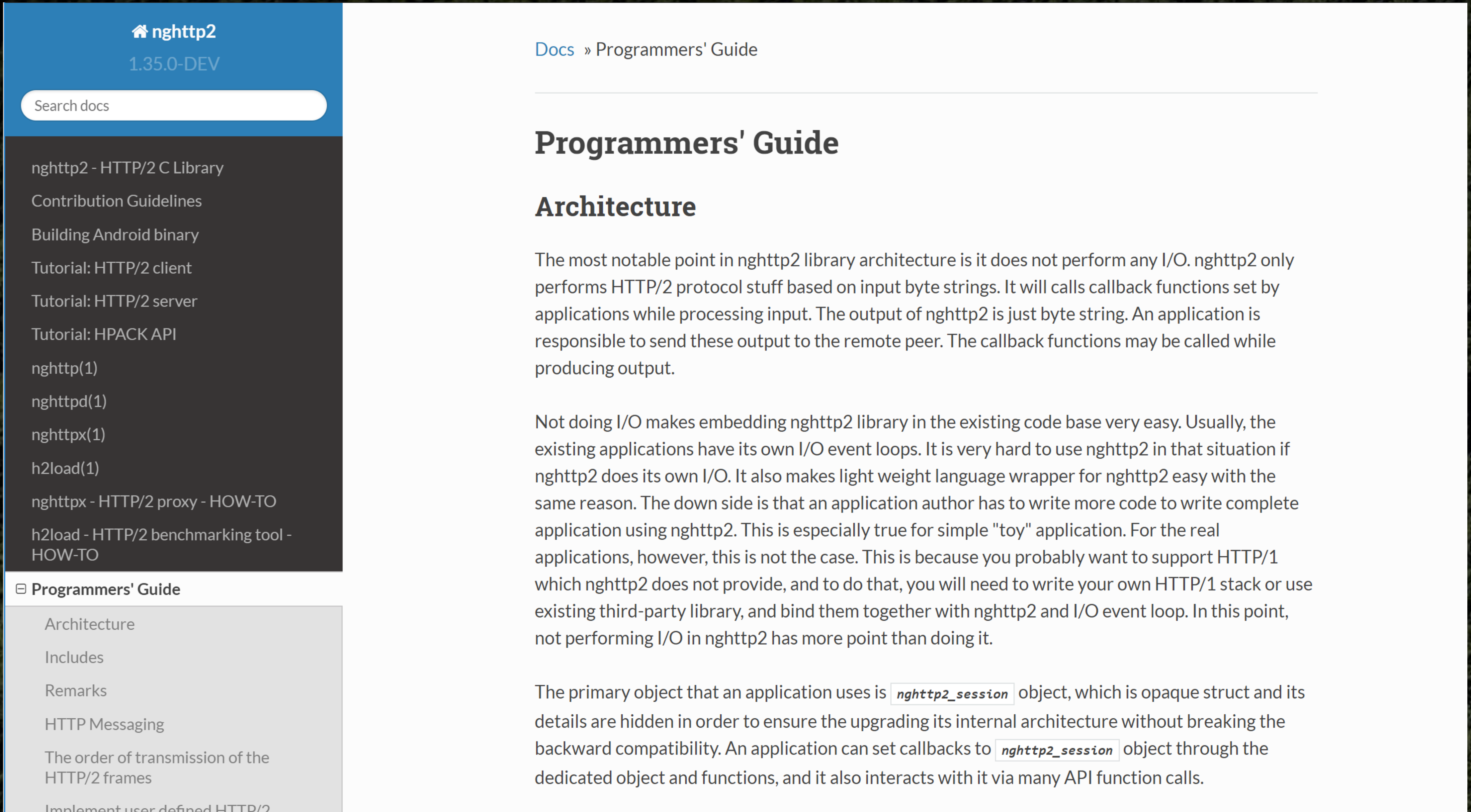
HTTPS://GITHUB.COM/GOLANG/NET/TREE/MASTER/HTTP2/H2I



Demo

```
$ h2i
Usage: h2i <hostname>

  -insecure
        Whether to skip TLS cert validation
  -nextproto string
        Comma-separated list of NPN/ALPN protocol names to negotiate. (default "h2,h2-14")

$ h2i google.com
Connecting to google.com:443 ...
Connected to 74.125.224.41:443
Negotiated protocol "h2-14"
[FrameHeader SETTINGS len=18]
  [MAX_CONCURRENT_STREAMS = 100]
  [INITIAL_WINDOW_SIZE = 1048576]
  [MAX_FRAME_SIZE = 16384]
[FrameHeader WINDOW_UPDATE len=4]
  Window-Increment = 983041

h2i> PING h2iSayHI
[FrameHeader PING flags=ACK len=8]
  Data = "h2iSayHI"
h2i> headers
(as HTTP/1.1)> GET / HTTP/1.1
(as HTTP/1.1)> Host: ip.appspot.com
(as HTTP/1.1)> User-Agent: h2i/brad-n-blake
(as HTTP/1.1)>
```

