Web Performance in 2017

with @bighappyface

Big thanks to DrupalCon Team

Big thanks to you 👏
(it’s almost a wrap)

Please note

- This session assumes familiarity. I am speaking as if folks are already into this stuff.
- This session is marked as advanced. My apologies for casual techno-speak.
- This session is not exhaustive. I cover things that I think are interesting and important.

The plan

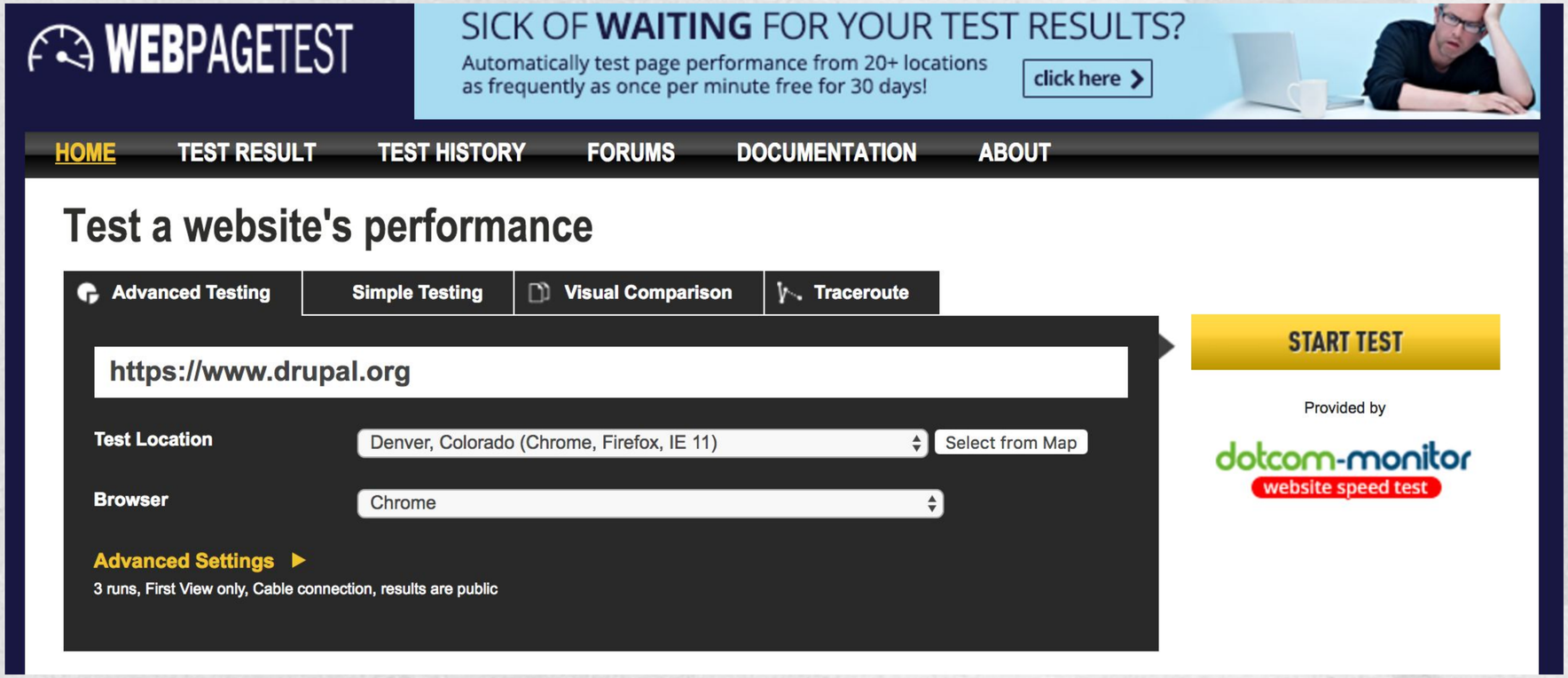
1. Measurement
2. Networking (HTTP/Browsers)
3. UI (HTML/CSS)
4. Code (JS)

Q&A after each section to focus the interactions.

Measurement

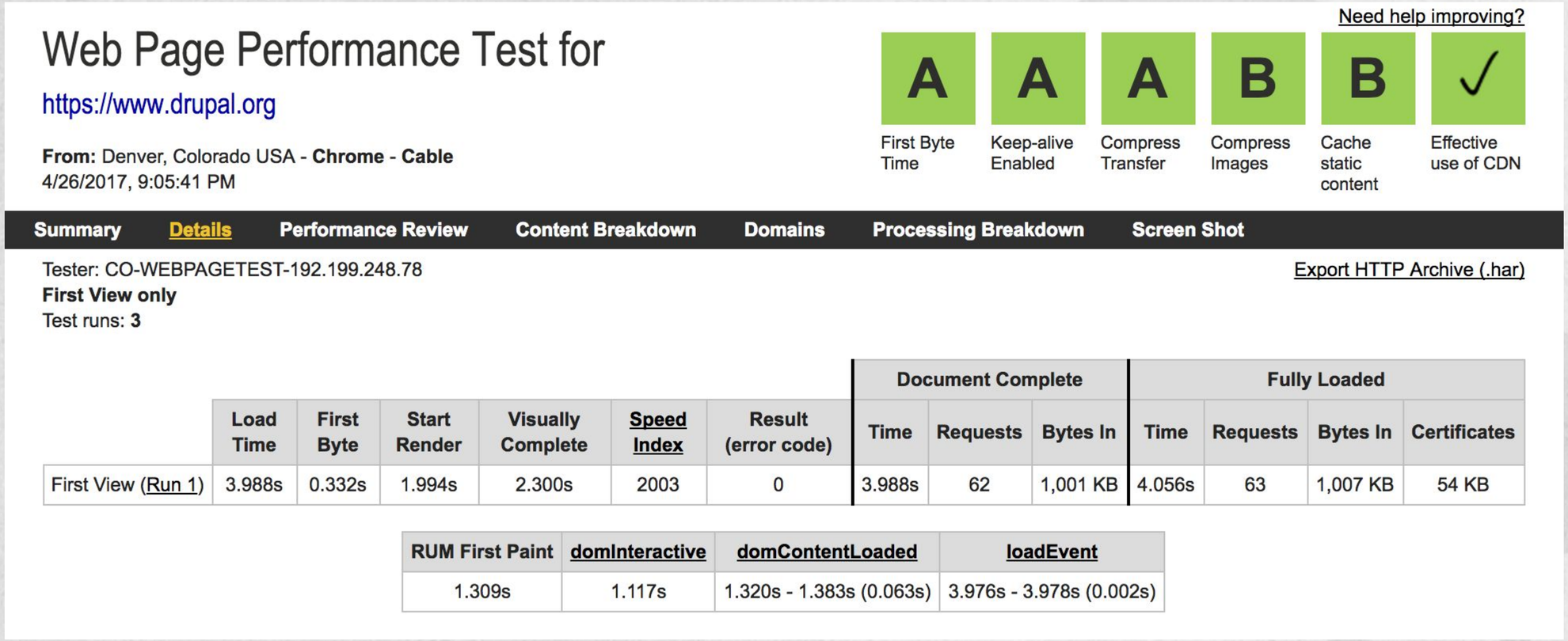
Measurement IS performance

WebPageTest



free and awesome

WebPageTest



Web Page Performance Test for

https://www.drupal.org

From: Denver, Colorado USA - **Chrome** - **Cable**
4/26/2017, 9:05:41 PM

Need help improving?

