

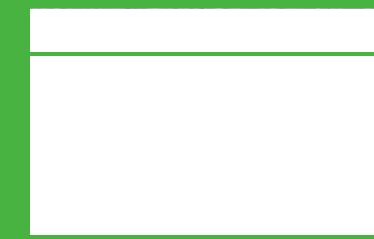
DEBUGGING YOUR CODE WITH CHROME DEV TOOLS

#DZY

Animation used throughout this presentation

Find Out More!

Search for terms on these slides!



tiny.cc/kf-wc-2016

Debug:

verb [T] (REMOVE MISTAKES)

to remove bugs (= mistakes) from a computer program:

to debug a program

Cambridge Dictionaries Online

Debugging:

Monitoring application flow and state.

Debugging HTML & CSS

The Elements Tab

Right Click + Inspect Element

Ctrl (⌘) + Shift + C

#DZY

3D Solar System

by @JulianGarnier

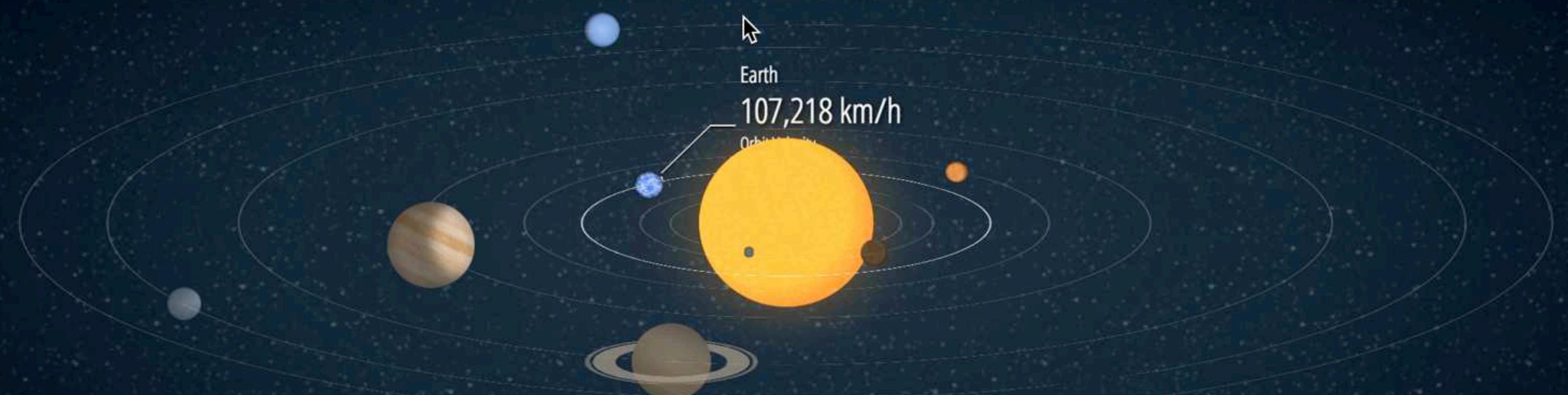
2D

+

Speed

Size

Distance



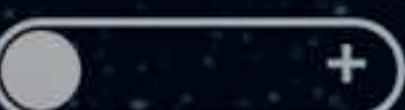
Sun Mercury Venus Earth Mars Jupiter Saturn Uranus Neptune

3D Solar System

by @JulianGarnier



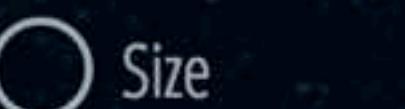
2D



+



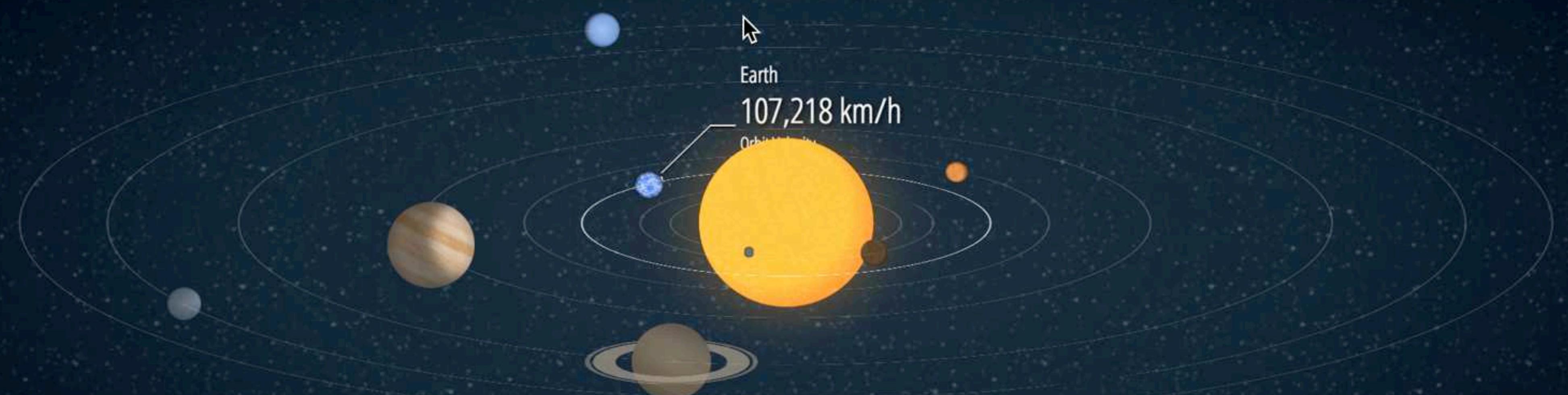
Speed



Size



Distance



Sun Mercury Venus Earth Mars Jupiter Saturn Uranus Neptune

Filter

+ .cls ⌂ ⌂

body { idhuG:52

font-size: 10px;
font-family: 'Open Sans', sans-serif;
font-weight: 300;
background-color: #08090A; ↴

}

body { idhuG:31

line-height: 1;

}

html, body, div, span, applet, object, iframe, idhuG:20

Styles Event Listeners DOM Breakpoints Properties

Filter

+ .cls ⌂ ⌂

body { idhuG:52

font-size: 10px;
font-family: 'Open Sans', sans-serif;
font-weight: 300;
background-color: #08090A; ↴

}

body { idhuG:31

line-height: 1;

}

html, body, div, span, applet, object, iframe, idhuG:20

3D Solar System

by @JulianGarnier



2D



+



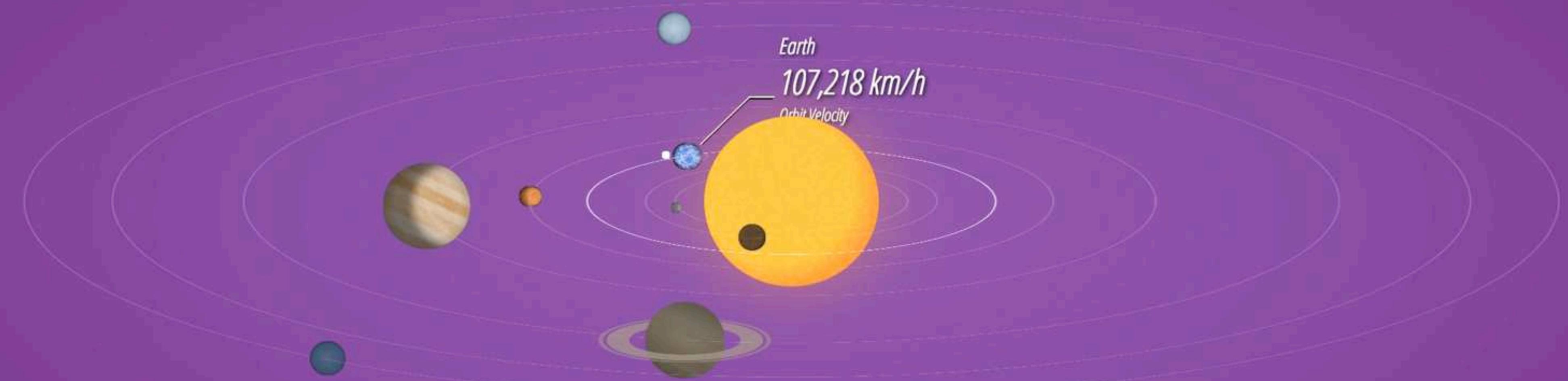
Speed



Size



Distance



Sun

Mercury

Venus

Earth

Mars

Jupiter

Saturn

Uranus

Neptune

3D Solar System

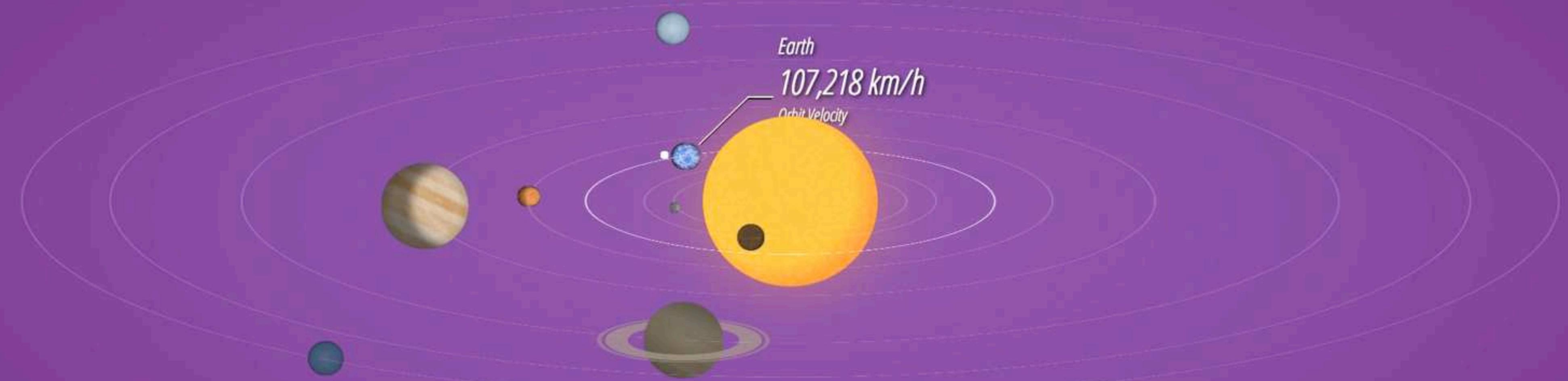
by @JulianGarnier



Speed

Size

Distance



Sun

Mercury

Venus

Earth

Mars

Jupiter

Saturn

Uranus

Neptune

Filter



```
element.style {  
}
```

```
#galaxy {  
    position: relative;  
    width: 100%;  
    height: 100%;  
}  
  
: 
```

```
html, body, div, span, applet, object, iframe, h1, h2, h3, idhuG:20  
h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite,  
code, del, dfn, em, img, ins, kbd, q, s, samp, small, strike, strong,  
sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li, fieldset,  
form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, canvas, details, embed, figure, figcaption, footer,  
header, hgroup, menu, nav, output, ruby, section, summary, time,  
mark, audio, video {
```

Filter



```
element.style {  
}
```

```
#galaxy {  
    position: relative;  
    width: 100%;  
    height: 100%;  
}  
  
: 
```

```
html, body, div, span, applet, object, iframe, h1, h2, h3, idhuG:20  
h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite,  
code, del, dfn, em, img, ins, kbd, q, s, samp, small, strike, strong,  
sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li, fieldset,  
form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, canvas, details, embed, figure, figcaption, footer,  
header, hgroup, menu, nav, output, ruby, section, summary, time,  
mark, audio, video {
```