HTTPS://NGHTTP2.ORG/DOCUMENTATION/



nghttp2
1.35.0-DEV

Search docs

Docs » Programmers' Guide

Programmers' Guide

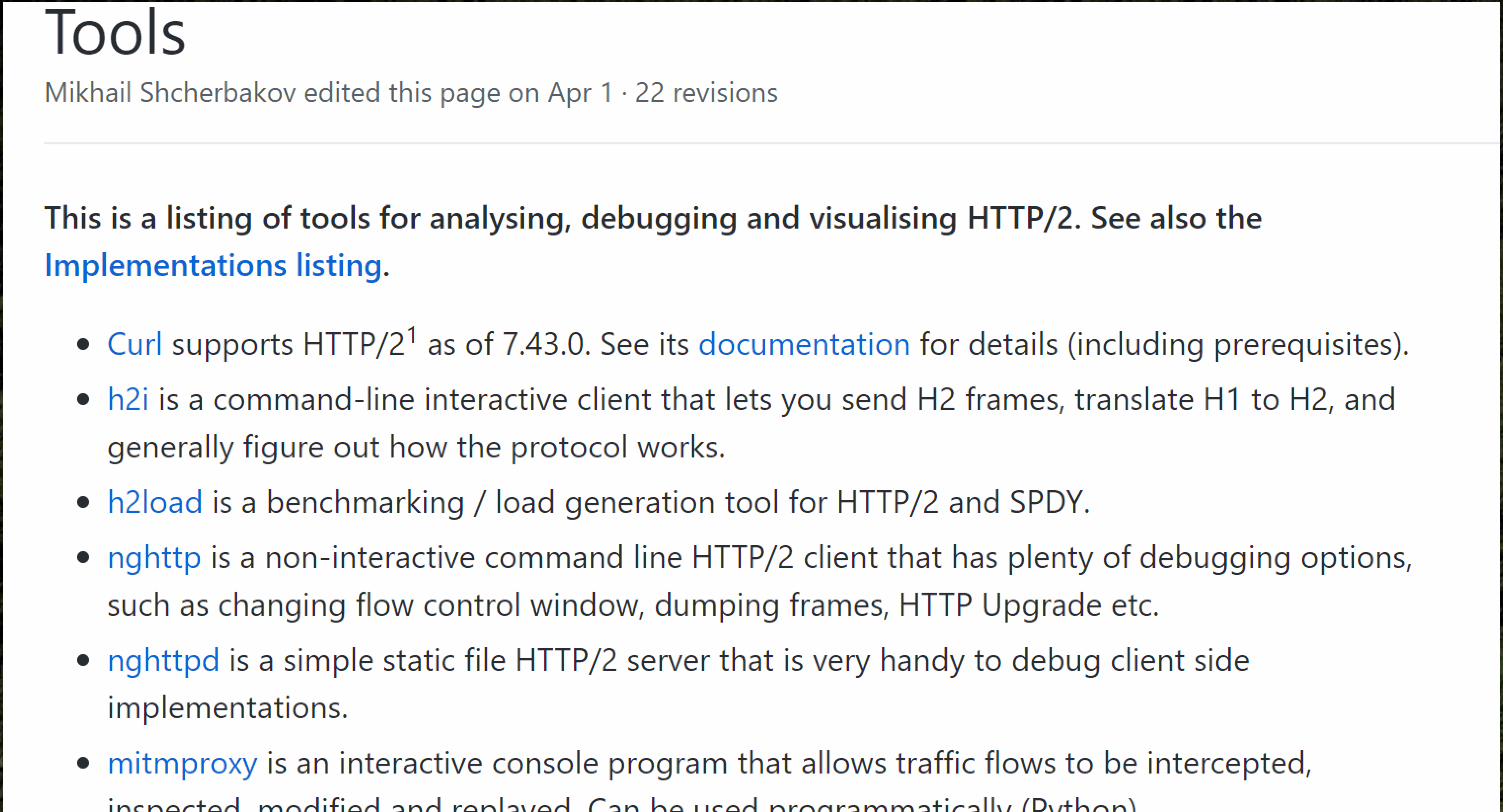
Architecture

The most notable point in nghttp2 library architecture is it does not perform any I/O. nghttp2 only performs HTTP/2 protocol stuff based on input byte strings. It will calls callback functions set by applications while processing input. The output of nghttp2 is just byte string. An application is responsible to send these output to the remote peer. The callback functions may be called while producing output.

Not doing I/O makes embedding nghttp2 library in the existing code base very easy. Usually, the existing applications have its own I/O event loops. It is very hard to use nghttp2 in that situation if nghttp2 does its own I/O. It also makes light weight language wrapper for nghttp2 easy with the same reason. The down side is that an application author has to write more code to write complete application using nghttp2. This is especially true for simple "toy" application. For the real applications, however, this is not the case. This is because you probably want to support HTTP/1 which nghttp2 does not provide, and to do that, you will need to write your own HTTP/1 stack or use existing third-party library, and bind them together with nghttp2 and I/O event loop. In this point, not performing I/O in nghttp2 has more point than doing it.

The primary object that an application uses is `nghttp2_session` object, which is opaque struct and its details are hidden in order to ensure the upgrading its internal architecture without breaking the backward compatibility. An application can set callbacks to `nghttp2_session` object through the dedicated object and functions, and it also interacts with it via many API function calls.

HTTPS://GITHUB.COM/HTTP2/HTTP2-SPEC/WIKI/TOOLS



Tools

Mikhail Shcherbakov edited this page on Apr 1 · 22 revisions

This is a listing of tools for analysing, debugging and visualising HTTP/2. See also the Implementations listing.

- Curl supports HTTP/2[1] as of 7.43.0. See its documentation for details (including prerequisites).
- h2i is a command-line interactive client that lets you send H2 frames, translate H1 to H2, and generally figure out how the protocol works.
- h2load is a benchmarking / load generation tool for HTTP/2 and SPDY.
- nghttp is a non-interactive command line HTTP/2 client that has plenty of debugging options, such as changing flow control window, dumping frames, HTTP Upgrade etc.
- nghttpd is a simple static file HTTP/2 server that is very handy to debug client side implementations.
- mitmproxy is an interactive console program that allows traffic flows to be intercepted, inspected, modified and replayed. Can be used programmatically (Python)