A	A	A	B	B	✓
First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN

Summary | Details | Performance Review | Content Breakdown | Domains | Processing Breakdown | Screen Shot

Tester: CO-WEBPAGETEST-192.199.248.78
First View only
Test runs: **3**

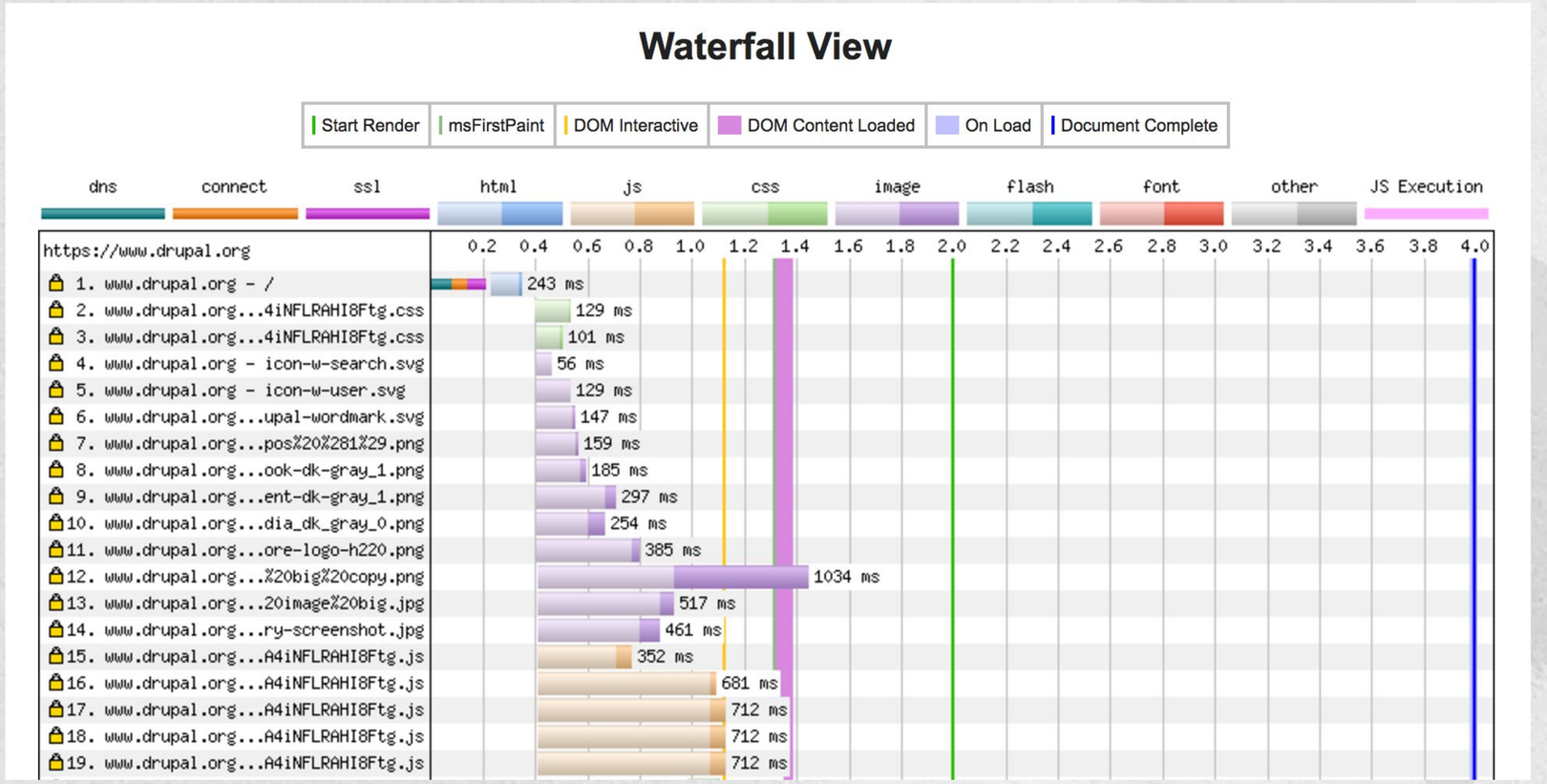
Export HTTP Archive (.har)

							Document Complete			Fully Loaded			
	Load Time	First Byte	Start Render	Visually Complete	Speed Index	Result (error code)	Time	Requests	Bytes In	Time	Requests	Bytes In	Certificates
First View (Run 1)	3.988s	0.332s	1.994s	2.300s	2003	0	3.988s	62	1,001 KB	4.056s	63	1,007 KB	54 KB

RUM First Paint	domInteractive	domContentLoaded	loadEvent
1.309s	1.117s	1.320s - 1.383s (0.063s)	3.976s - 3.978s (0.002s)