Filter

+ .cls ¶ ◆

```
element.style {  
}
```

```
#galaxy {  
    position: relative;  
    width: 100%;  
    height: 100%;  
}
```

idhuG:68

```
html, body, div, span, applet, object, iframe, h1, h2, h3, idhuG:20  
h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite,  
code, del, dfn, em, img, ins, kbd, q, s, samp, small, strike, strong,  
sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li, fieldset,  
form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, canvas, details, embed, figure, figcaption, footer,  
header, hgroup, menu, nav, output, ruby, section, summary, time,  
mark, audio, video {
```

Filter

+ .cls ¶ ◆

```
element.style {  
}
```

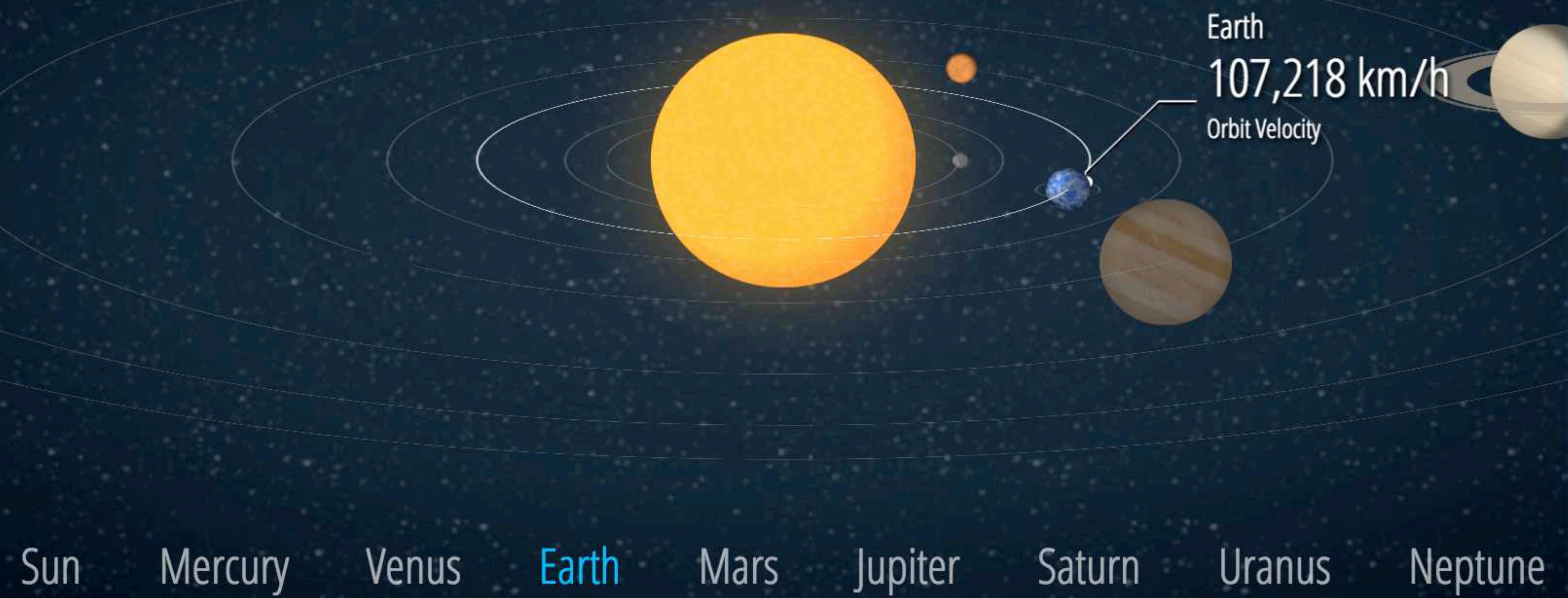
```
#galaxy {  
    position: relative;  
    width: 100%;  
    height: 100%;  
}
```

idhuG:68

```
html, body, div, span, applet, object, iframe, h1, h2, h3, idhuG:20  
h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite,  
code, del, dfn, em, img, ins, kbd, q, s, samp, small, strike, strong,  
sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li, fieldset,  
form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, canvas, details, embed, figure, figcaption, footer,  
header, hgroup, menu, nav, output, ruby, section, summary, time,  
mark, audio, video {
```

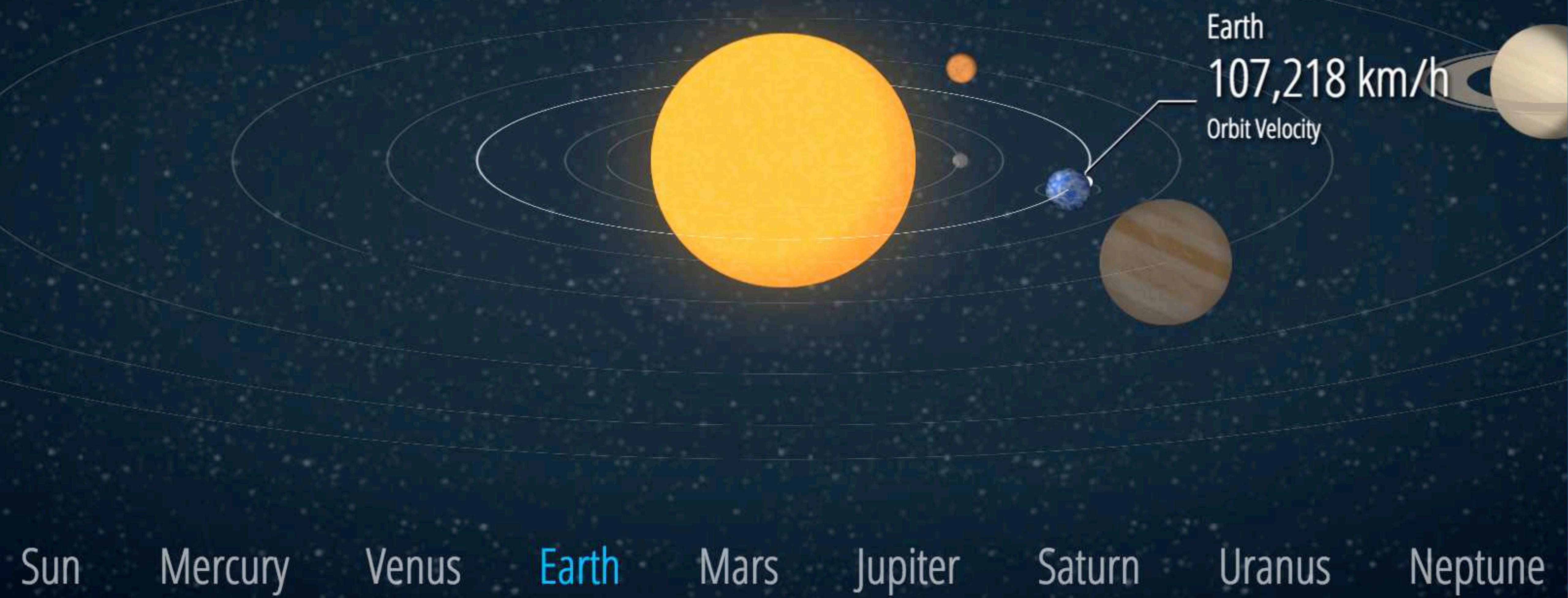
3D Solar System

by @JulianGarnier



3D Solar System

by @JulianGarnier



```
<!DOCTYPE html>
<html>
  ><head>...</head>
  ...<body class="home-template">
    ><header class="main-header no-cover">...</header>
    ><main id="content" class="content" role="main">...</main>
    ><div class="decoration">...</div>
    ><footer class="site-footer clearfix">...</footer>
      <script src="/public/jquery.min.js?v=80e65900a2"></script>
      <script type="text/javascript" src="/assets/dist/scripts.min.js?v=80e65900a2"></script>
    </body>
</html>
```

```
<!DOCTYPE html>
<html>
  ><head>...</head>
  ...<body class="home-template">
    ><header class="main-header no-cover">...</header>
    ><main id="content" class="content" role="main">...</main>
    ><div class="decoration">...</div>
    ><footer class="site-footer clearfix">...</footer>
      <script src="/public/jquery.min.js?v=80e65900a2"></script>
      <script type="text/javascript" src="/assets/dist/scripts.min.js?v=80e65900a2"></script>
    </body>
</html>
```

Blog Tool, Publishing Platfo ×

https://wordpress.org

Apps Bookmarks Framerate

Responsive ▾ 400 × 1113 49% ▾

WORDPRESS.ORG

Ready to get started? Download WordPress

Themes

Twenty Sixteen

Twenty Fourteen

WordPress is web software you can use to create a beautiful website, blog, or app. We like to say that WordPress is both free and priceless at the same time.

The core software is built by hundreds of community volunteers, and when you're ready for more there are thousands of plugins and themes available to transform your site into almost anything you can imagine. Over 60 million people have chosen WordPress to power the place on the web they call "home" — we'd love you to join the family.

Ready to get started? Download WordPress 4.4.2

WordPress is also available in English (UK) (also Cymraeg, Gàidhlig).

WordPress Swag

Elements Console Sources Network ▾

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en">
  <head profile="http://gmpg.org/xfn/11">...</head>
  ...<body id="wordpress-org">
    <div id="wporg-header">...</div>
    <div id="download-mobile">...</div>
      <!-- feed is cached -->
    <div id="home>Welcome" style="padding-bottom: 14px;">...</div>
    <div id="lang-guess-wrap">...</div>
    <div id="home-below">...</div>
    <div id="wporg-footer">...</div>
    <script type="text/javascript">...</script>
    <script type="text/javascript">...</script>
    <script type="text/javascript">_qevents.push( { qacct:"p-18-mFFk41448M" } )</script>
```

html body#wordpress-org

Styles Event Listeners DOM Breakpoints Properties

Filter + .cls ⚡

```
element.style {
```

```
body { wp4.css?40:16
  font-size: 62.5%;
  background: #fff;
  font-family: sans-serif;
  line-height: 22px;
}
```

margin -

border -

padding -

400 × 2579.970 - - -

Blog Tool, Publishing Platfo ×

https://wordpress.org

Apps Bookmarks Framerate

Responsive ▾ 400 × 1113 49% ▾

WORDPRESS.ORG

Ready to get started? Download WordPress

Themes Twenty Thirteen Twenty Eleven Twenty Fourteen

WordPress is web software you can use to create a beautiful website, blog, or app. We like to say that WordPress is both free and priceless at the same time.

The core software is built by hundreds of community volunteers, and when you're ready for more there are thousands of plugins and themes available to transform your site into almost anything you can imagine. Over 60 million people have chosen WordPress to power the place on the web they call "home" — we'd love you to join the family.

Ready to get started? Download WordPress 4.4.2

WordPress is also available in English (UK) (also Cymraeg, Gàidhlig).

WordPress Swag

Elements Console Sources Network ▾

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en">
  <head profile="http://gmpg.org/xfn/11">...</head>
  ...<body id="wordpress-org">
    <div id="wporg-header">...</div>
    <div id="download-mobile">...</div>
      <!-- feed is cached -->
    <div id="home>Welcome" style="padding-bottom: 14px;">...</div>
    <div id="lang-guess-wrap">...</div>
    <div id="home-below">...</div>
    <div id="wporg-footer">...</div>
    <script type="text/javascript">...</script>
    <script type="text/javascript">...</script>
    <script type="text/javascript">_qevents.push( { qacct:"p-18-mFFk41448M" } )</script>
```

html body#wordpress-org

Styles Event Listeners DOM Breakpoints Properties

Filter + .cls ↻

element.style {

}

body { wp4.css?40:16

font-size: 62.5%;

background: #fff;

font-family: sans-serif;

line-height: 22px;