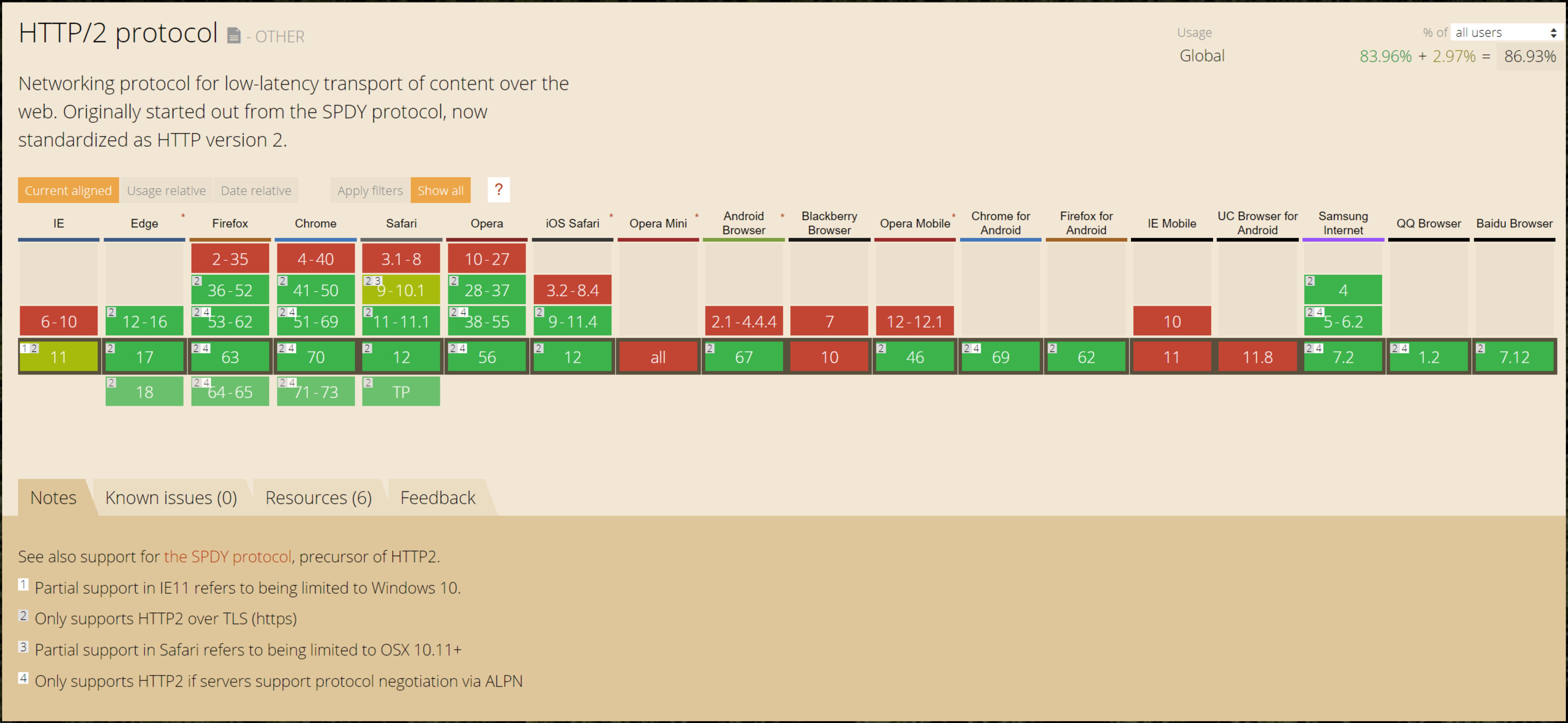
<JOKE-BREAK>

WHAT DO YOU CALL A FRENCHMAN WEARING SANDALS?

PHILIPPE PHILOPPE

</JOKE-BREAK>

HOW MY APP WILL WORK WITH HTTP/2



HTTP/2 protocol - OTHER

Usage: % of all users

Global: 83.96% + 2.97% = 86.93%

Networking protocol for low-latency transport of content over the web. Originally started out from the SPDY protocol, now standardized as HTTP version 2.

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser
		🟥 2-35	🟥 4-40	🟥 3.1-8	🟥 10-27												
		🟩 [2] 36-52	🟩 [2] 41-50	🟨 [2 3] 9-10.1	🟩 [2] 28-37	🟥 3.2-8.4									🟩 [2] 4		
🟥 6-10	🟩 [2] 12-16	🟩 [2 4] 53-62	🟩 [2 4] 51-69	🟩 [2] 11-11.1	🟩 [2 4] 38-55	🟩 [2] 9-11.4		🟥 2.1-4.4.4	🟥 7	🟥 12-12.1			🟥 10		🟩 [2 4] 5-6.2		
🟨 [1 2] 11	🟩 [2] 17	🟩 [2 4] 63	🟩 [2 4] 70	🟩 [2] 12	🟩 [2 4] 56	🟩 [2] 12	🟥 all	🟩 [2] 67	🟥 10	🟩 [2] 46	🟩 [2 4] 69	🟩 [2] 62	🟥 11	🟥 11.8	🟩 [2 4] 7.2	🟩 [2 4] 1.2	🟩 [2] 7.12
	🟩 [2] 18	🟩 [2 4] 64-65	🟩 [2 4] 71-73	🟩 [2] TP													

Notes | Known issues (0) | Resources (6) | Feedback

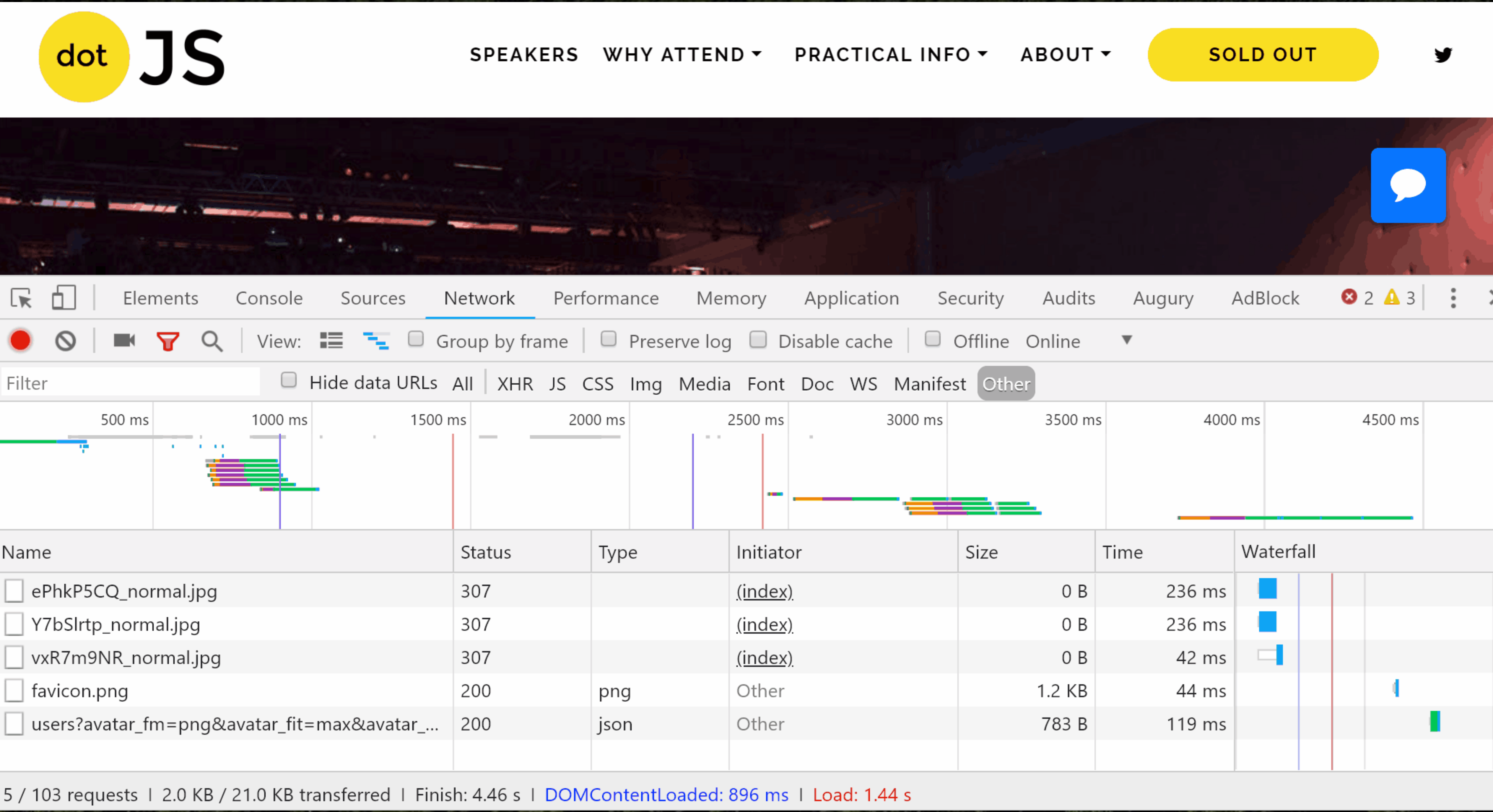
See also support for the SPDY protocol, precursor of HTTP2.