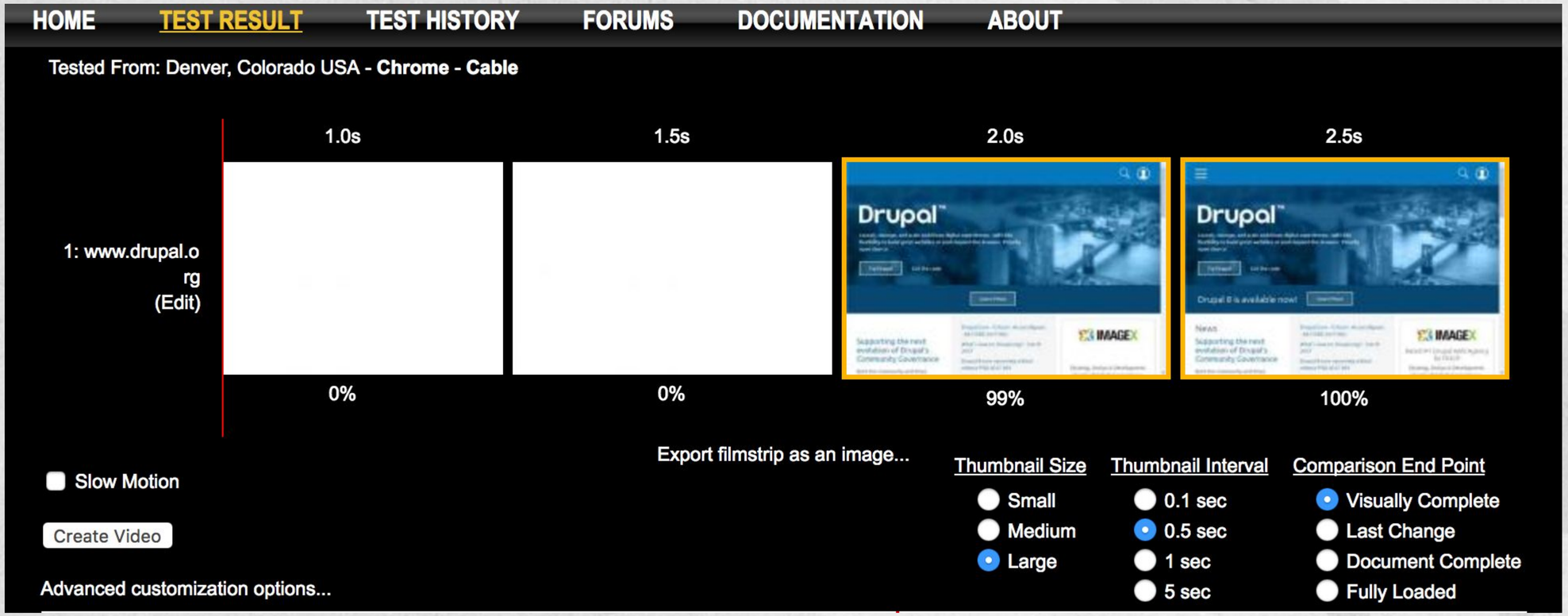
clear metrics and reporting

WebPageTest



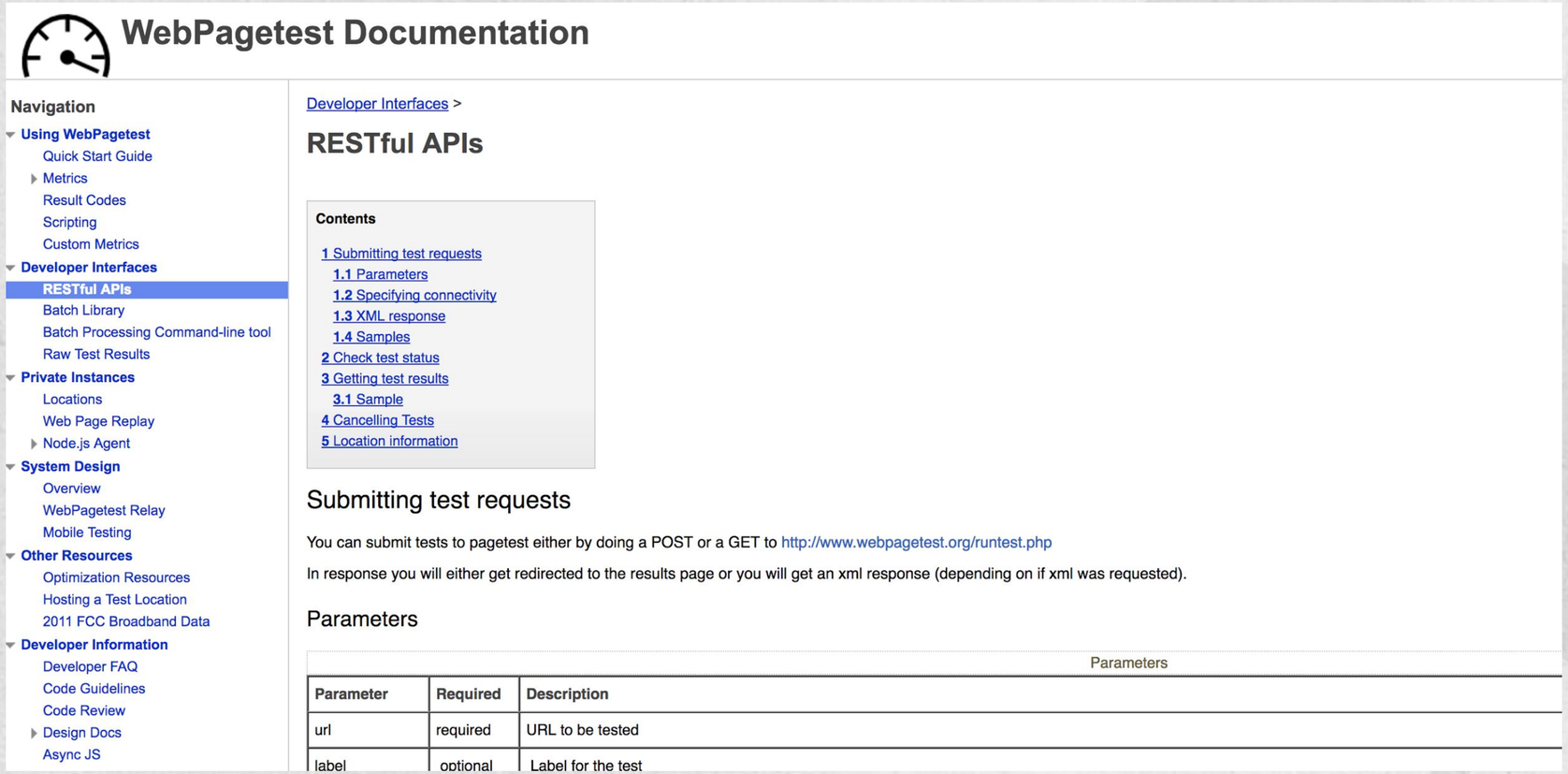
killer waterfall

WebPageTest



recording!!!

WebPageTest



WebPagetest Documentation

Navigation

- Using WebPagetest
  - Quick Start Guide
  - Metrics
  - Result Codes
  - Scripting
  - Custom Metrics
- Developer Interfaces
  - RESTful APIs
  - Batch Library
  - Batch Processing Command-line tool
  - Raw Test Results
- Private Instances
  - Locations
  - Web Page Replay
  - Node.js Agent
- System Design
  - Overview
  - WebPagetest Relay
  - Mobile Testing
- Other Resources
  - Optimization Resources
  - Hosting a Test Location
  - 2011 FCC Broadband Data
- Developer Information
  - Developer FAQ
  - Code Guidelines
  - Code Review
  - Design Docs
  - Async JS

Developer Interfaces >

RESTful APIs

Contents

1 Submitting test requests
- 1.1 Parameters
- 1.2 Specifying connectivity
- 1.3 XML response
- 1.4 Samples

2 Check test status
3 Getting test results
- 3.1 Sample

4 Cancelling Tests
5 Location information

Submitting test requests

You can submit tests to pagetest either by doing a POST or a GET to http://www.webpagetest.org/runtest.php

In response you will either get redirected to the results page or you will get an xml response (depending on if xml was requested).

Parameters

Parameters

Parameter	Required	Description
url	required	URL to be tested
label	optional	Label for the test

very handy API

WebPageTest

Some key metrics that are important to you.

1. SpeedIndex (calculated metric -- target < 1000)
2. First Paint (reported by browser)
3. Start Render (capture showing non-blank frame)
4. Visually Complete (a.k.a. End Render)
5. Document Complete Metrics (when the DOM is processed)
6. Fully Loaded Metrics (when all is said and done)

https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics

Great WPT plug. What’s the point?

Metrics can be monitored
Monitoring can be automated

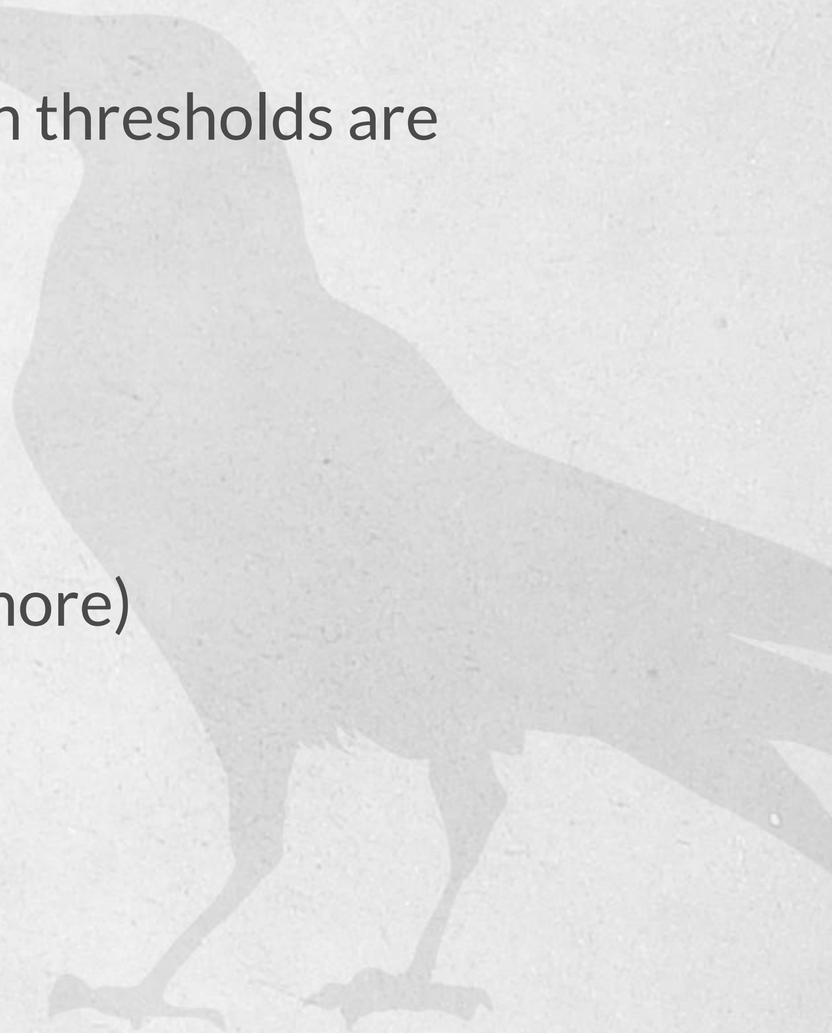
Performance budget

Budgets make it possible to set constraints and alert when thresholds are breached. It’s a really good idea.

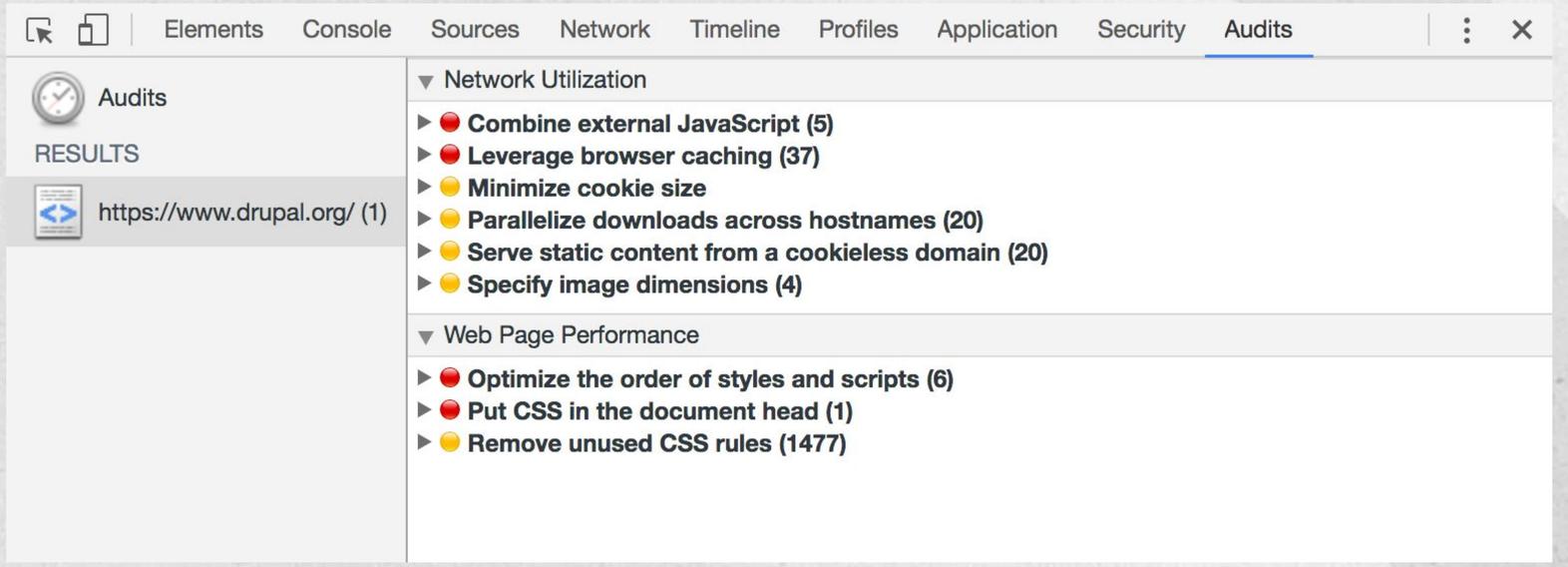
Some options:

1. WPT private instance with some automation
2. NPM packages (varying methods)
3. Services (dynatrace.com, webperformance.com, lots more)
4. Google PageSpeed Insights (API automation)
5. More that I am definitely leaving out

Great information on this topic online.