margin -

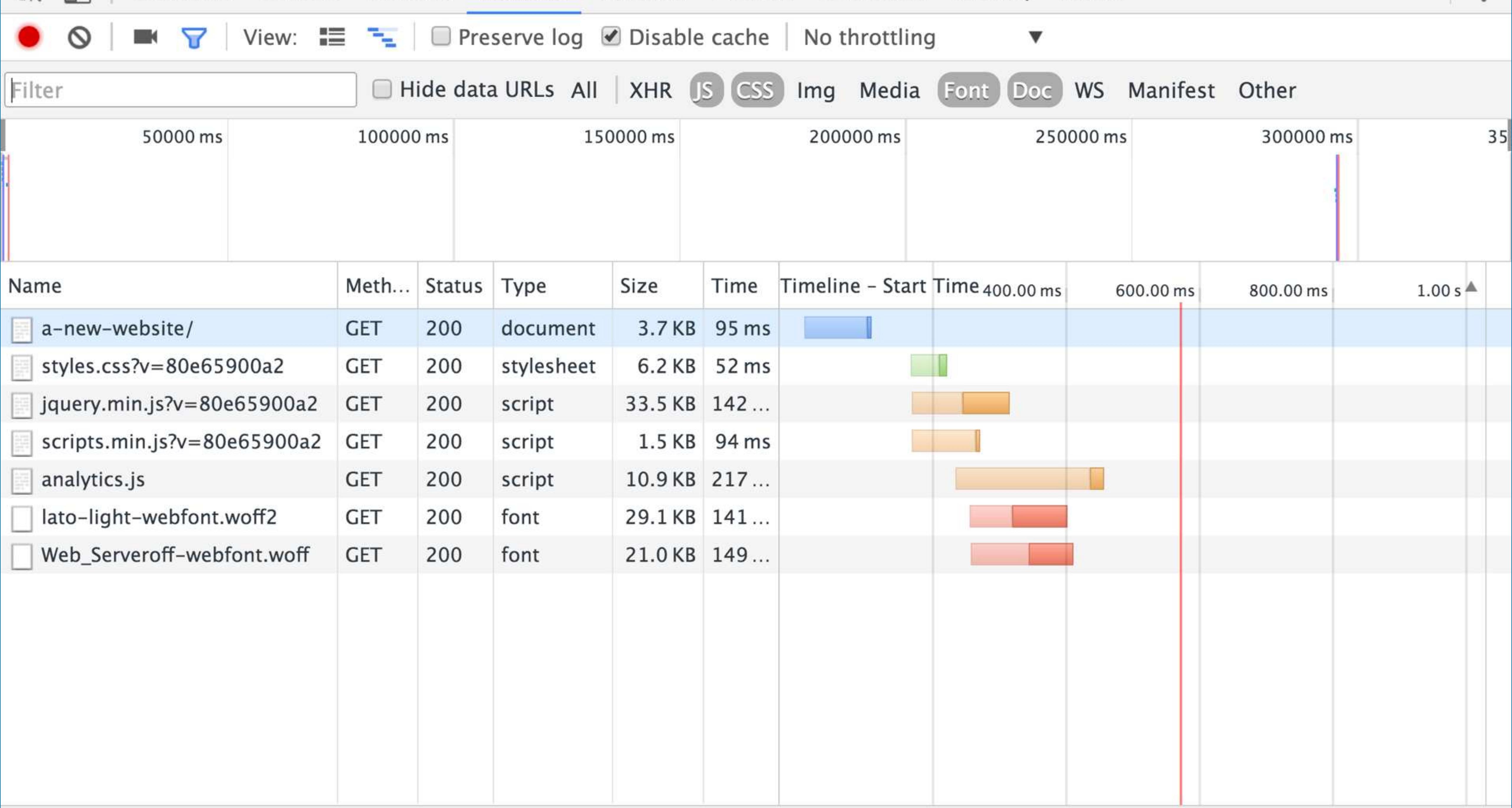
border -

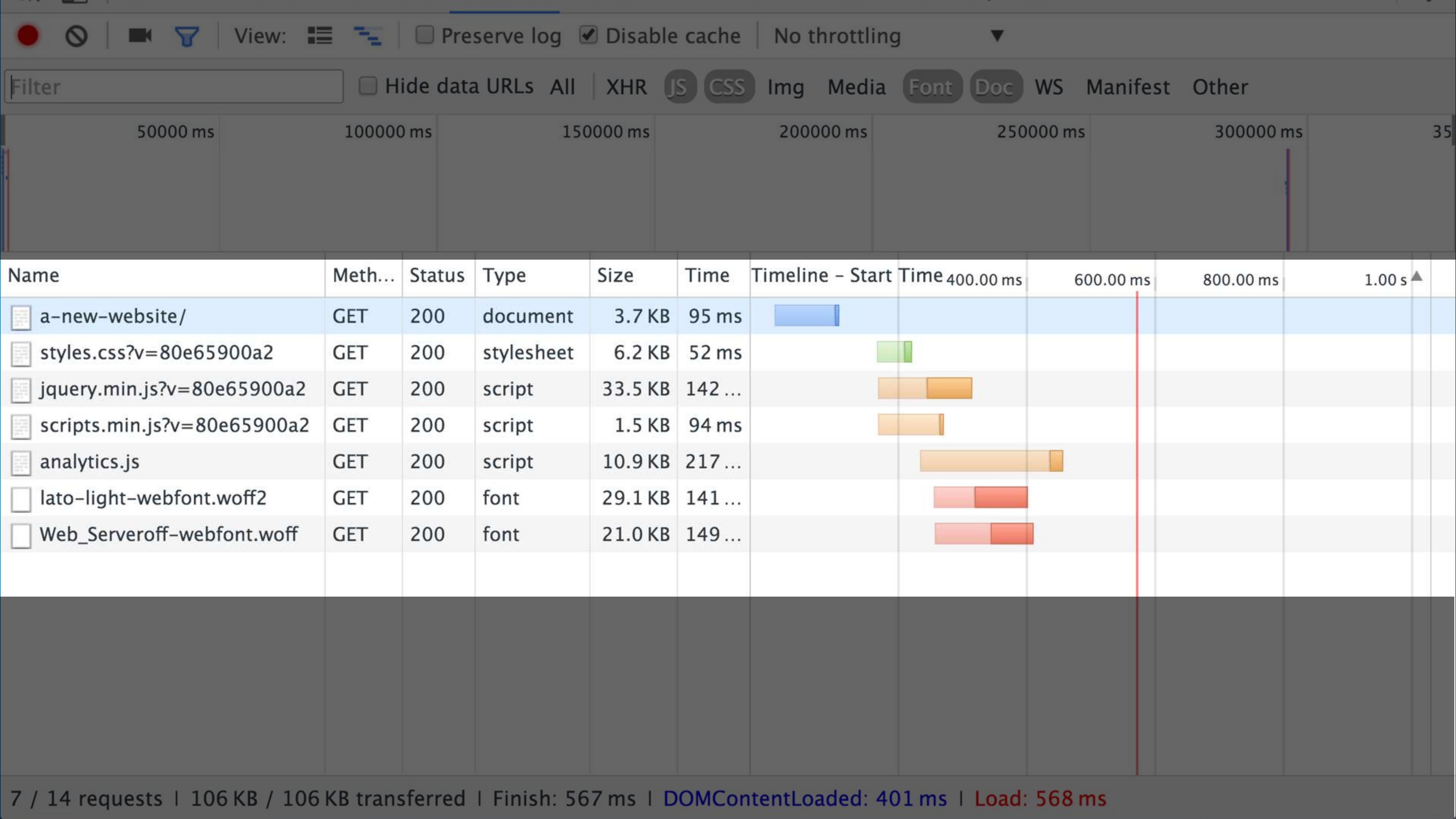
padding -

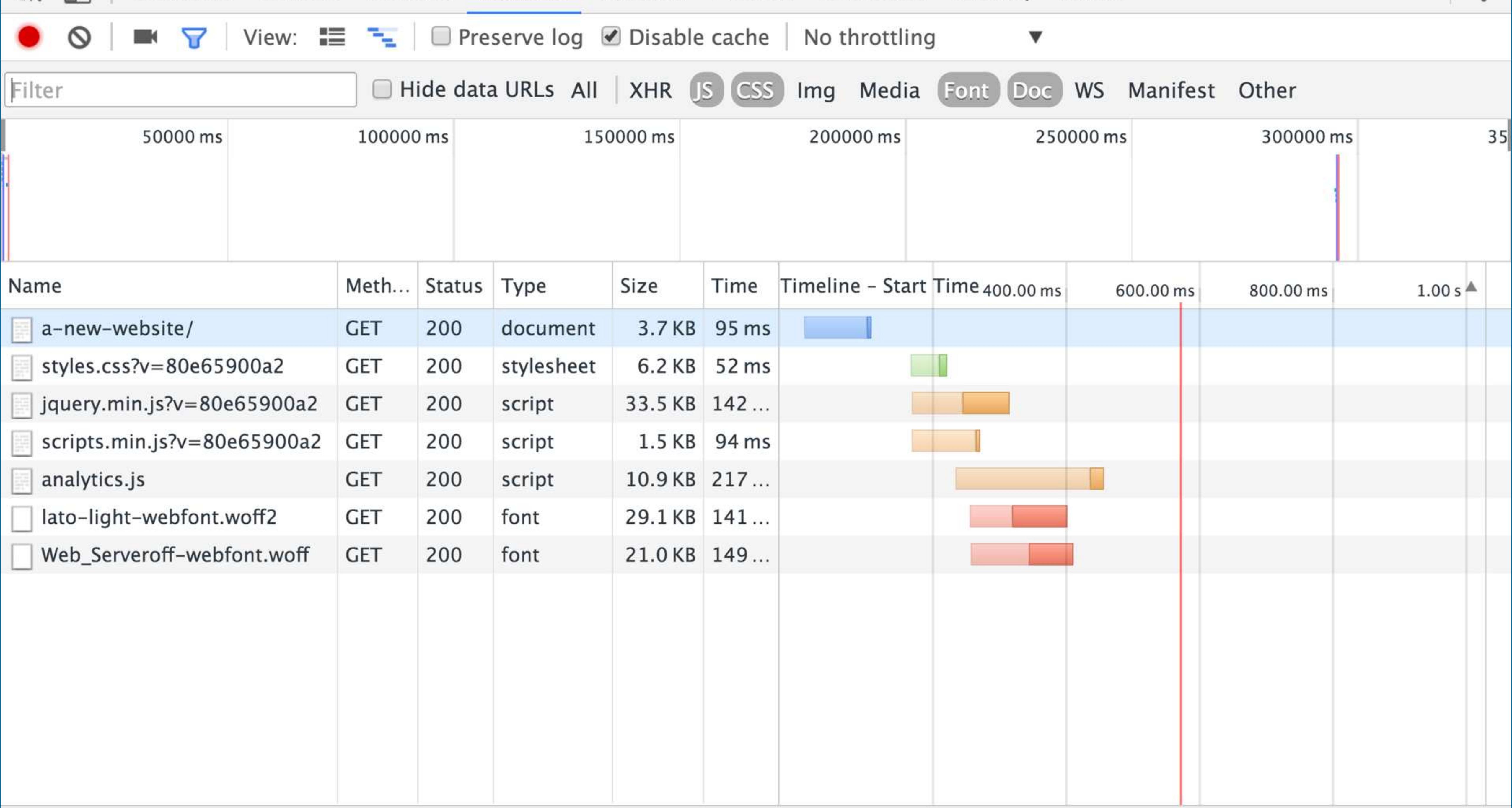
400 × 2579.970 - - -

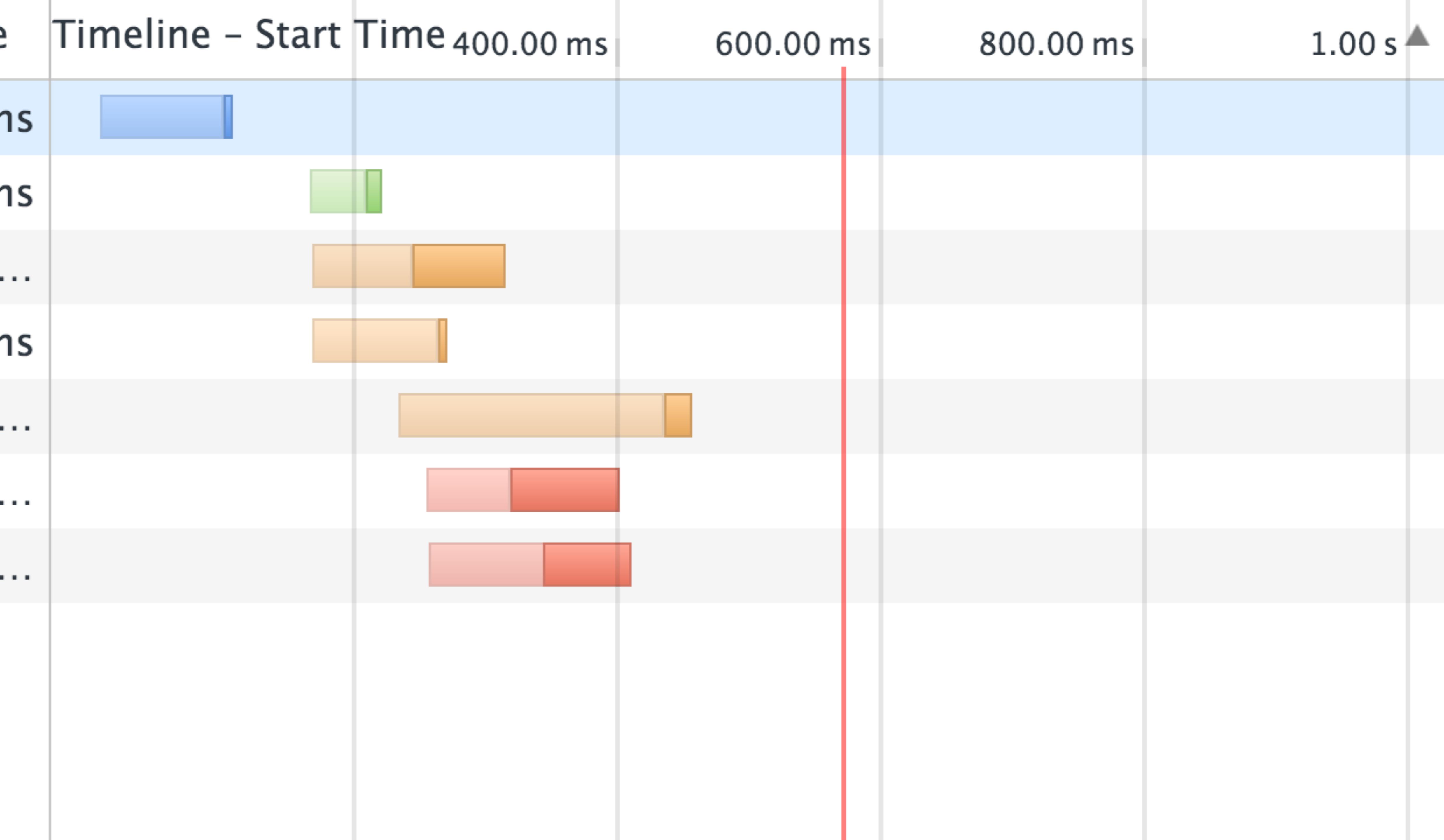
Network Debugging

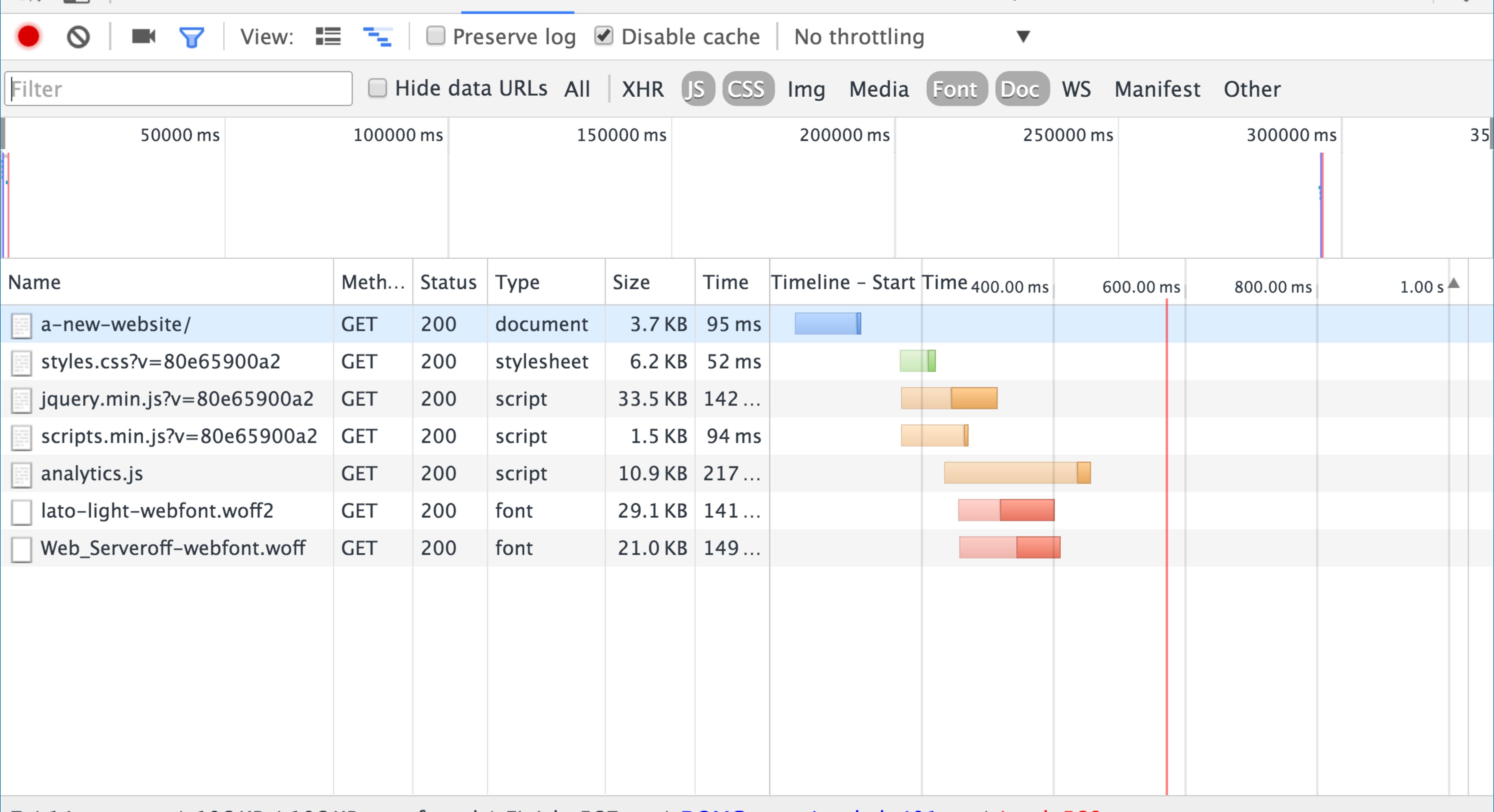
The Network Tab

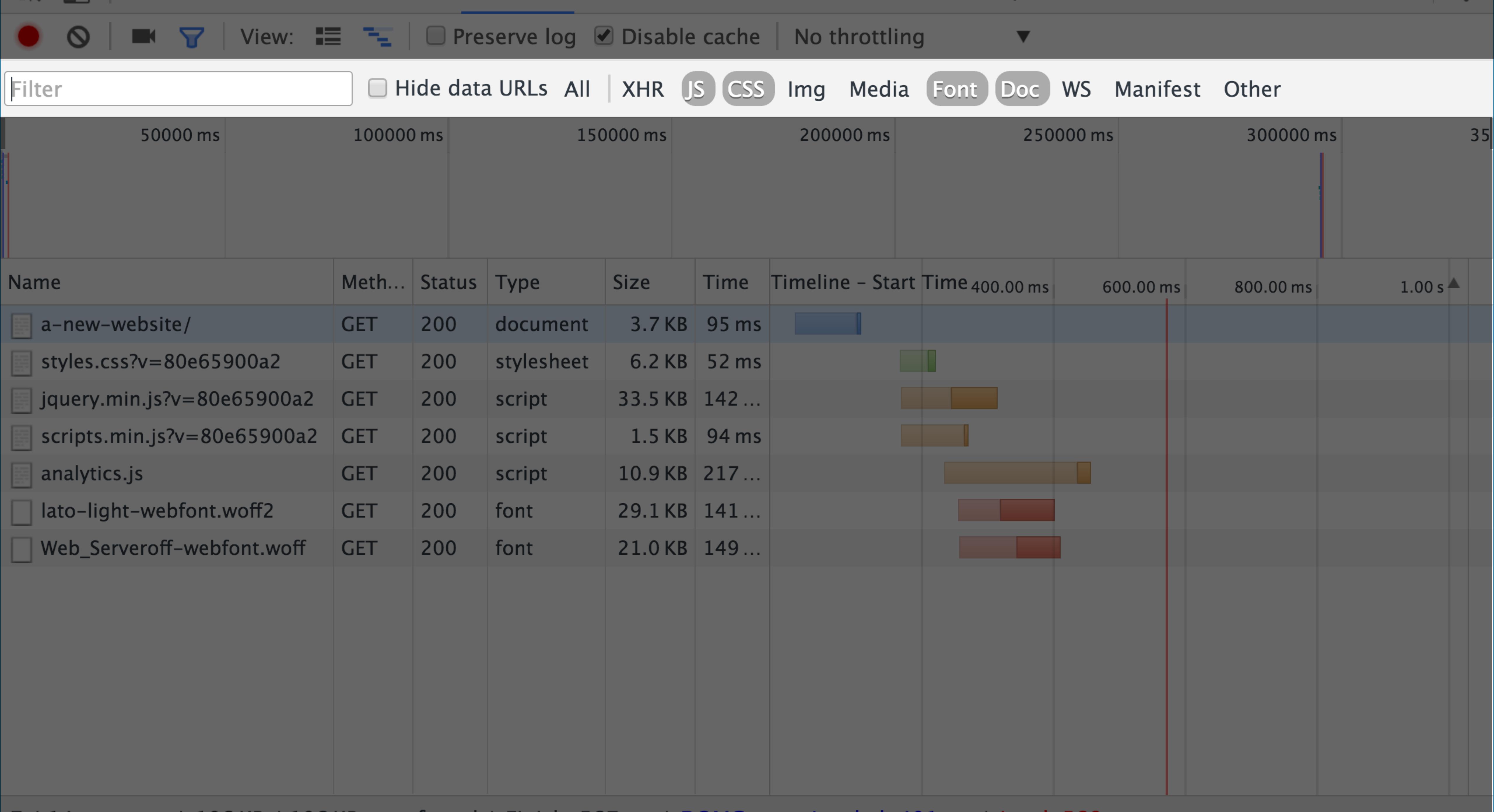


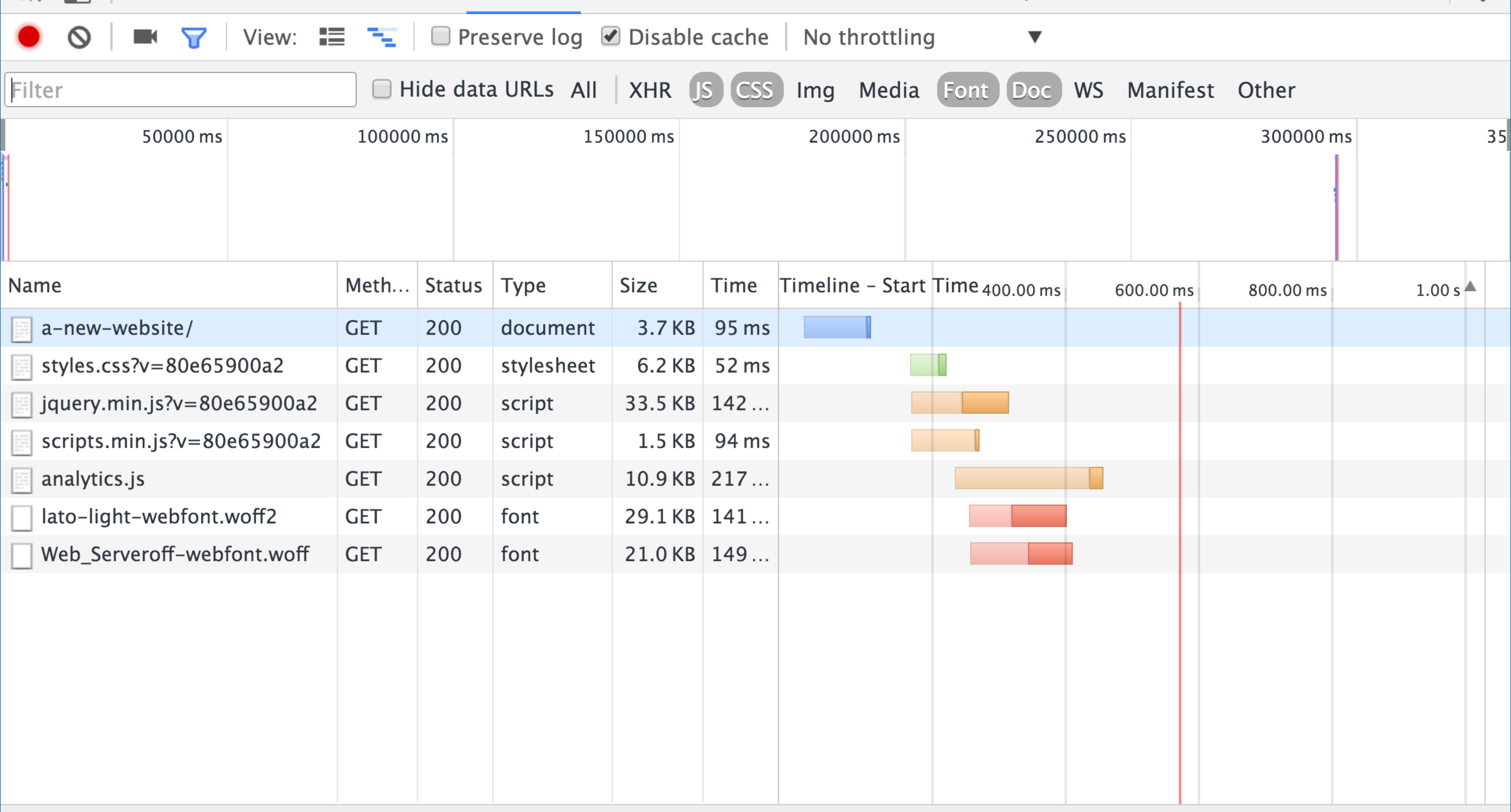