[1] Partial support in IE11 refers to being limited to Windows 10.

[2] Only supports HTTP2 over TLS (https)

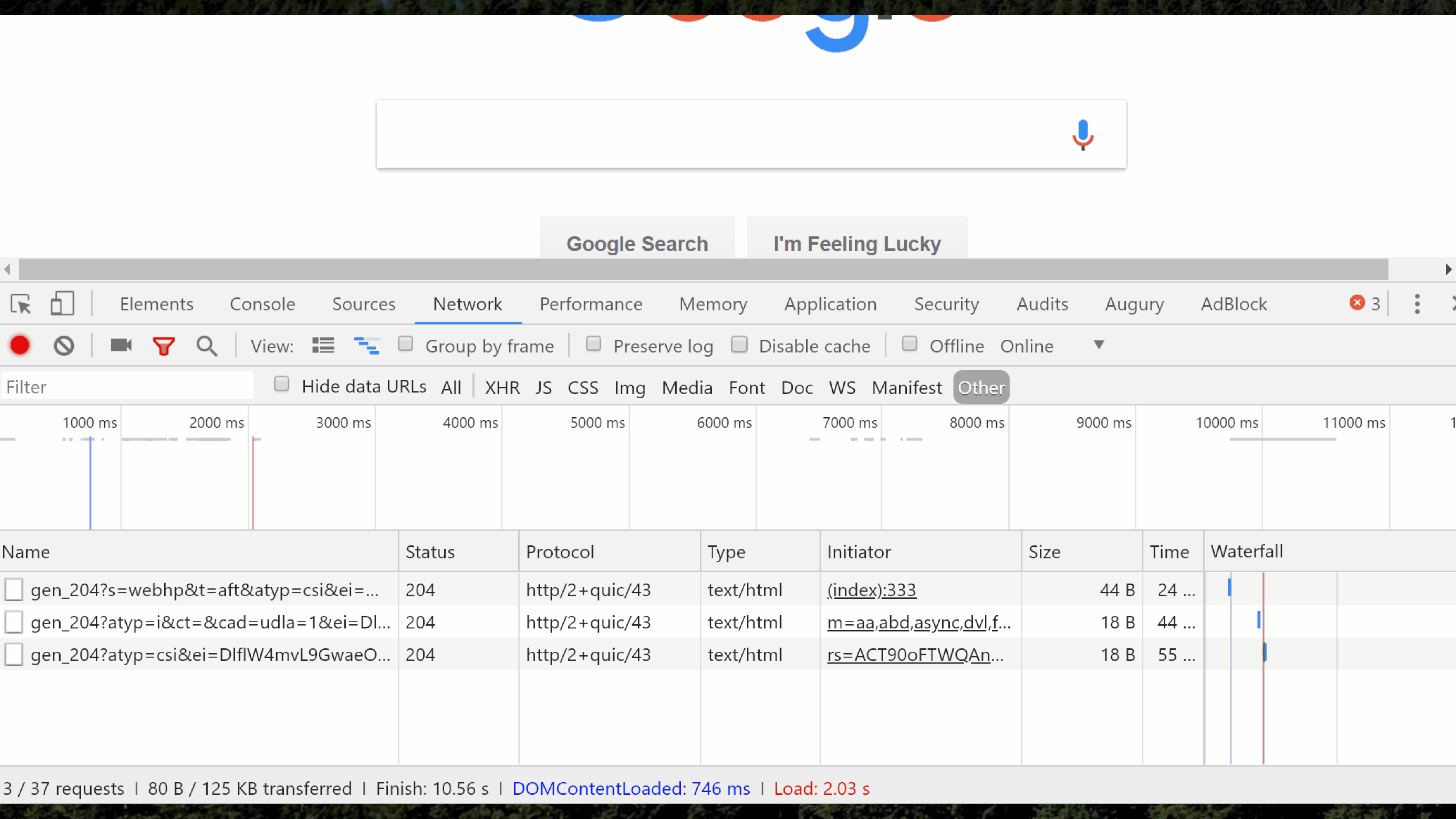
[3] Partial support in Safari refers to being limited to OSX 10.11+

[4] Only supports HTTP2 if servers support protocol negotiation via ALPN



Name	Status	Type	Initiator	Size	Time	Waterfall
ePhkP5CQ_normal.jpg	307		(index)	0 B	236 ms	
Y7bSlrtp_normal.jpg	307		(index)	0 B	236 ms	
vxR7m9NR_normal.jpg	307		(index)	0 B	42 ms	
favicon.png	200	png	Other	1.2 KB	44 ms	
users?avatar_fm=png&avatar_fit=max&avatar_...	200	json	Other	783 B	119 ms	