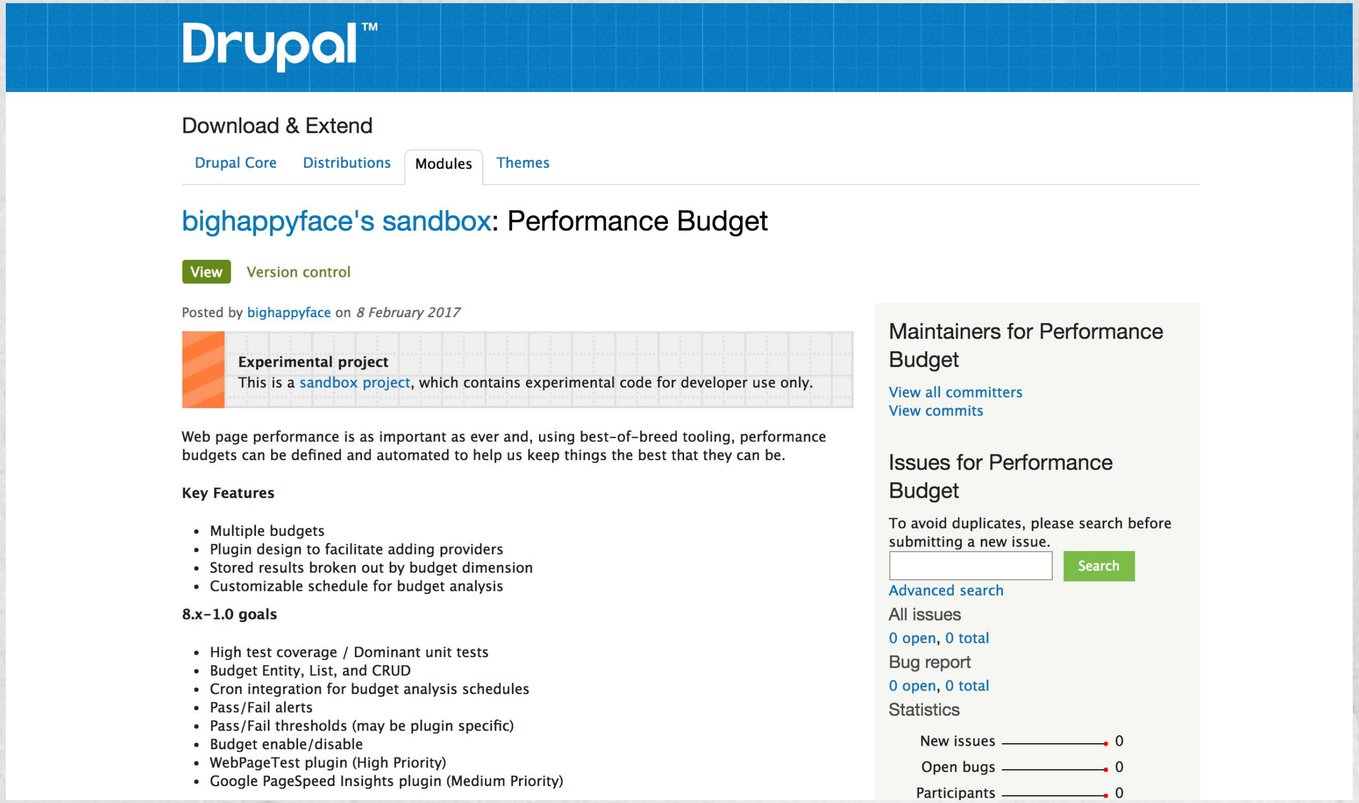


Dev tools for measurement



Chrome Audits Tab

D8 highlights for measurement



My sandbox module

Q&A for measurement?

Networking (HTTP/Browsers)

HTTP/2

What's new?

1. Binary instead of textual (easier to parse, more compact)
2. Fully multiplexed (parallel requests over a single origin connection)
3. Header compression to reduce overhead (HPACK, reduces round-trips)
4. Support for server pushes (proactive caching, good practices still TBD)

https://http2.github.io/faq/

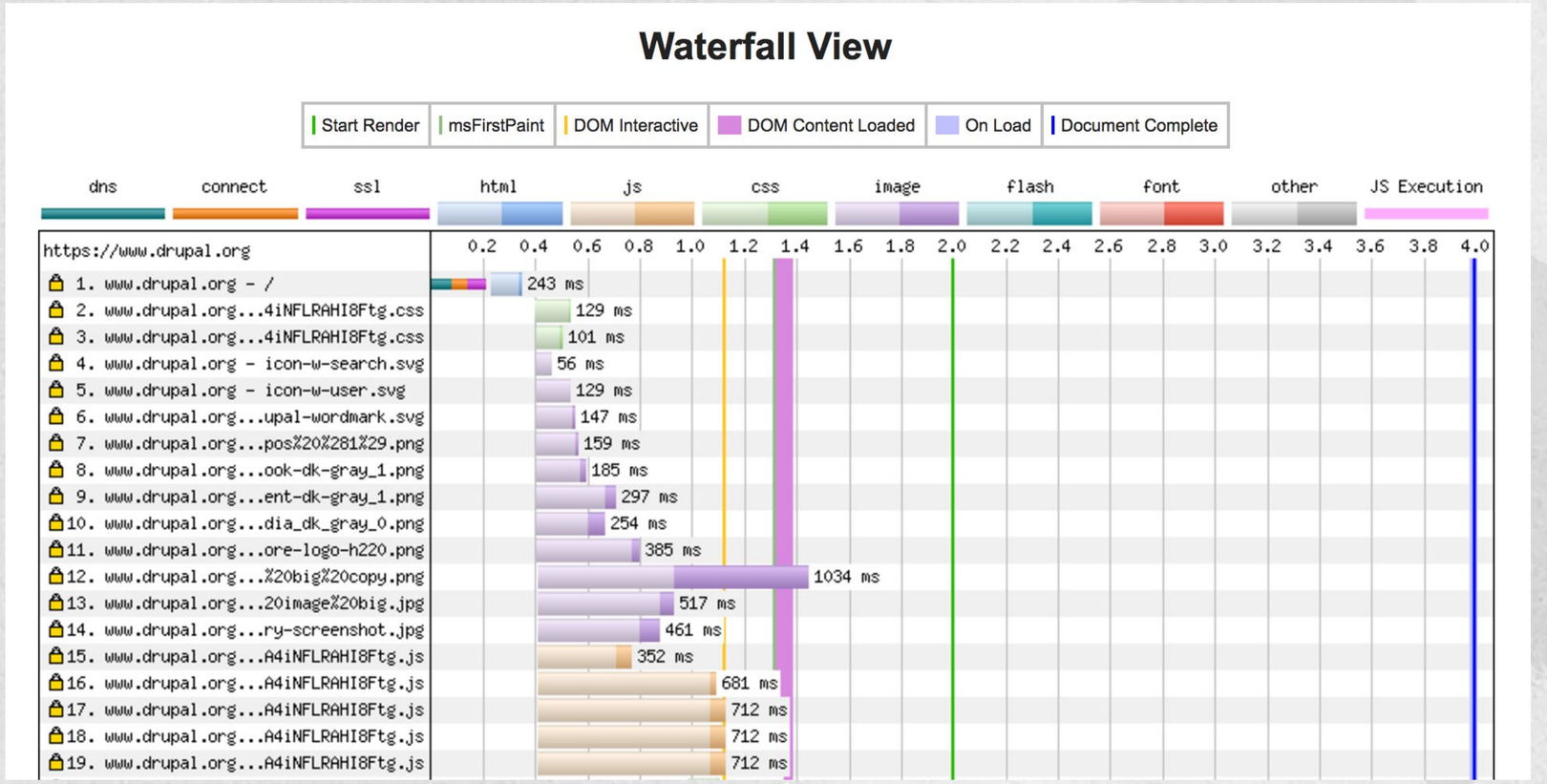
Headers

Headers describe documents and heavily influence the browser.