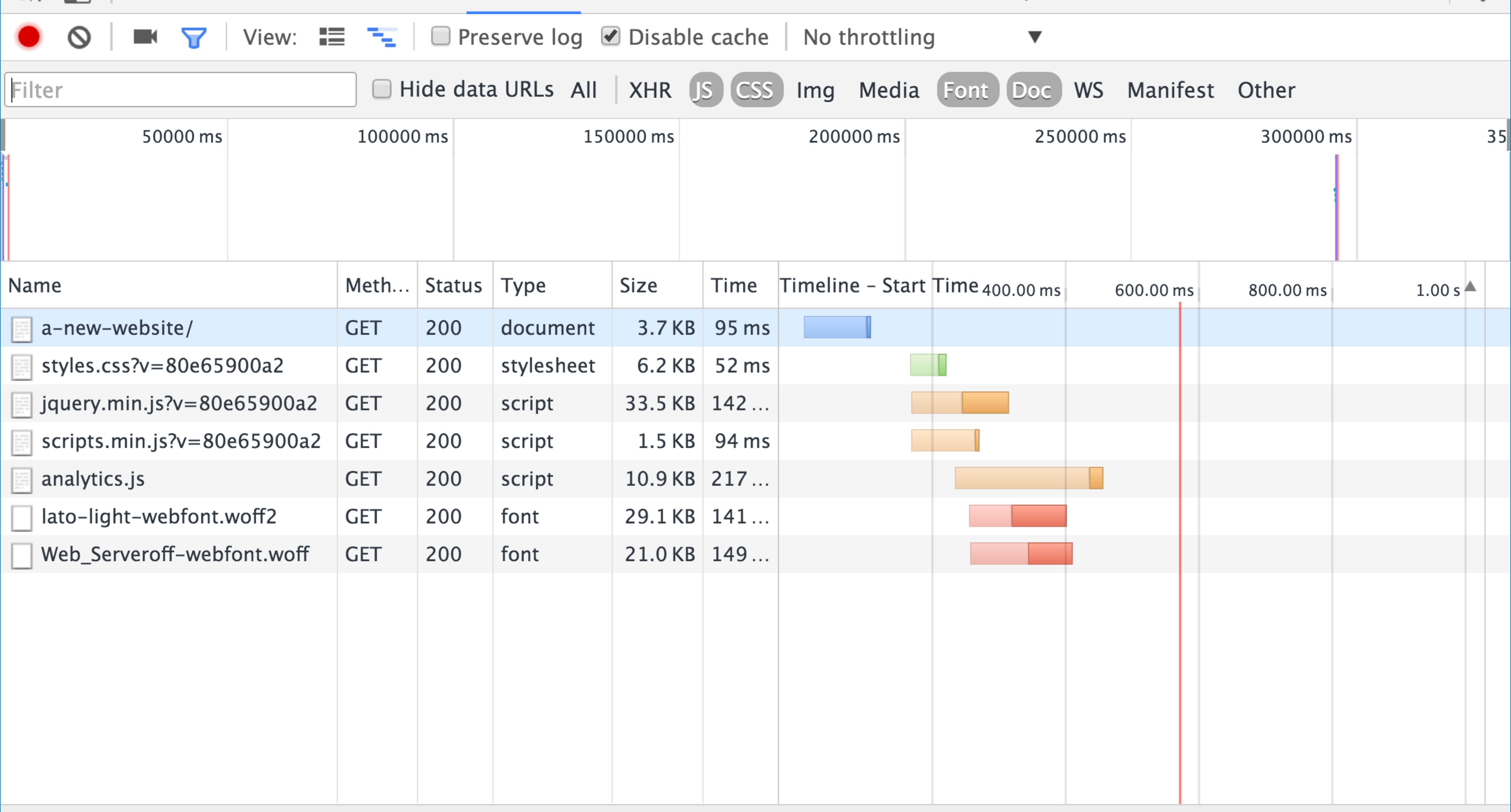












/ 106 KB transferred | Finish: 567 ms | DOMContentLoaded: 401 ms | Load: 568 ms

ources Network Timeline Profiles Resources > ⚠ 2 | ⋮

Preserve log Disable cache | No throttling ▼

to reload and capture filmstrip.

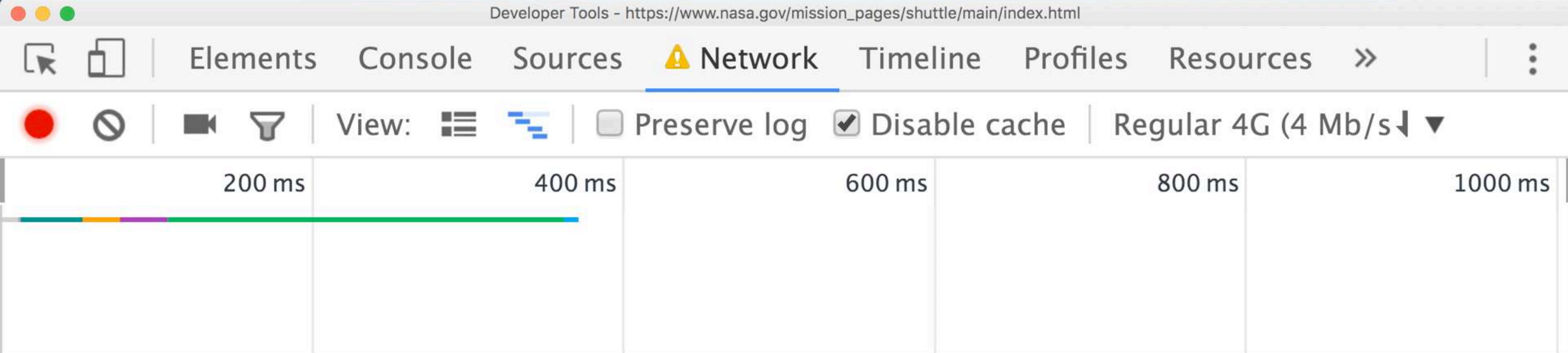
400 ms 600 ms 800 ms 1000 ms