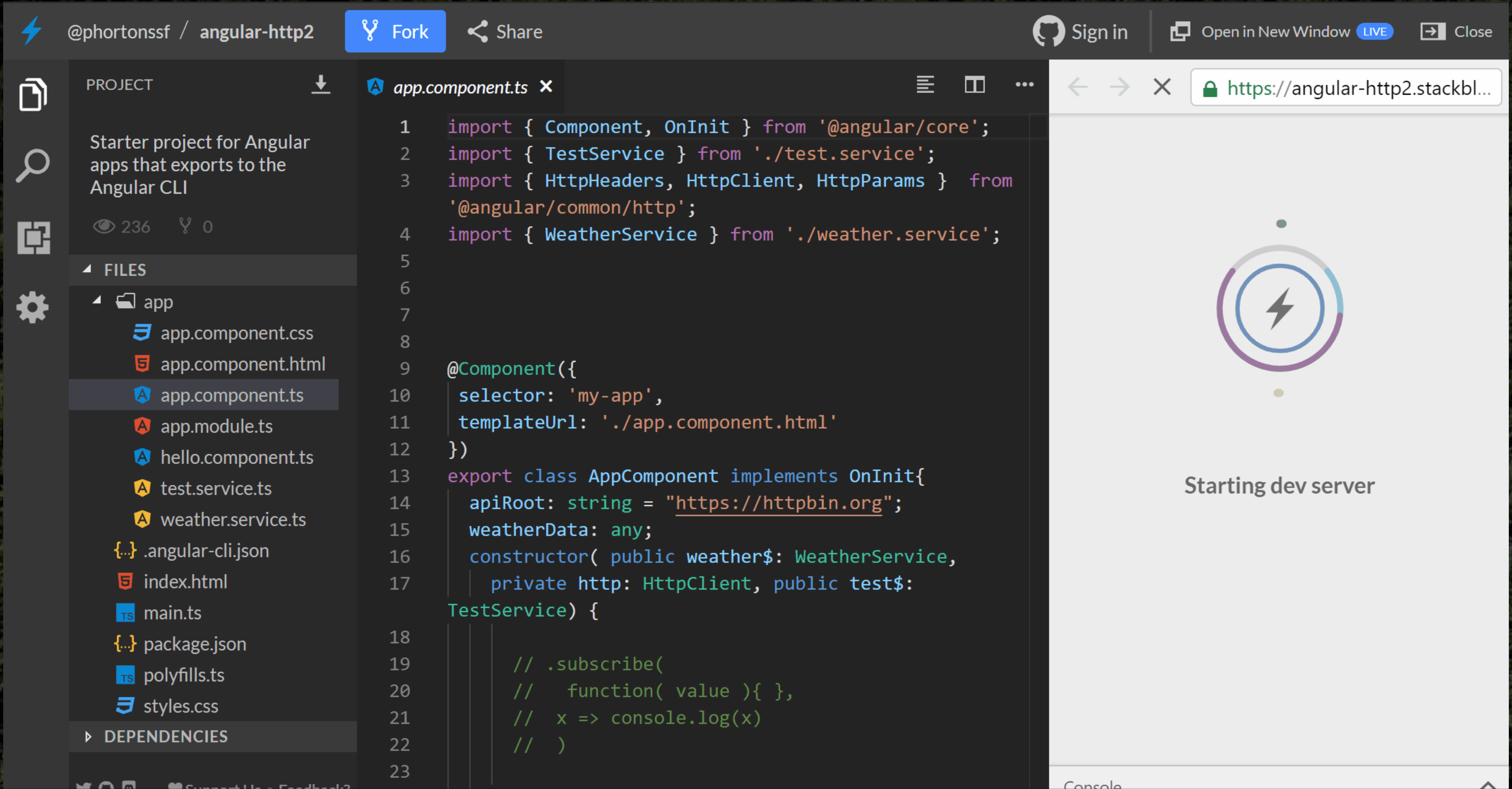
5 / 103 requests | 2.0 KB / 21.0 KB transferred | Finish: 4.46 s | DOMContentLoaded: 896 ms | Load: 1.44 s



Name	Status	Protocol	Type	Initiator	Size	Time	Waterfall
gen_204?s=webhp&t=aft&atyp=csi&ei=...	204	http/2+quic/43	text/html	(index):333	44 B	24 ...	
gen_204?atyp=i&ct=&cad=udla=1&ei=Dl...	204	http/2+quic/43	text/html	m=aa,abd,async,dvl,f...	18 B	44 ...	
gen_204?atyp=csi&ei=DlflW4mvL9GwaeO...	204	http/2+quic/43	text/html	rs=ACT90oFTWQAn...	18 B	55 ...	