1. Cache-Control (the big one)
2. Vary (rules for varying caches - encoding, cookie, user agent)
3. ETag (cache validation)
4. Cookies (heavy, large influence on cache)
5. Link (canonical/alternative URLs, Resource Hints - preconnect/prefetch)

https://developer.mozilla.org/en-US/docs/Web/HTTP/Caching
https://developer.mozilla.org/en-US/docs/Web/HTML/Link_types

Waterfall optimization



know thy waterfall

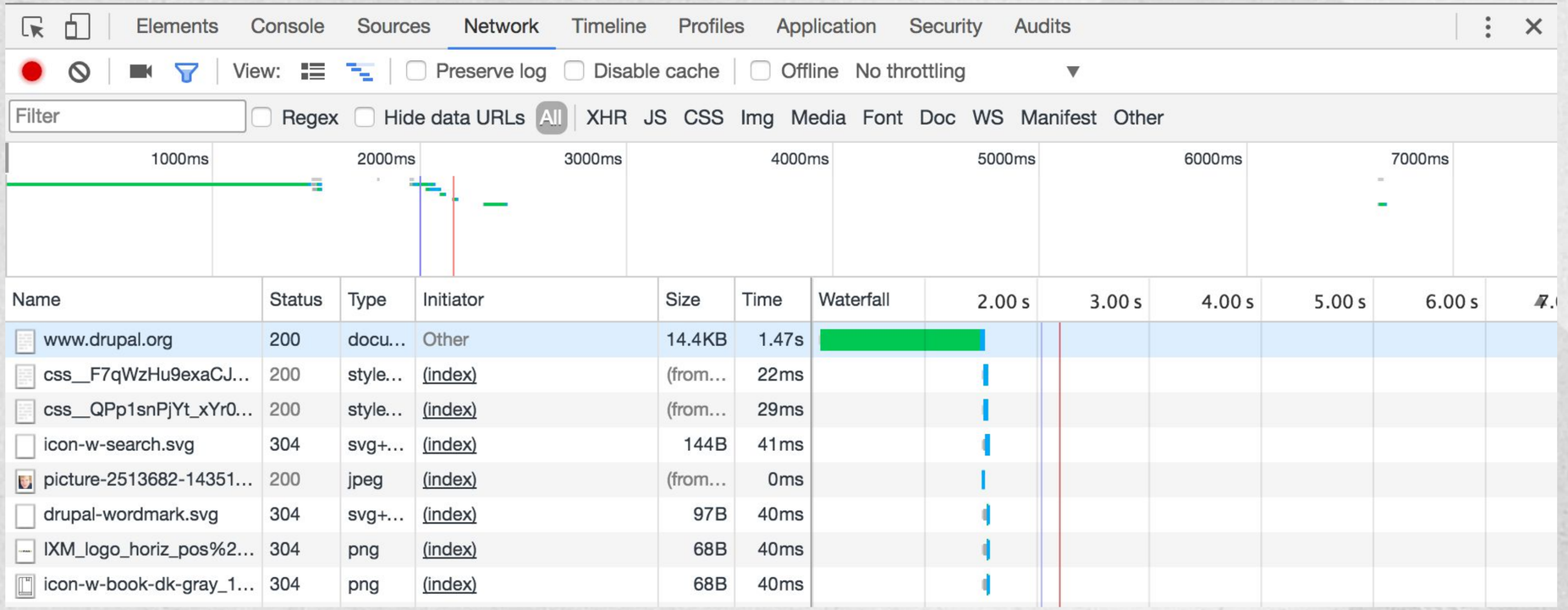
Waterfall optimization

Some tips:

1. Get requests for your domain to the top
2. Understand the impacts of DNS, SSL Negotiation
3. Understand how HTTP/2 vs. HTTP/1.1 waterfalls differ
4. Observe impacts of resource hints
5. Observe nuances of third-party drag

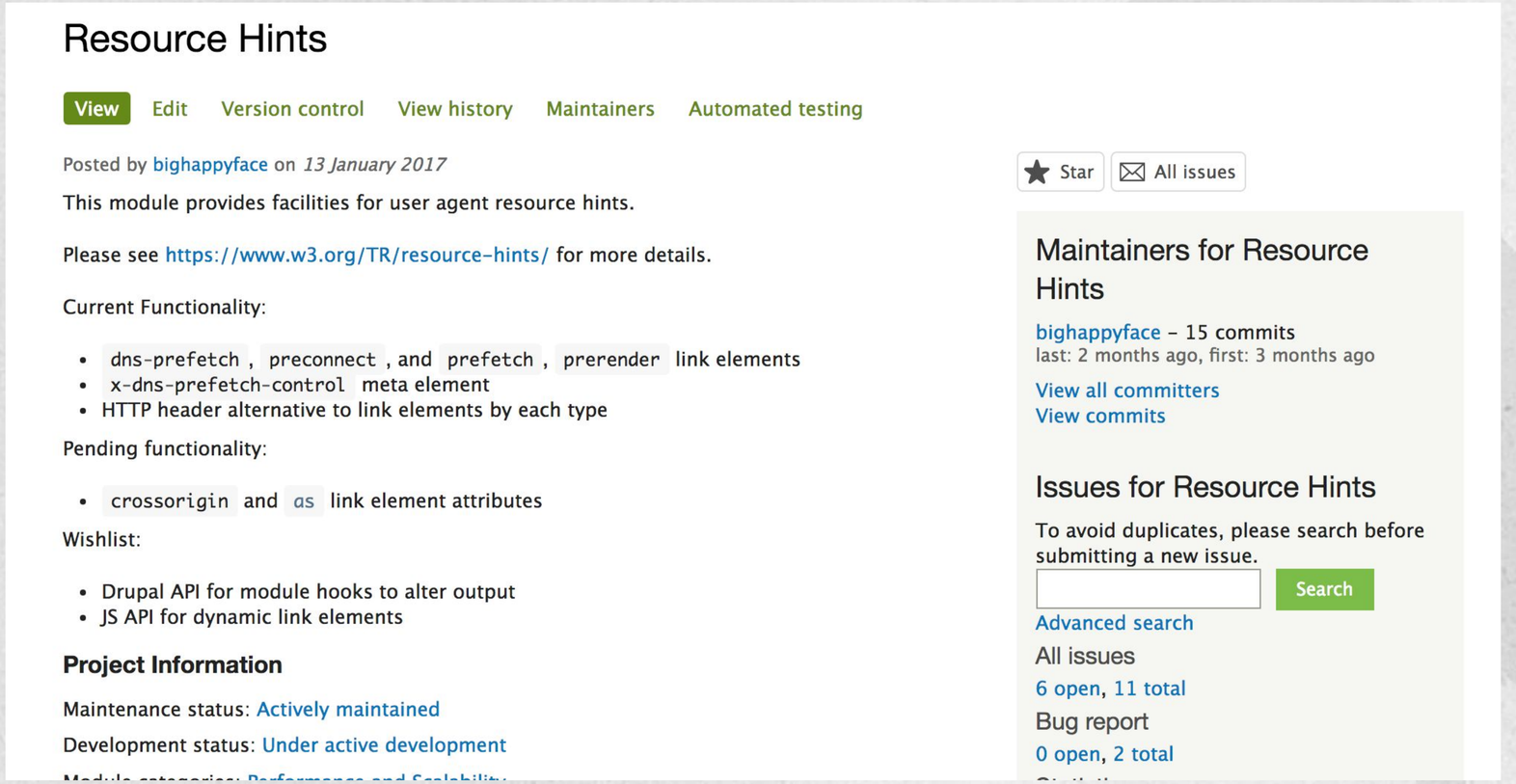
Lots of guidance online.

Dev tools for networking



Chrome Network Tab

D8 highlights for networking



Resource Hints

View Edit Version control View history Maintainers Automated testing

Posted by bighappyface on *13 January 2017*

This module provides facilities for user agent resource hints.

Please see https://www.w3.org/TR/resource-hints/ for more details.

Current Functionality:

- `dns-prefetch` , `preconnect` , and `prefetch` , `prerender` link elements
- `x-dns-prefetch-control` meta element
- HTTP header alternative to link elements by each type

Pending functionality:

- `crossorigin` and `as` link element attributes

Wishlist:

- Drupal API for module hooks to alter output
- JS API for dynamic link elements

Project Information

Maintenance status: Actively maintained

Development status: Under active development

Star All issues

Maintainers for Resource Hints

bighappyface – 15 commits
last: 2 months ago, first: 3 months ago

View all committers
View commits

Issues for Resource Hints

To avoid duplicates, please search before submitting a new issue.