ources Network Timeline Profiles Resources > ⚠ 2 | ⋮

Preserve log Disable cache | No throttling ▼

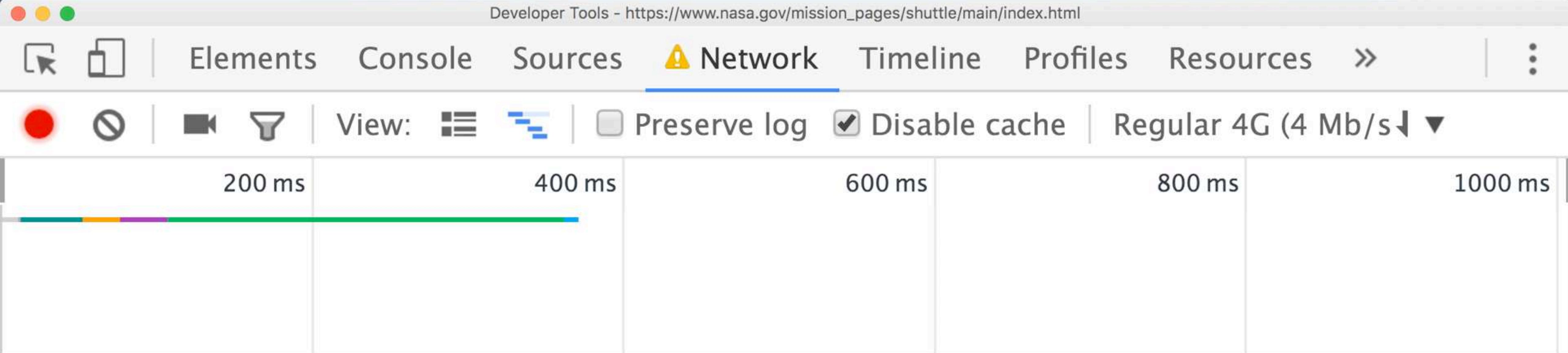
to reload and capture filmstrip.

400 ms 600 ms 800 ms 1000 ms



Recording network activity...

Perform a request or hit ⌘ R to record the reload.



Recording network activity...

Perform a request or hit ⌘ R to record the reload.