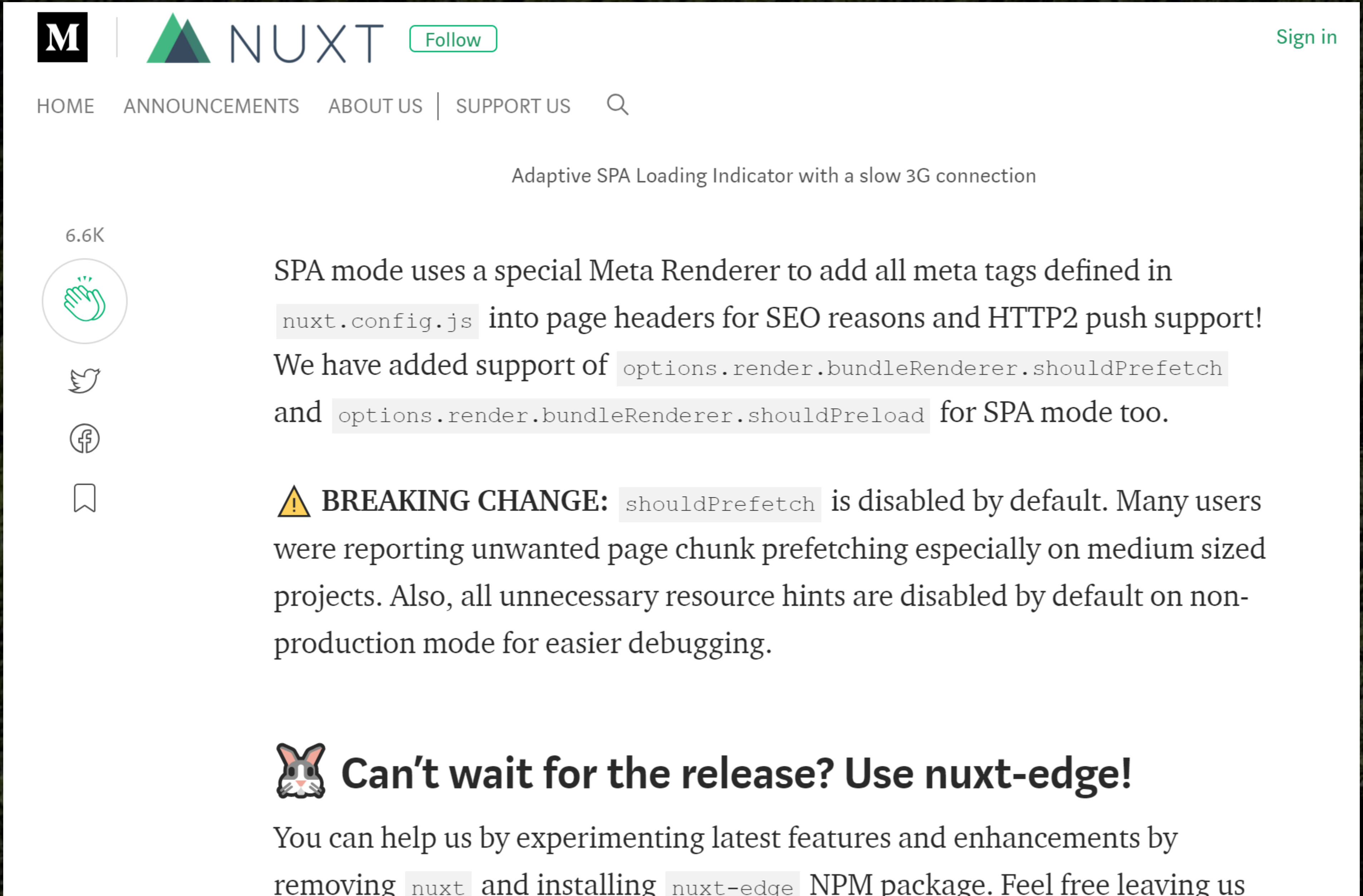
3 / 37 requests | 80 B / 125 KB transferred | Finish: 10.56 s | DOMContentLoaded: 746 ms | Load: 2.03 s

HTTPS://STACKBLITZ.COM/EDIT/ANGULAR-HTTP2



```
import { Component, OnInit } from '@angular/core';
import { TestService } from './test.service';
import { HttpHeaders, HttpClient, HttpParams }  from
'@angular/common/http';
import { WeatherService } from './weather.service';




@Component({
 selector: 'my-app',
 templateUrl: './app.component.html'
})
export class AppComponent implements OnInit{
  apiRoot: string = "https://httpbin.org";
  weatherData: any;
  constructor( public weather$: WeatherService,
    private http: HttpClient, public test$:
TestService) {

      // .subscribe(
      //   function( value ){ },
      //  x => console.log(x)
      //  )

```

HTTPS://MEDIUM.COM/NUXT/NUXT-2-IS-COMING-OH-YEAH-212C1A9E1A67



Adaptive SPA Loading Indicator with a slow 3G connection

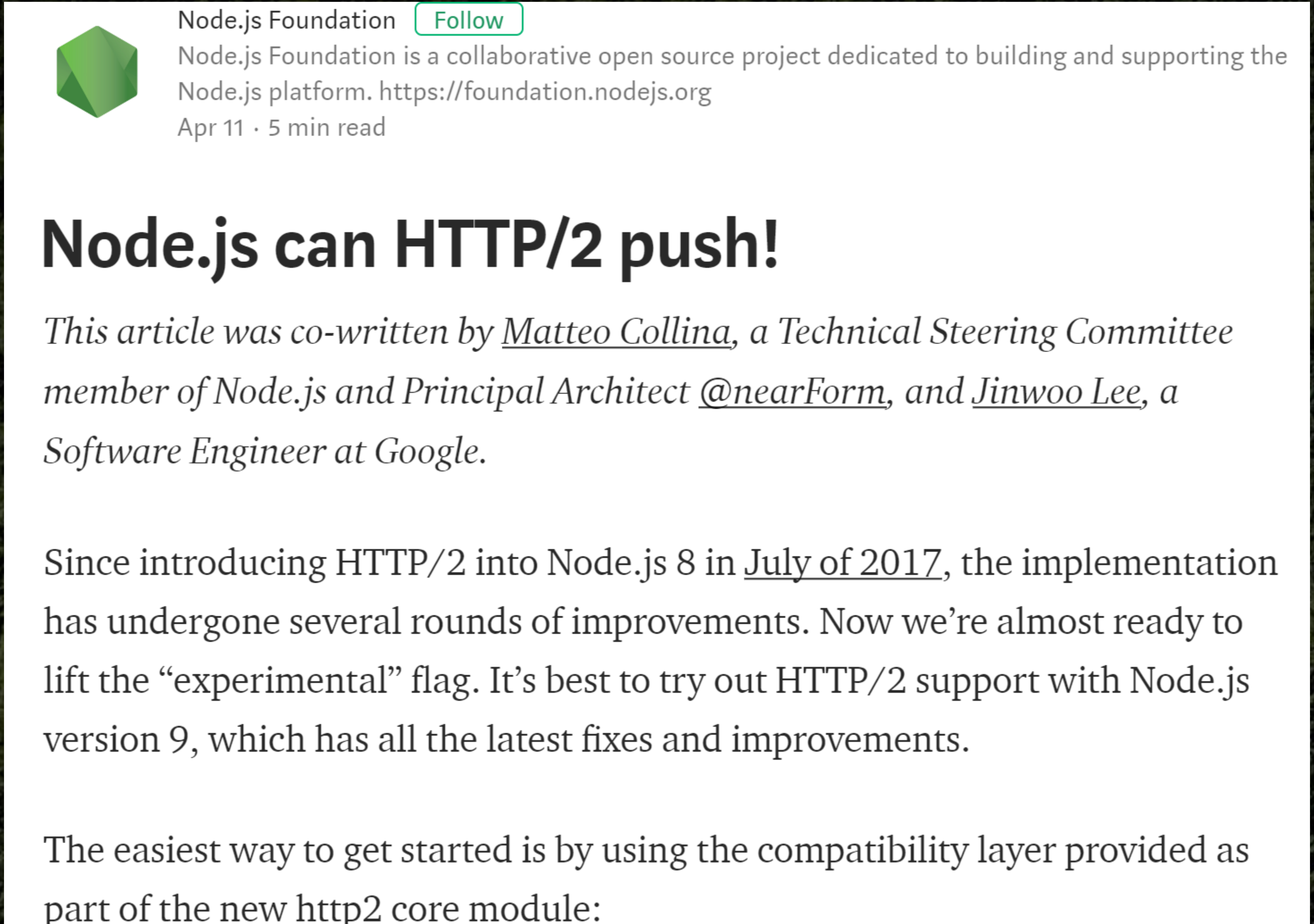
SPA mode uses a special Meta Renderer to add all meta tags defined in `nuxt.config.js` into page headers for SEO reasons and HTTP2 push support! We have added support of `options.render.bundleRenderer.shouldPrefetch` and `options.render.bundleRenderer.shouldPreload` for SPA mode too.