Search

Advanced search

All issues

6 open, 11 total

Bug report

0 open, 2 total

Resource Hints module

Q&A for networking?

UI (HTML/CSS)

Critical render path

What's a "critical render path?"

Set of steps browsers must take to convert resources into a useable site.

1. Understand your resources and how they affect rendering.
2. Understand the waterfall (blocking/waits, order of load, options to inline)
3. Minimize critical resources (minimum CSS, JS, images, fonts)
4. Minimize round-trips

https://developers.google.com/web/fundamentals/performance/critical-rendering-path/analyzing-crp

Fonts

Web fonts are the bane of performance.

1. Overloaded with glyphs, diacritics, metadata, kerning...
2. Too many formats (browser support concerns, Google Fonts tries to help)
3. EOT and TTF formats are not compressed by default
4. Font redraws and FOUC (flash of unrendered content)
5. Always blocks

https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/webfont-optimization

Dynamic/transformative styling

The future of CSS/styling?

1. PostCSS (transforming styles with JS)
   Note: Beware super-selectors (selector specificity) side effects
2. CSS containment (element isolation to improve layout recalculation)

http://postcss.org
https://developers.google.com/web/updates/2016/06/css-containment

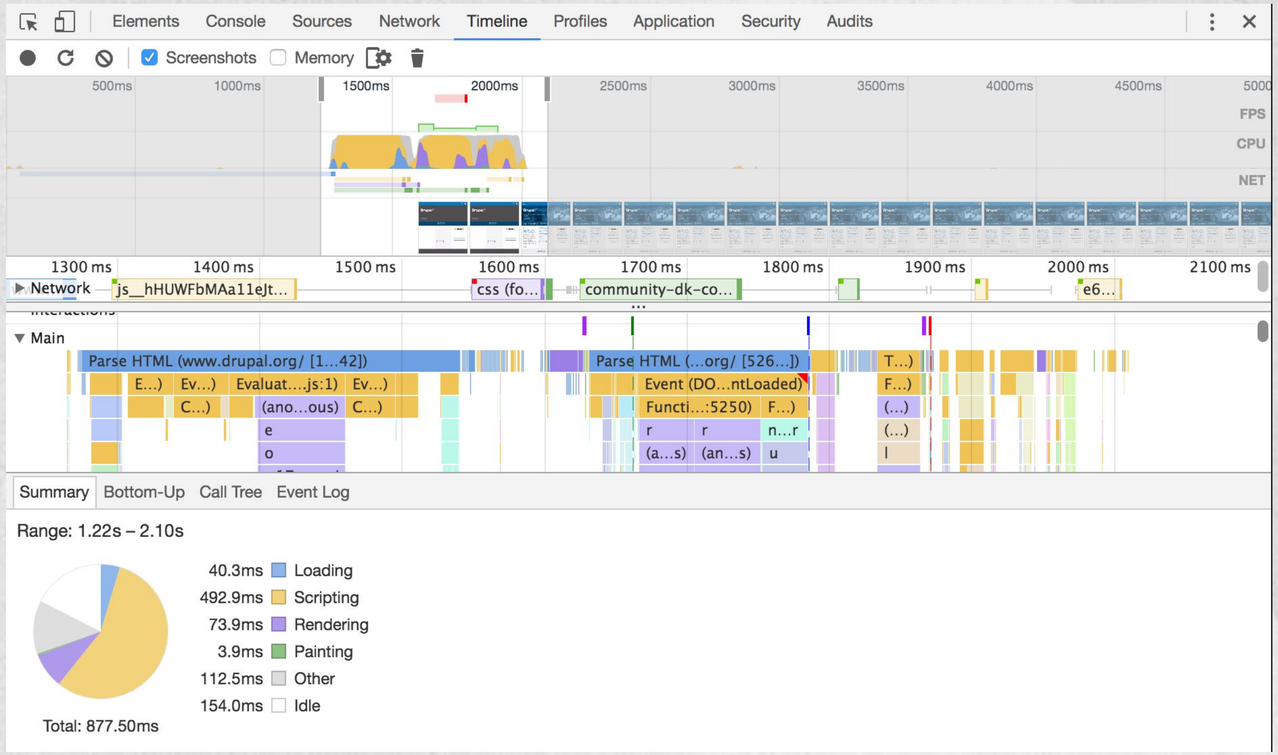
Web Components

Self-contained building blocks for building all the things.

1. Tech: HTML Templates, Shadow DOM, Custom elements and HTML Imports
2. Benefits: CSS scoping, DOM encapsulation, composition, isolation

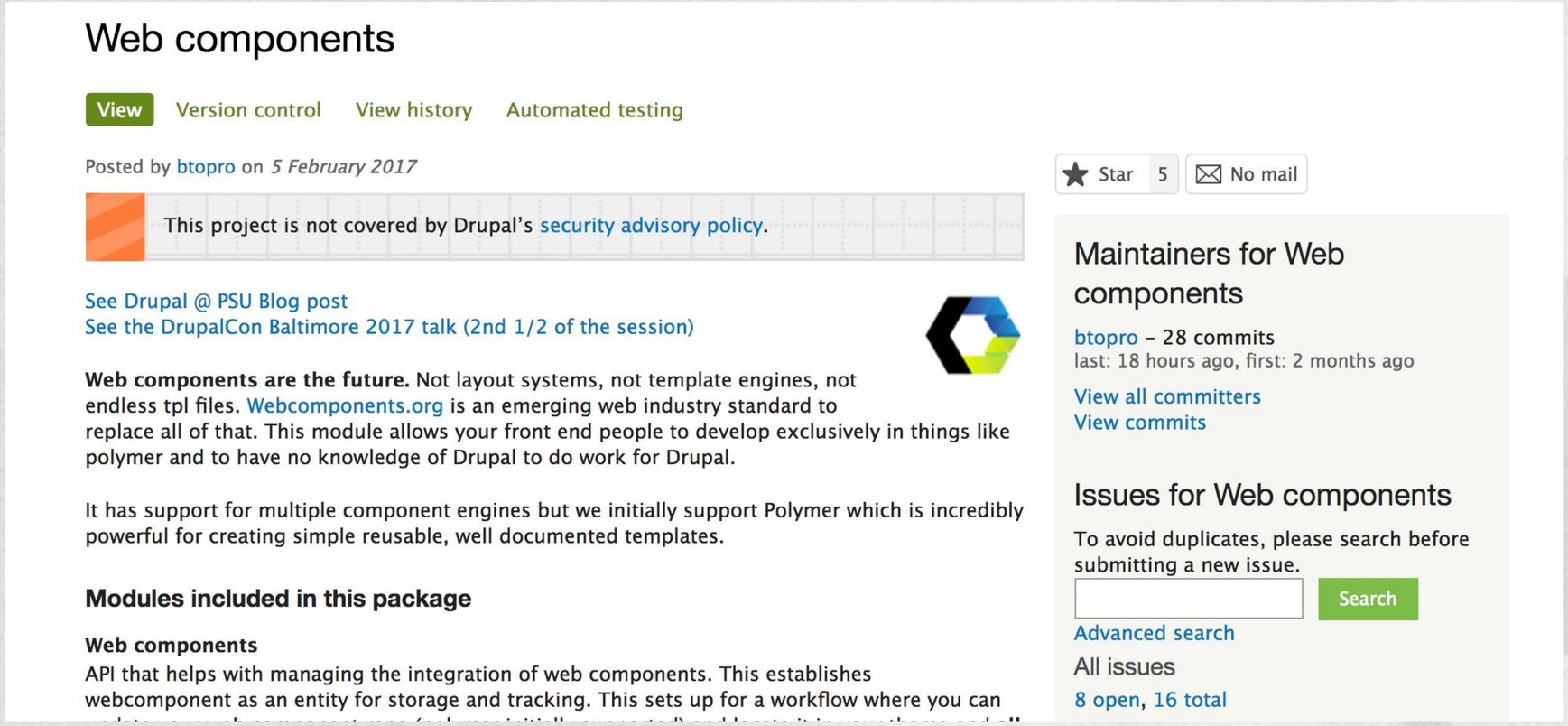
https://developer.mozilla.org/en-US/docs/Web/Web_Components
https://developers.google.com/web/fundamentals/getting-started/primers/shadowdom

Dev tools for UI



Chrome Timeline Tab

D8 highlights for UI



Web components

View | Version control | View history | Automated testing

Posted by btopro on *5 February 2017*

This project is not covered by Drupal's security advisory policy.