Elements

Console

Sources

⚠ Network



View:



Preserve log



200 ms

400 ms

600 ms

800 ms

I

Record (⌘ F) or reload (⌘ R) to display network

[Elements](#)[Console](#)[Sources](#)[Network](#)

View:



Preserve log



200 ms

400 ms

600 ms

800 ms

I

Record (⌘ F) or reload (⌘ R) to display network

Elements Console Sources Network Timeline Profiles Resources » ⚠ 2 | :

View: grid list Preserve log Disable cache No throttling ▾

Name	M...	Stat...	Type	Initiator	Size	Time	Timeline
InviteTri...	G...	200	xhr	Other	534 B	58 ms	
300lo.js...	G...	200	script	addthis_widget.js?	506 B	336 ms	
card-fee...	G...	200	styl...	index.html:47	339 B	58 ms	
InviteTri...	O...	200	xhr	wrapper.js:1	322 B	49 ms	
collect?v...	G...	200	gif	Other	64 B	43 ms	

Elements Console Sources Network Timeline Profiles Resources > ⚠ 2 :

View: grid list Preserve log Disable cache No throttling ▾

Name	M...	Stat...	Type	Initiator	Size	Time	Timeline
InviteTri...	G...	200	xhr	Other	534 B	58 ms	
300lo.js...	G...	200	script	addthis_widget.js?	506 B	336 ms	
card-fee...	G...	200	styl...	index.html:47	339 B	58 ms	█
InviteTri...	O...	200	xhr	wrapper.js:1	322 B	49 ms	
collect?v...	G...	200	gif	Other	64 B	43 ms	█

Script Debugging

The Sources tab

Debugging Scripts:

```
console.log("debug");
```

Debugging Scripts:

debugger;

Elements Network Sources Timeline Profiles Resources Audits Console

Sour... Con... Snip... main.js x

▶ (no domain) 1 \$('.select-navigation select').selectNavigation();
▶ ajax.googleapis.com 2
▶ hello.myfonts.net 3 var scrollMax = \$(document).height() - window.innerHeight,
▶ localhost:9001 4 windowScroll = getWindowScroll(),
5 ticking = false;
▶ css 6
▶ js 7 \$(window).on('scroll', function onScroll () {
▶ (index) 8 windowScroll = getWindowScroll();
9 updateFrame(windowScroll);
10});
11 /*function updateFrame(windowScroll) {
12 animateCloud(windowScroll);
13 }
14
15 function getWindowScroll() {
16 return (1 / (\$(document).height() - window.innerHeight)) * w
17 */
18 */
19
20 function updateFrame(windowScroll) {
21 debugger;
22 if (!ticking) {
23 window.requestAnimationFrame(function animateFrame () {
24 ticking = false;
25 animateCloud(getWindowScroll());
26 });
27 }
28 ticking = true;
29 }
30
31 function getWindowScroll () {
32 return 1 / (scrollMax * window.pageYOffset);
33 }
34
{} Line 21, Column 2

▶ Watch Expressions + C
window.pageYOffset: 269
\$('h1').first().text(): "Katie F..."

▶ Call Stack □ Async
2 stack frames are hidden (black-boxed). Show
updateFrame main.js:21
onScroll main.js:9

▶ Scope Variables

▶ Local
▶ this: Window
windowScroll: 0.0000188991973...

▶ Global Window
Infinity: Infinity
\$: function (e,t){return new b...
AnalyserNode: function Analyse...
ApplicationCache: function App...
ApplicationCacheErrorEvent: fu...
Array: function Array() { [nat...
ArrayBuffer: function ArrayBuffer...
Attr: function Attr() { [nativ...
Audio: function HTMLAudioEleme...
AudioBuffer: function AudioBuf...
AudioBufferSourceNode: functio...
AudioContext: function AudioCo...
AudioDestinationNode: functio...
AudioListener: function AudioL...

```
height() - window.innerHeight)) * w  
roll) {  
  
onFrame(function animateFrame () {  
    windowScroll);  
  
    window.pageYOffset);
```

▼ Scope Variables

▼ Local

- **this**: Window
- **windowScroll**: 0.0000188991973...

▼ Global

- **Infinity**: Infinity
- **\$**: function (e,t){return new b...
- **AnalyserNode**: function Analyse...
- **ApplicationCache**: function App...
- **ApplicationCacheErrorEvent**: fu...
- **Array**: function Array() { [nat...
- **ArrayBuffer**: function ArrayBuffer...
- **Attr**: function Attr() { [nativ...
- **Audio**: function HTMLAudioEleme...
- **AudioBuffer**: function AudioBuf...
- **AudioBufferSourceNode**: functio...
- **AudioContext**: function AudioCo...
- **AudioDestinationNode**: function...

```
    ).selectNavigation();

    eight() - window.innerHeight,
    scroll(),
    ;

on onScroll () {
    scroll();
    ;
}

scroll) {
};

height() - window.innerHeight)) * w
```

The screenshot shows a browser developer tools debugger interface. At the top, there are standard navigation icons: back, forward, search, and others. Below the toolbar, the "Watch Expressions" panel is open, displaying two entries: `window.pageYOffset: 269` and `($('h1').first().text(): "Katie F...`. To the right of the watch expressions are a "+" button and a "C" button. Below the watch expressions is the "Call Stack" panel, which contains the message "2 stack frames are hidden (black-boxed). [Show](#)". A checkbox labeled "Async" is located to the right of the call stack title. The call stack itself lists two frames: "updateFrame" at main.js:21 and "onScroll" at main.js:9. Below the call stack is the "Scope Variables" panel, which is currently collapsed. Under "Scope Variables" is the "Local" panel, which is also collapsed. Inside the "Local" panel, there are two entries: `this: Window` and `windowScroll: 0.00000188991973...`.

Watch Expressions

Call Stack

2 stack frames are hidden (black-boxed). [Show](#)

updateFrame main.js:21

onScroll main.js:9

Scope Variables

Local

this: Window

windowScroll: 0.00000188991973...

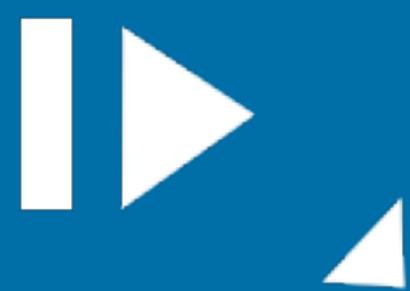


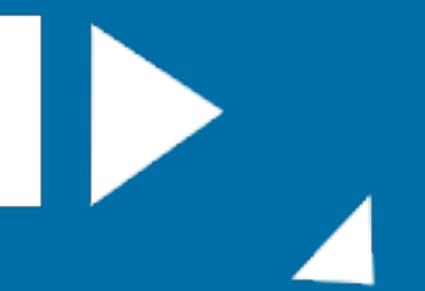
```
).selectNavigation();  
  
height() - window.innerHeight,  
scroll(),  
  
on onScroll () {  
    scroll();  
}  
  
Scroll) {  
};
```

The screenshot shows the DevTools Call Stack panel. At the top, there are several control buttons: a play/pause button, a refresh button, and other navigation controls. Below the controls, there are two sections: "Watch Expressions" and "Call Stack". The "Call Stack" section is expanded, showing the following stack frames:

- updateFrame (main.js:21)
- onScroll (main.js:9)

A message in the middle of the stack says: "2 stack frames are hidden (black-boxed). [Show](#)".

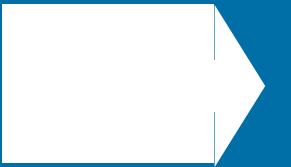




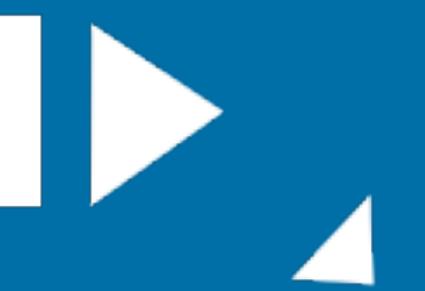
Resume script execution

```
this.get('model').save(options);
```

```
this.save = function (options) {  
    this.status = this.get('willPublish')  
    return get(this, keyName);  
}
```



{



Resume script execution

▶ `this.get('model').save(options);`

```
this.save = function (options) {  
    this.status = this.get('willPublish')  
    return get(this, keyName);  
}
```

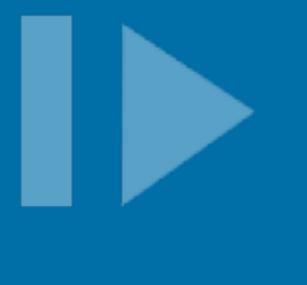


Resume script execution

▶ `this.get('model').save(options);`

```
this.save = function (options) {  
    this.status = this.get('willPublish')  
    return get(this, keyName);
```

`this.status == 'published';`



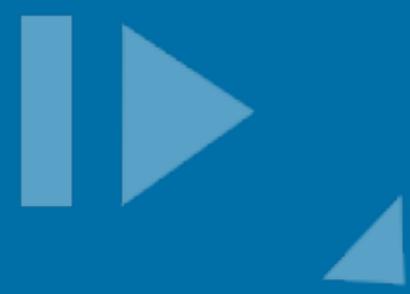
Step over next function

```
markdown = model.get('markdown');  
title = model.get('title');
```



Step over next function

```
markdown = model.get('markdown');  
title = model.get('title');
```



Step over next function

```
markdown = model.get('markdown');  
title = model.get('title');
```



Step into next function

```
this.get('model').save(options);
```

```
get: function(keyName) {  
    return get(this, keyName);  
},
```



Step into next function

```
this.get('model').save(options);
```

```
get: function(keyName) {  
    return get(this, keyName);  
},
```



Step into next function

```
this.get('model').save(options);
```

```
get: function(keyName) {  
    return get(this, keyName);  
},
```



Step into next function

```
this.get('model').save(options);
```

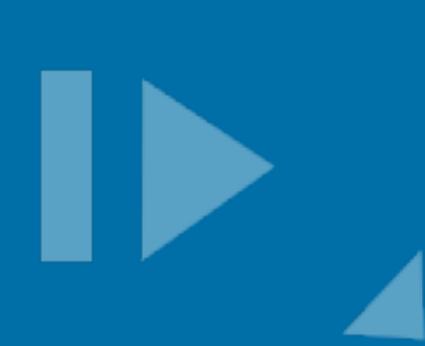
```
get: function(keyName) {  
    return get(this, keyName);  
},
```



Step out of current function

```
this.get('model').save(options);
```

```
this.save = function (options) {  
  this.status = this.get('willPublish')  
  return get(this, keyName);  
}
```



Step out of current function

```
this.get('model').save(options);
```

```
this.save = function (options) {  
    this.status = this.get('willPublish')  
    return get(this, keyName);  
}
```



Step out of current function

```
this.get('model').save(options);
```

```
this.save = function (options) {  
  this.status = this.get('willPublish')  
  return get(this, keyName);  
}
```

Snippets

Sources Content sc...

Snippets

Identity Reg Form

Unbind Paste Handlers

>

<

Identity Reg Form

Unbind Paste Handlers

```
1 var USERNAME = "katie.fenn";
2 var NAME = "Katie";
3 var SURNAME = "Fenn";
4 var PASSWORD = "letmein1";
5 var DOB = "1970-01-01";
6 var GENDER = "F";
7 var EMAIL = "katie.fenn@googlemail.com";
8 var DEPOSIT = "1000";
9 var COUNTRY = "IT";      // Italy
10 var COUNTY = "036";     // Modena
11 var CITY = "0536";      // Maranello
12 var PROVINCE = "MO";    // Modena
13 var PET = "bob";
14 var TAXCODE = "FENKAT84C58E904X";
15
16 var event = document.createEvent("HTMLEvents");
17 event.initEvent("change", false, true);
18
19 // Username
20 document.querySelector('#field-username').value = USERNAME;
21 document.querySelector('#field-username').dispatchEvent(event);
22
23 // Tax code
```

Workspaces

Should disabling JavaScript be an option in the UI?

In its latest release of its web browser, Firefox 23, [Mozilla has removed the option to disable JavaScript](#) from the user interface. It's still a configurable option that can be changed in the browser's about:config interface, so professional and enthusiast users shouldn't have a problem.