⚠️ **BREAKING CHANGE:** `shouldPrefetch` is disabled by default. Many users were reporting unwanted page chunk prefetching especially on medium sized projects. Also, all unnecessary resource hints are disabled by default on non-production mode for easier debugging.

🐰 Can't wait for the release? Use nuxt-edge!

You can help us by experimenting latest features and enhancements by removing `nuxt` and installing `nuxt-edge` NPM package. Feel free leaving us

HTTPS://MEDIUM.COM/THE-NODE-JS-COLLECTION/NODE-JS-CAN-HTTP-2-PUSH-B491894E1BB1



Node.js Foundation Follow
Node.js Foundation is a collaborative open source project dedicated to building and supporting the Node.js platform. https://foundation.nodejs.org
Apr 11 · 5 min read

Node.js can HTTP/2 push!

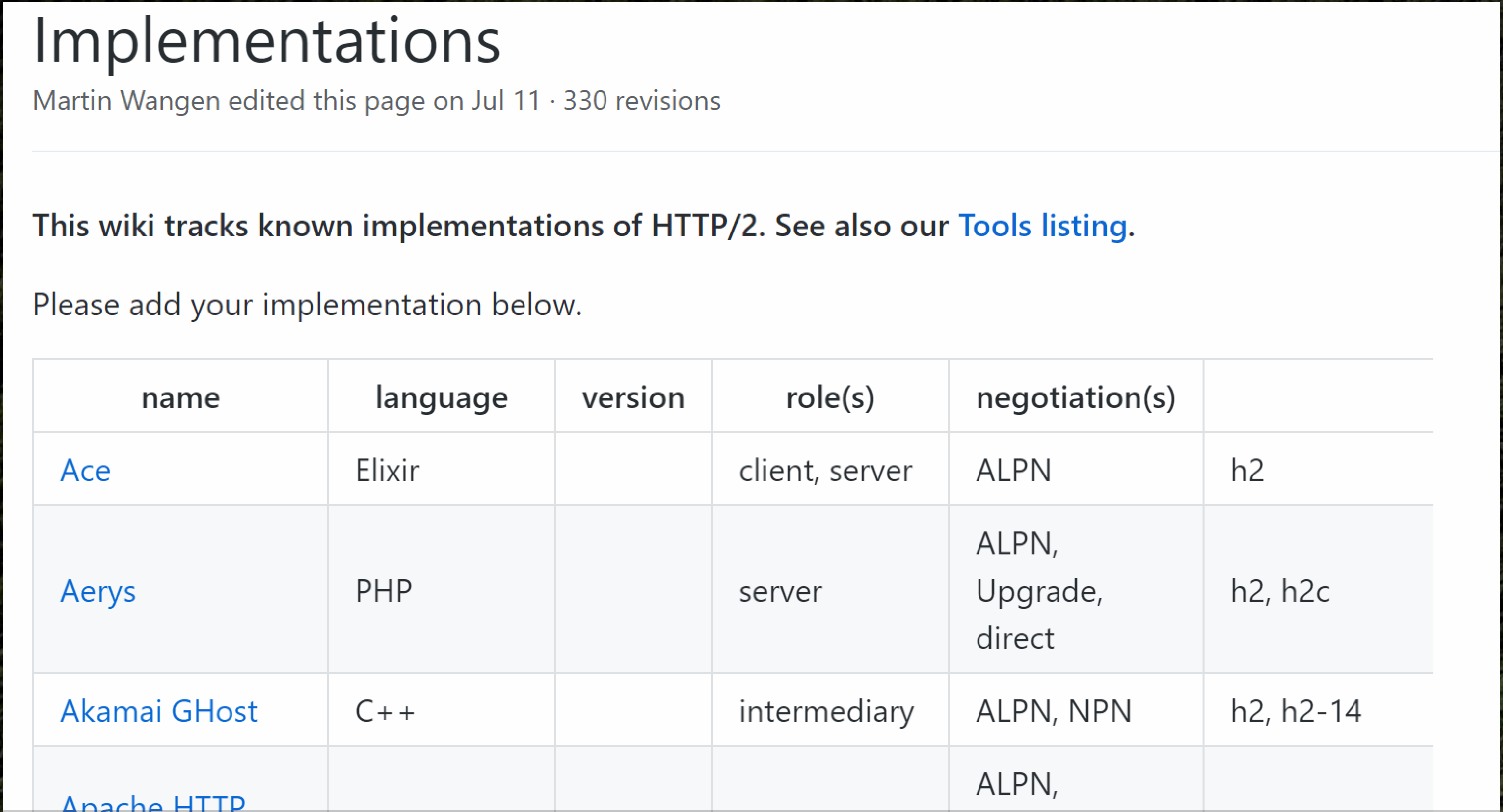
This article was co-written by Matteo Collina, a Technical Steering Committee member of Node.js and Principal Architect @nearForm, and Jinwoo Lee, a Software Engineer at Google.