See Drupal @ PSU Blog post
See the DrupalCon Baltimore 2017 talk (2nd 1/2 of the session)

Web components are the future. Not layout systems, not template engines, not endless tpl files. Webcomponents.org is an emerging web industry standard to replace all of that. This module allows your front end people to develop exclusively in things like polymer and to have no knowledge of Drupal to do work for Drupal.

It has support for multiple component engines but we initially support Polymer which is incredibly powerful for creating simple reusable, well documented templates.

Modules included in this package

Web components
API that helps with managing the integration of web components. This establishes webcomponent as an entity for storage and tracking. This sets up for a workflow where you can

Star 5 | No mail

Maintainers for Web components

btopro – 28 commits
last: 18 hours ago, first: 2 months ago

View all committers
View commits

Issues for Web components

To avoid duplicates, please search before submitting a new issue.

Search

Advanced search

All issues

8 open, 16 total

Web Components module

Q&A for UI?

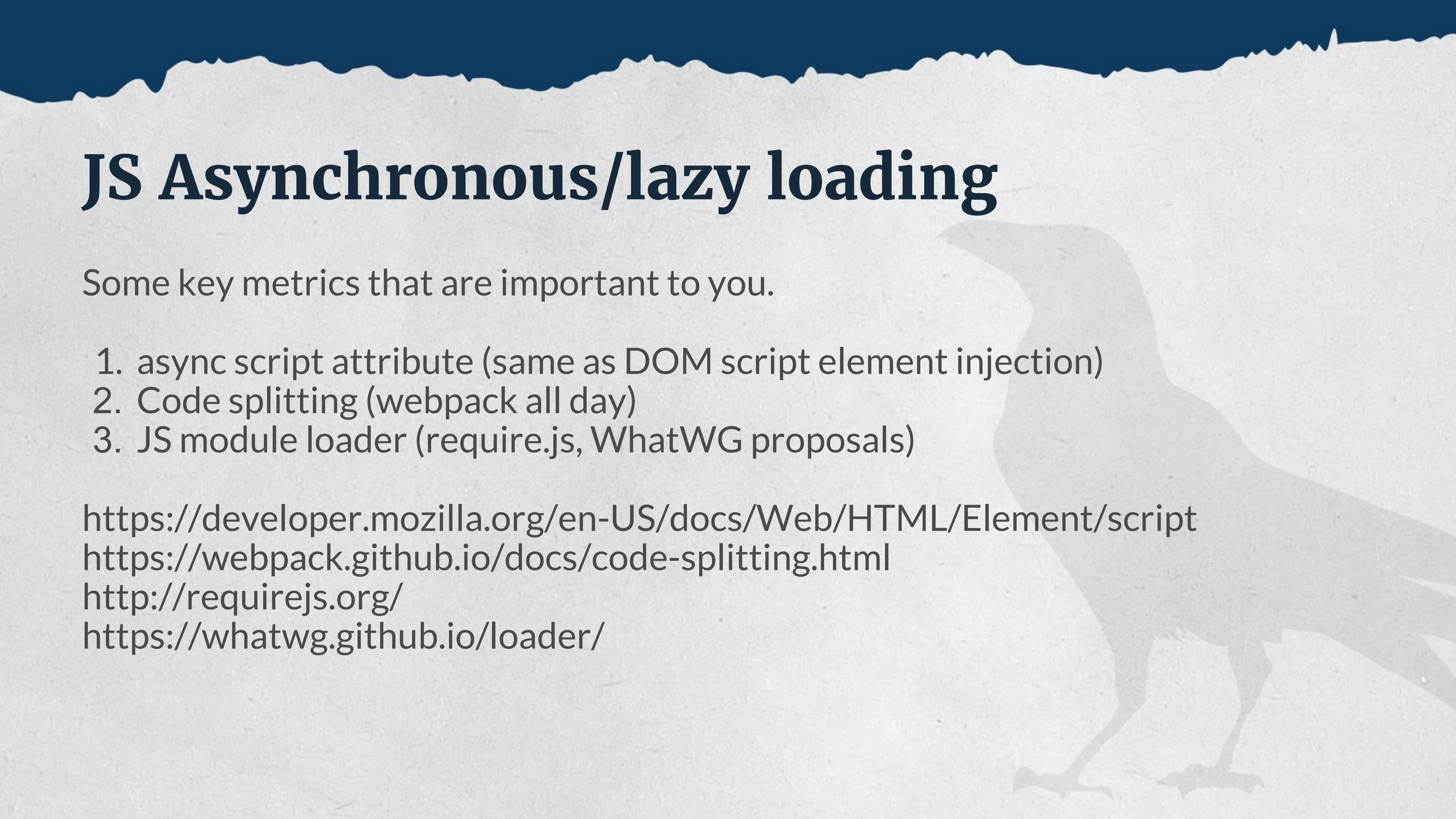
Code (JS)

JS Asynchronous/lazy loading

Some key metrics that are important to you.

1. async script attribute (same as DOM script element injection)
2. Code splitting (webpack all day)
3. JS module loader (require.js, WhatWG proposals)

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script
https://webpack.github.io/docs/code-splitting.html
http://requirejs.org/
https://whatwg.github.io/loader/



JS Transpiling and ES6+ browser support

Modern JS and all of its goodness.

1. Babel, babel, babel
2. babel-preset-env (automatically determine the Babel plugins and polyfills you need - smaller artifacts)

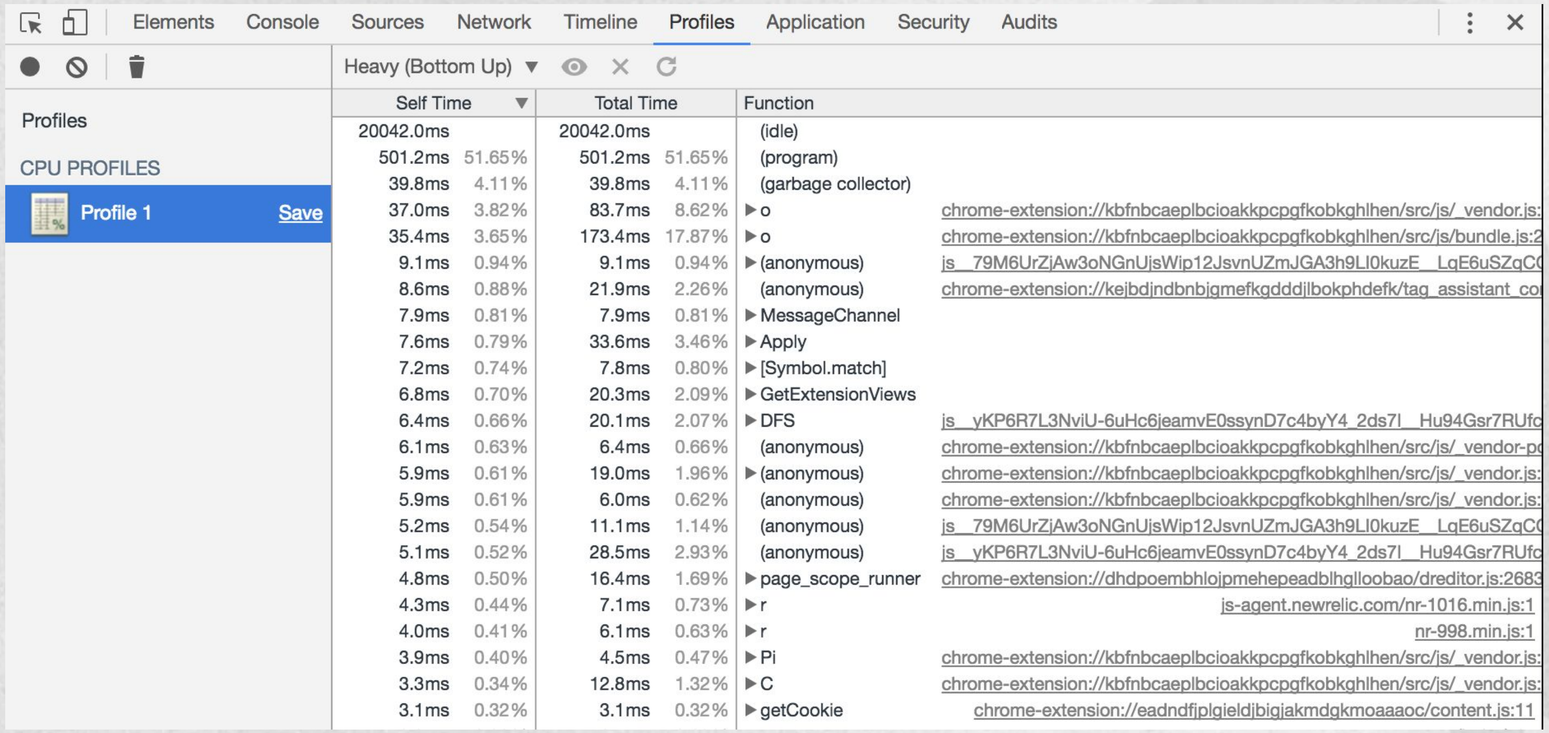
https://babeljs.io
https://github.com/babel/babel-preset-env