A debate is unfolding at Ars Technica regarding this decision, and the wider

The screenshot shows the Mozilla Firefox Developer Tools interface. The top navigation bar includes icons for search, elements, network, sources, timeline, profiles, resources, audits, and console. The sources tab is active, showing a file tree on the left and the content of the file _colours.scss on the right. The file tree shows a local host entry, an assets folder containing css and dist subfolders, a public folder, an index file, and a www.google-analytics entry. The _colours.scss file contains the following SCSS code:

```
$white: #eeeeee;  
$blue-white: #dddfdf;  
  
$heading-colour: $white;  
$body-colour: $blue-white;
```

The right side of the interface features a toolbar with various icons for file operations and a sidebar with sections for Watch, Call Stack, Scope, Breakpoints, DOM Breakpoints, and XHR Breakpoints. The Scope section indicates "Not Paused". The Breakpoints section indicates "No Breakpoints". The DOM Breakpoints and XHR Breakpoints sections are empty.

{ } Line 5, Column 27

Should disabling JavaScript be an option in the UI?

In its latest release of its web browser, Firefox 23, [Mozilla has removed the option to disable JavaScript](#) from the user interface. It's still a configurable option that can be changed in the browser's about:config interface, so professional and enthusiast users shouldn't have a problem.

A debate is unfolding at Ars Technica regarding this decision, and the wider

The screenshot shows the Mozilla Firefox developer tools interface. The top navigation bar includes icons for search, elements, network, sources, timeline, profiles, resources, audits, and console. The "Sources" tab is currently selected. Below the tabs, the left sidebar lists file paths: "(no domain)", "127.0.0.1:2368", "assets/css", "assets/dist", "public", and "(index)". The main content area displays the code for "_colours.scss". The code is as follows:

```
1 $white: #eeeeee;
2 $blue-white: #ffffdd;
3
4 $heading-colour: $white;
5 $body-colour: $blue-white;
```

The right sidebar contains several sections: "Watch" (with a plus icon), "Call Stack" (with an "Async" checkbox), "Scope" (marked as "Not Paused"), "Breakpoints" (marked as "No Breakpoints"), "DOM Breakpoints", and "XHR Breakpoints". At the bottom of the code editor, it says "Line 5, Column 27".

Quick-Open and Pretty-Printing

Sources Content sc... Snippets  

▼  www.katiefenn.co.uk

- ▶  assets
- ▶  public

 (index)

- ▶  (no domain)
- ▶  www.google-analytics.com
- ▶  assets

Hit Cmd+P to open a file

Sources Content sc... Snippets  

▼  www.katiefenn.co.uk

- ▶  assets
- ▶  public

 (index)

- ▶  (no domain)
- ▶  www.google-analytics.com
- ▶  assets

Hit Cmd+P to open a file

Find out more!

DevTools Blackboxing

DevTools Workspaces

Performance Profiling

1. Timeline

Katie Jenn

Web developer

Co-organiser: First Play Sheffield

Video game enthusiast

If you liked this...

Bunny

Coding Horror

Comlicity

Daring Fireball

James Allen on F1

Signal vs. Noise

The Space Shuttle

Waffle

Should disabling JavaScript be an option in the UI?

Posted 2013-08-11 10:54:00 +0100

In its latest release of its web browser, Firefox 23, Mozilla has [removed the option to disable JavaScript](#) from the user interface. It's still a configurable option that can be changed in the browser's about:config interface, so professional and enthusiast users shouldn't have a problem.

A [debate is unfolding at Ars Technica](#) regarding this decision, and the wider merits of JavaScript on the web.

Some are claiming JavaScript is unnecessary, that "pure HTML with CSS is sufficient for the functional needs of about 95% of the web". Others support the use of JavaScript as its capabilities can be used to design more sophisticated interfaces and deliver an effective user experience. One person even slams web developers' use of JavaScript as a "divine right to execute scripts in my browser" that should be resisted.

I'm sure that 95% of the internet doesn't *need* JavaScript. After all, a significant proportion of the internet is porn, and that certainly can't be made better with JavaScript. I'm also a huge proponent of the power of the web to spread news and empower communities; custom-designed select boxes sure as hell aren't necessary for that.

However, I'd argue that JavaScript has become very important for 95% of the web by traffic: Facebook, Twitter, Google Mail. JavaScript is used on these websites to meaningfully enhance the user experience. Such enhancements are routinely required by people paying for websites, although they are also often paying for the guidance to use JavaScript wisely.

JavaScript *is* and *should be* an integral part of the web, and Mozilla is right to withdraw this option from the consumer user. User stories rarely state legacy devices as the reason for supporting the absence of JavaScript. In fact the last time I was given this requirement it was specified in the context of users disabling JavaScript in otherwise functional browsers. I won't deny that there are use cases for devices that do not support JavaScript, but those use cases are now rare.

Custom-design select boxes are still entirely unnecessary, though.

People Don't Buy Why You Do It They Buy

#DZY

Katie Fenn FIA WEC localhost:9001

Developer Tools - http://localhost:9001/

Elements Network Sources Timeline Profiles Resources Audits Console Causes JS Profiler Memory Paint

200 ms 400 ms 600 ms 800 ms 1000 ms

Should disable UI?

Posted 2013-08-11 10:22 by Katie Fenn

In its latest release of its browser, Mozilla has removed JavaScript from the user's browser's about:config problem.

A debate is unfolding at Mozilla about whether to remove JavaScript from the user's browser's about:config problem.

Some are claiming JavaScript is unnecessary for the functional needs of about:config. Others believe that the capabilities can be used to improve the user experience. One person has suggested that users should be able to execute scripts in my browser's about:config page.

I'm sure that 95% of the websites on the internet is porn, and I'm not sure if I'm a proponent of the power of JavaScript. However, I'm not a fan of the designed select boxes seen on many websites.

However, I'd argue that Facebook, Twitter, Google+, and other social media sites have improved the user experience. Sure, they're not perfect, although they are also not designed to be.

JavaScript is and should be an option from the consumer's perspective, supporting the absence of a feature specified in the context of the user's request. I deny that there are use cases where JavaScript is not needed, and are now rare.

Custom-design select boxes are a good example of a feature that is not needed in most cases.

RECORDS

Summary Layers

#DZY

Katie Fenn FIA WEC localhost:9001

Developer Tools - http://localhost:9001/

Elements Network Sources Timeline Profiles Resources Audits Console Causes JS Profiler Memory Paint

200 ms 400 ms 600 ms 800 ms 1000 ms

Should disable UI?

Posted 2013-08-11 10:22 by Katie Fenn

In its latest release of its browser, Mozilla has removed JavaScript from the user's browser's about:config problem.

A debate is unfolding at Mozilla about whether to remove JavaScript from the user's browser's about:config problem.

Some are claiming JavaScript is unnecessary for most functional needs of about:config. Others believe that capabilities can be used to improve the user experience. One person has suggested that users should be able to execute scripts in my browser's about:config page.

I'm sure that 95% of the websites on the internet is porn, and I'm not sure if I'm a proponent of the power of JavaScript. However, I'm not a fan of the designed select boxes seen on many websites.

However, I'd argue that Facebook, Twitter, Google+, and other social media sites have improved the user experience. Sure, they're not perfect, although they are also not perfect.

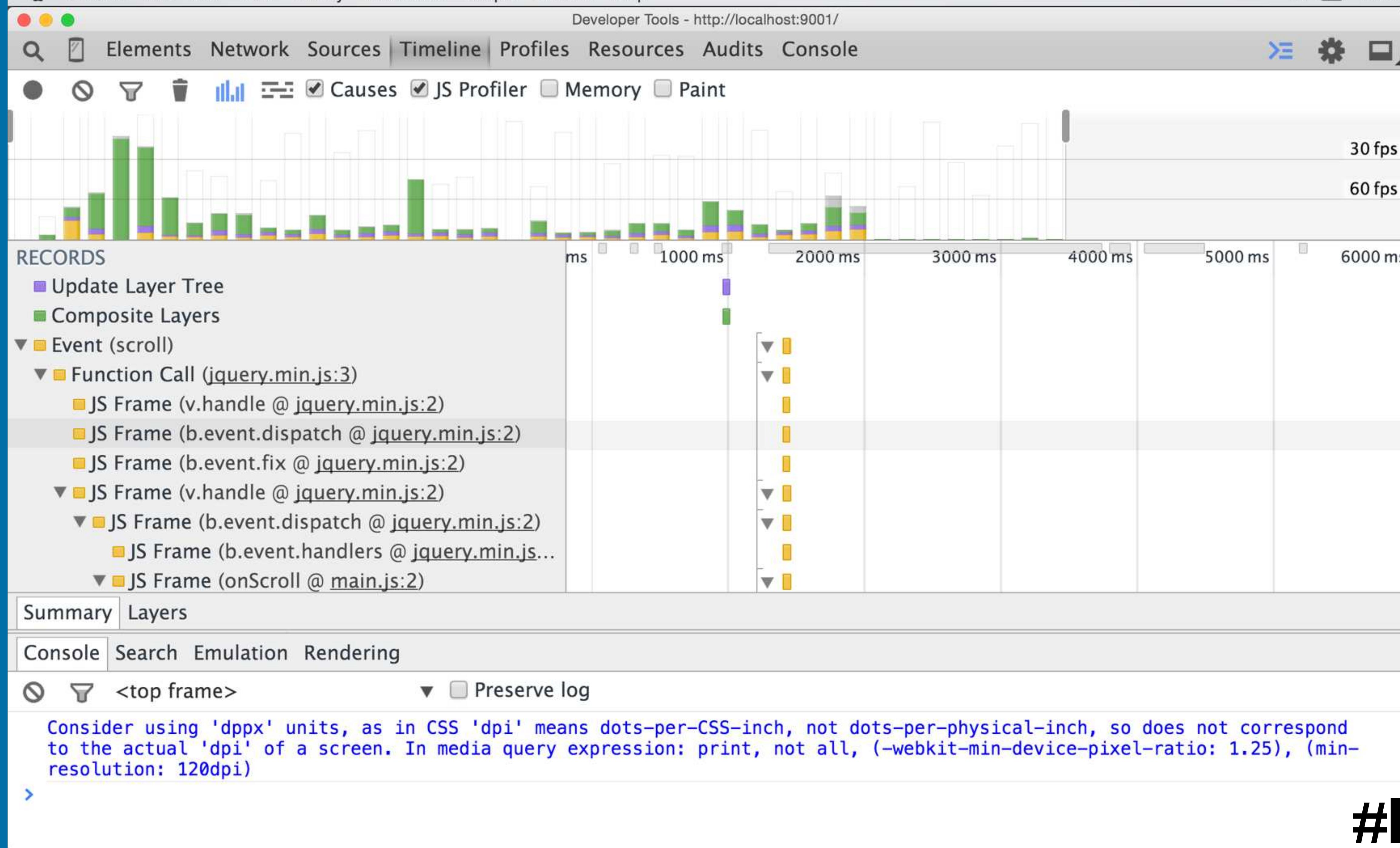
JavaScript is and should be an option from the consumer's perspective. Supporting the absence of JavaScript in the context of the user's choice to deny that there are use cases where it is needed is now rare.

Custom-design select boxes are now rare.