Since introducing HTTP/2 into Node.js 8 in July of 2017, the implementation has undergone several rounds of improvements. Now we're almost ready to lift the "experimental" flag. It's best to try out HTTP/2 support with Node.js version 9, which has all the latest fixes and improvements.

The easiest way to get started is by using the compatibility layer provided as part of the new http2 core module:

```javascript
async function main() {
  const {key, cert} = await createServerOptions();
  // Browsers support only https for HTTP/2.
  const app = fastify({https: {key, cert}, http2: true});

  // Create and register AutoPush plugin. It should be registered as the first
  // in the middleware chain.
  app.register(fastifyAutoPush.staticServe, {root: STATIC_DIR});

  await app.listen(PORT);
  console.log(`Listening on port ${PORT}`);
}
```

HTTPS://GITHUB.COM/HTTP2/HTTP2-SPEC/WIKI/IMPLEMENTATIONS



Implementations

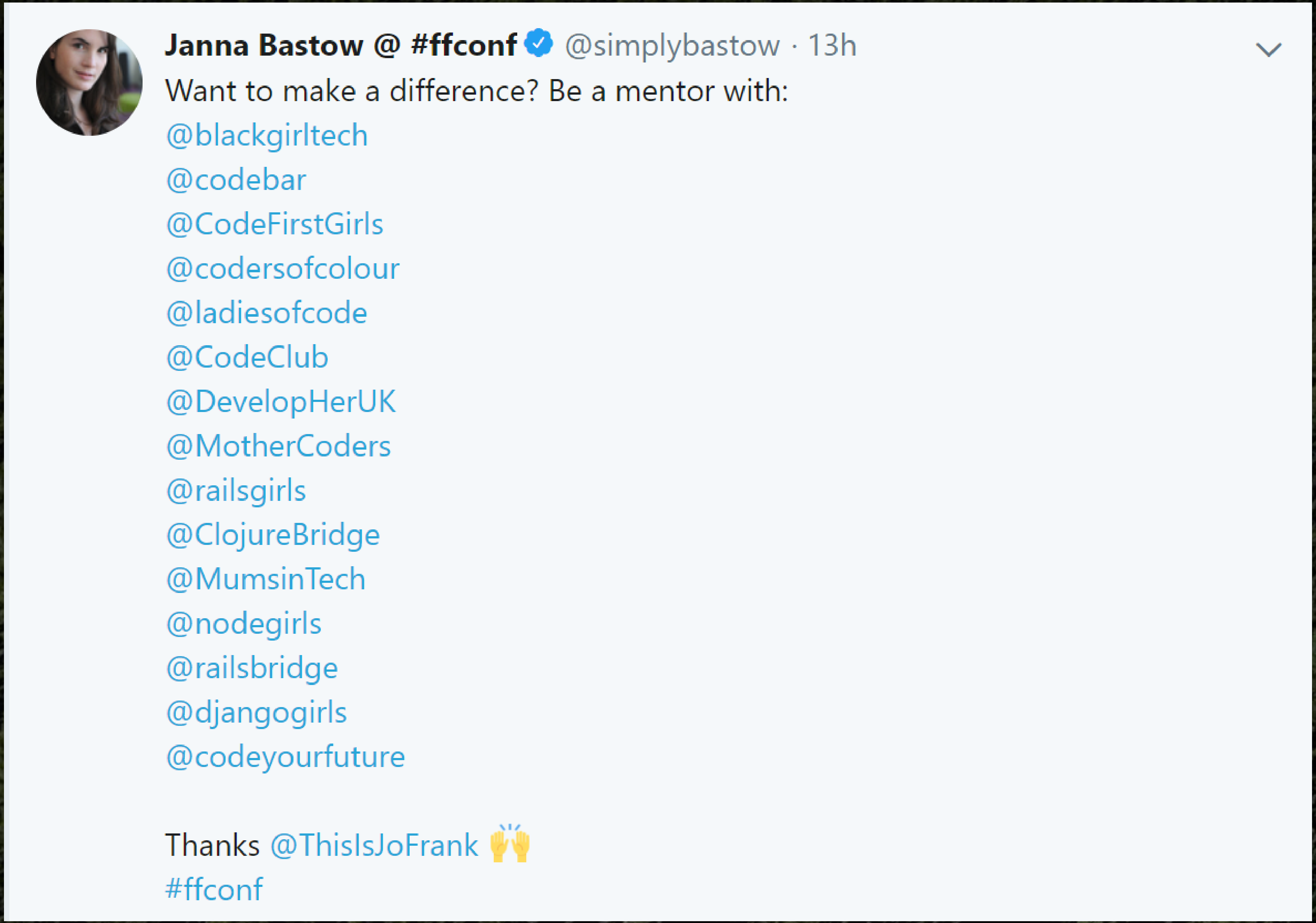
Martin Wangen edited this page on Jul 11 · 330 revisions

This wiki tracks known implementations of HTTP/2. See also our Tools listing.

Please add your implementation below.

name	language	version	role(s)	negotiation(s)	
Ace	Elixir		client, server	ALPN	h2
Aerys	PHP		server	ALPN, Upgrade, direct	h2, h2c
Akamai GHost	C++		intermediary	ALPN, NPN	h2, h2-14
Apache HTTP				ALPN,	

ONE MORE THING...


Janna Bastow @ #ffconf @simplybastow · 13h

Want to make a difference? Be a mentor with:
@blackgirltech
@codebar
@CodeFirstGirls
@codersofcolour
@ladiesofcode
@CodeClub
@DevelopHerUK
@MotherCoders
@railsgirls
@ClojureBridge
@MumsinTech
@nodegirls
@railsbridge
@djangogirls
@codeyourfuture

Thanks @ThisIsJoFrank 🙌
#ffconf

MENTORS WANTED

@ANGULARGIRLS
@BLACKGIRLTECH
@CODEBAR
@CODECLUB
@CODEFIRSTGIRLS
@CODEYOURFUTURE
@CODERSOFCOLOUR
@CLOJUREBRIDGE
@DEVELOPHERUK
@DJANGOGIRLS
@GIRLSCRIPT1
@GOLANGBRIDGE
@LADIESOFCODE
@MOTHERCODERS
@MUMSINTECH
@NODEGIRLS
@RAILSBRIDGE
@RAILSGIRLS
@TECHNOVATION
@VUEVIXENS
@WOMENWHOGO

THANK YOU, DOTJS!