General code performance

1. ASM.js (optimizable, low-level subset of JavaScript)
2. Slow code is slow code (measure complexity)

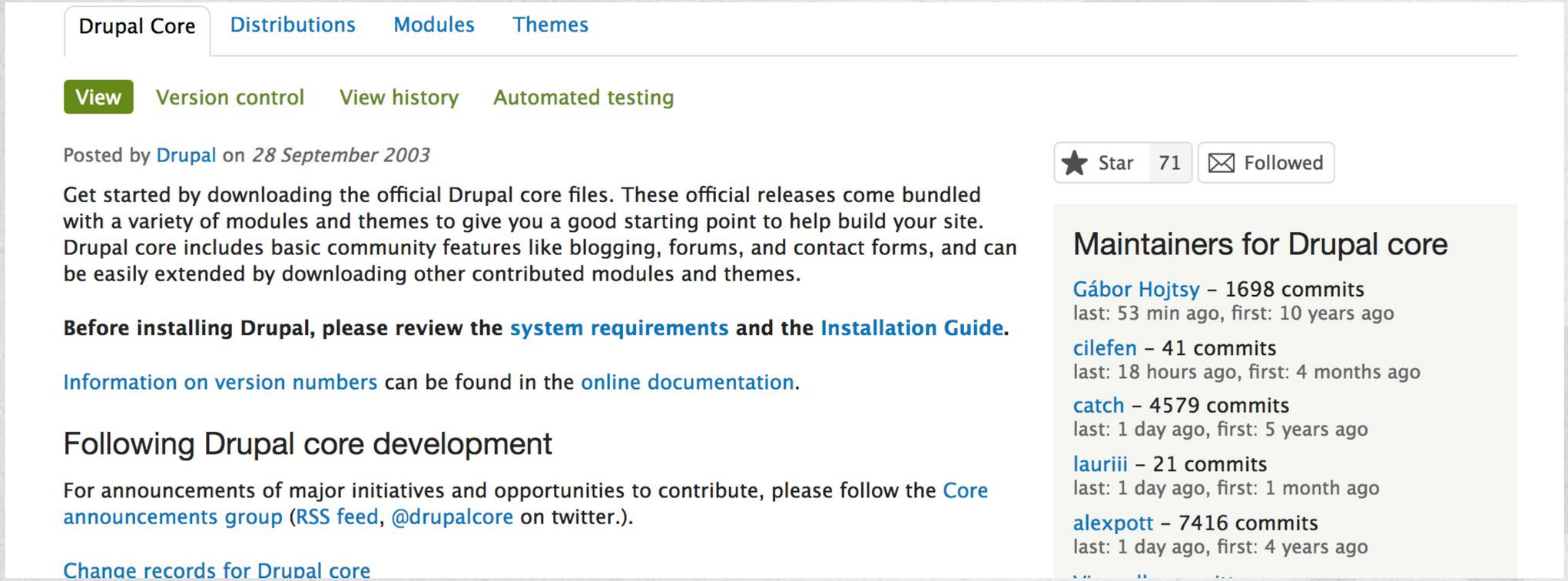
http://asmjs.org/
http://bigocheatsheet.com/

Dev tools for code



Chrome Profiles Tab

D8 highlights for code



Drupal 8 is better for JS

Q&A for code?

Some thoughts

low performance

high performance

Performance challenges

(get out of your comfort zone)

Performance frustrates

(progress comes slow)

Performance rewards

(validation all around)

One more thing

2017.texascamp.org - June 2-4, 2017

SUBMISSION DEADLINE FAST APPROACHING

Visit this page to keep up with Texas Camp-related announcements.

2017.texascamp.org/submit-session (closes 4/28)

Join Us for Contribution Sprints

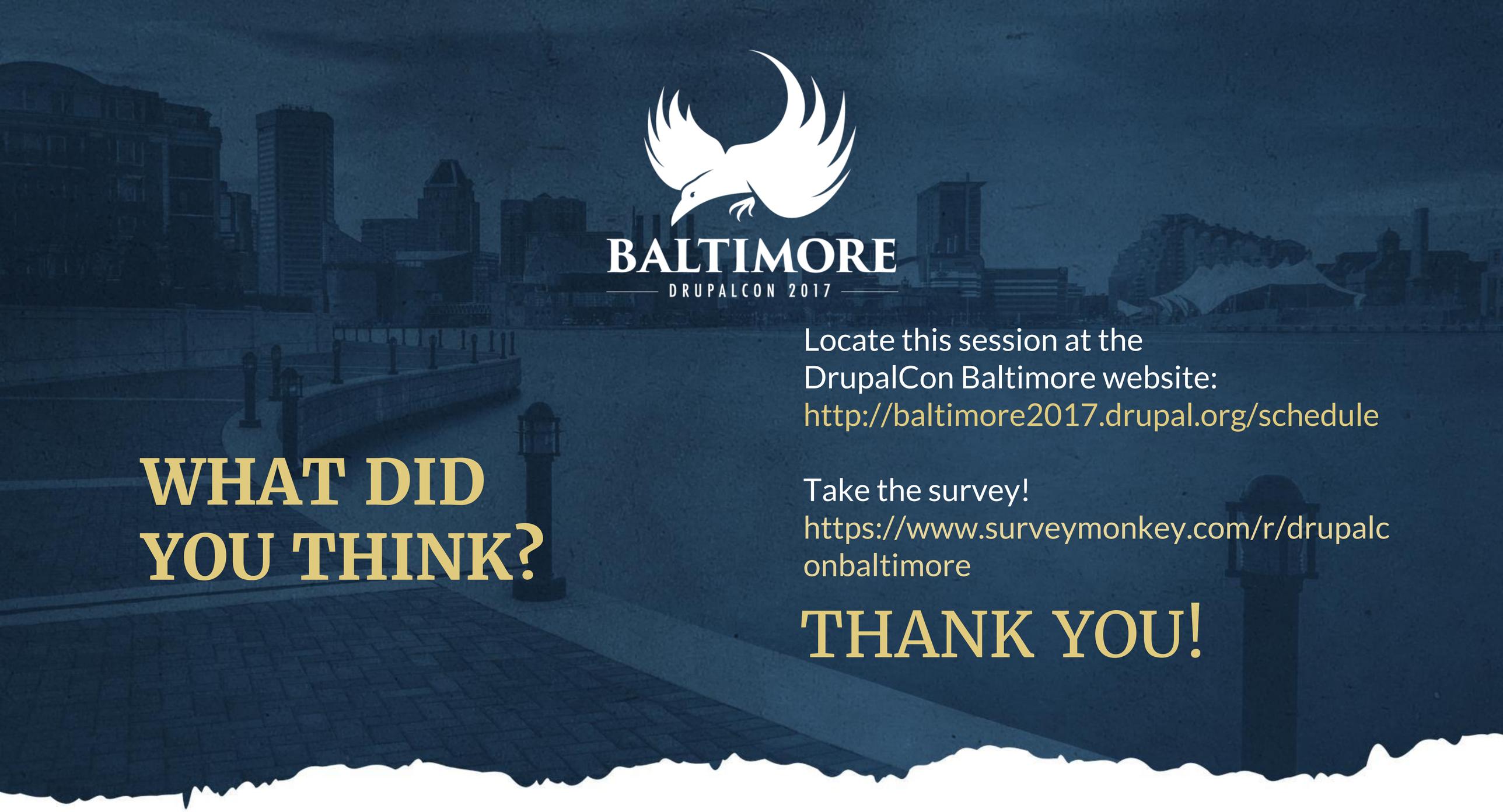
Friday, April 28, 2017

First-Time Sprinter Workshop
9:00am-12:00pm
Room: 307-308

Mentored Core Sprint
9:00am-12:00pm
Room:301-303

General Sprints
9:00am-6:00pm
Room:309-310

#drupalsprints



WHAT DID YOU THINK?

Locate this session at the DrupalCon Baltimore website:
http://baltimore2017.drupal.org/schedule

Take the survey!
https://www.surveymonkey.com/r/drupalconbaltimore

THANK YOU!

Web Performance in 2017

with @bighappyface