RECORDS

Summary Layers

#DZY





Elements

Network

Sources

Timeline

Profiles

Resources

Audits

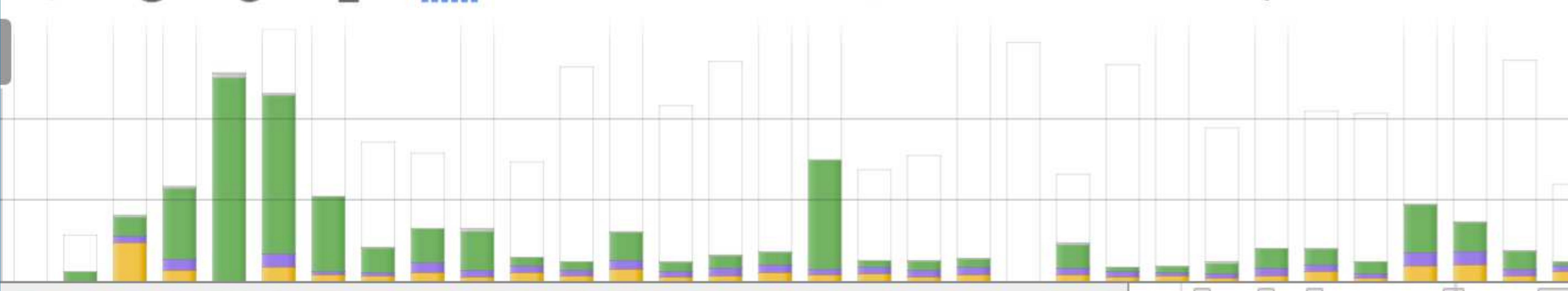


Causes

JS Profiler

Memory

Paint



RECORDS

 Update Layer Tree Composite Layers Event (scroll) Function Call ([jquery.min.js:3](#)) JS Frame ([v.handle @ jquery.min.js:2](#))

ms

1000 ms

#DZY



Elements

Network

Sources

Timeline

Profiles

Resources

Audits

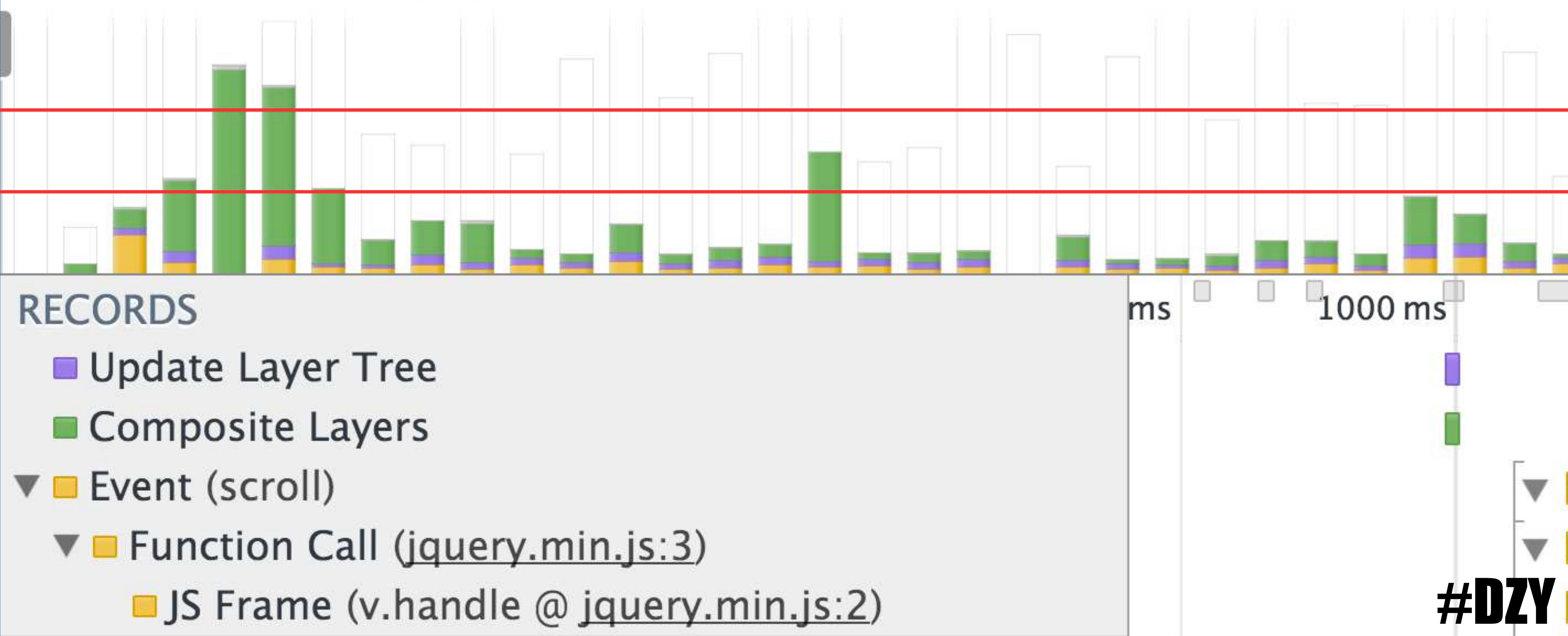


Causes

JS Profiler

Memory

Paint





Elements

Network

Sources

Timeline

Profiles

Resources

Audits



Causes



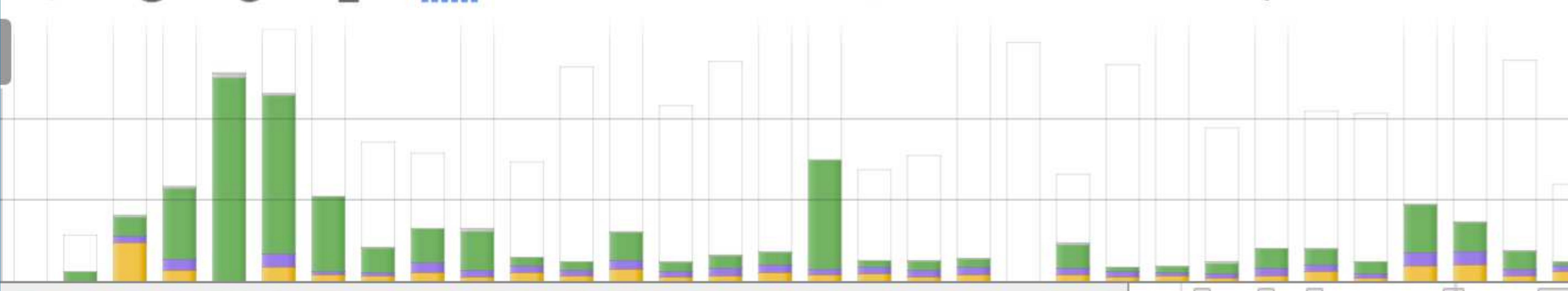
JS Profiler



Memory



Paint



RECORDS

 Update Layer Tree Composite Layers Event (scroll) Function Call ([jquery.min.js:3](#)) JS Frame ([v.handle @ jquery.min.js:2](#))

#DZY

RECORDS

ms

1000 ms

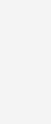
Update Layer Tree



Composite Layers



Event (scroll)

Function Call ([jquery.min.js:3](#))JS Frame ([v.handle @ jquery.min.js:2](#))JS Frame ([b.event.dispatch @ jquery.min.js:2](#))JS Frame ([b.event.fix @ jquery.min.js:2](#))JS Frame ([v.handle @ jquery.min.js:2](#))JS Frame ([b.event.dispatch @ jquery.min.js:2](#))JS Frame ([b.event.handlers @ jquery.min.js...](#))JS Frame ([onScroll @ main.js:2](#))

Summary

Layers

Console

Search Emulation Rendering

#DZY



<top frame>



Preserve log

Katie Fenn FIA WEC

localhost:9001

Should disable UI?

Posted 2013-08-11 10:15

In its latest release of its browser, Mozilla has disabled [JavaScript](#) from the user's browser's about:config interface. This is a problem.

A [debate is unfolding at Mozilla's bugzilla](#) regarding the future of JavaScript on the web.

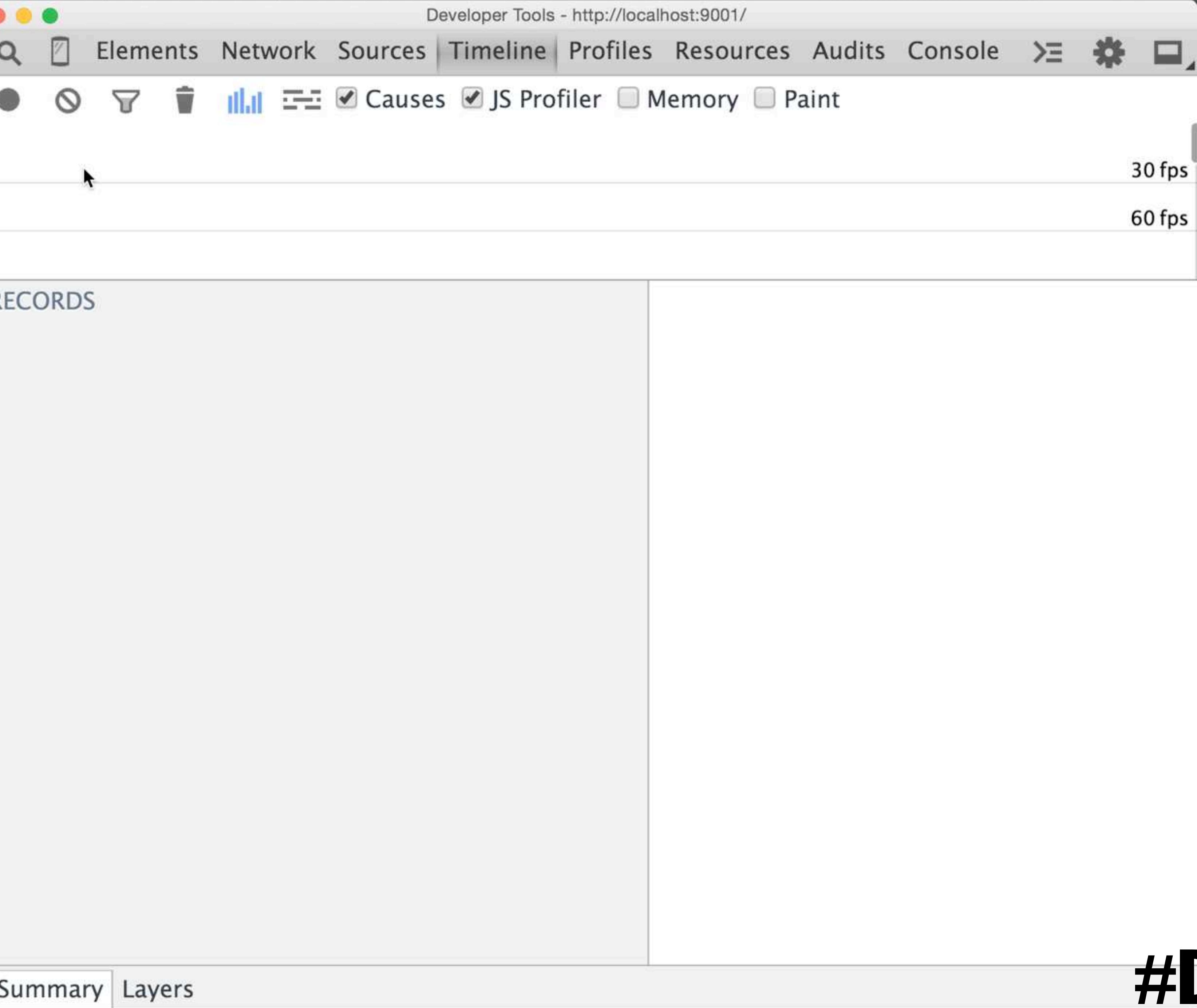
Some are claiming JavaScript is unnecessary for basic functional needs of about:config. Others believe that the capabilities can be used to enhance user experience. One person even suggested that users should be able to execute scripts in my browser's memory.

I'm sure that 95% of the content on the internet is porn, and I'm not sure that a proponent of the power of JavaScript would be in favor of that. However, I'd argue that custom-designed select boxes should be supported.

However, I'd argue that Facebook, Twitter, Google+, and other social media sites have ruined the user experience. Sure, they're useful, but although they are also cool, they're not what the web was designed for.

JavaScript *is* and *should be* supported. It's a valuable option from the consumer's perspective, supporting the absence of functionality specified in the context of the page. I deny that there are use cases where JavaScript is not needed. They are now rare.

Custom-design select boxes are cool, but they're not what the web was designed for.



Katie Fenn FIA WEC

localhost:9001

Should disable UI?

Posted 2013-08-11 10:15

In its latest release of its browser, Mozilla has disabled [JavaScript](#) from the user's browser's about:config interface. This is a problem.

A [debate is unfolding at Mozilla's bugzilla](#) over whether to disable JavaScript on the web.

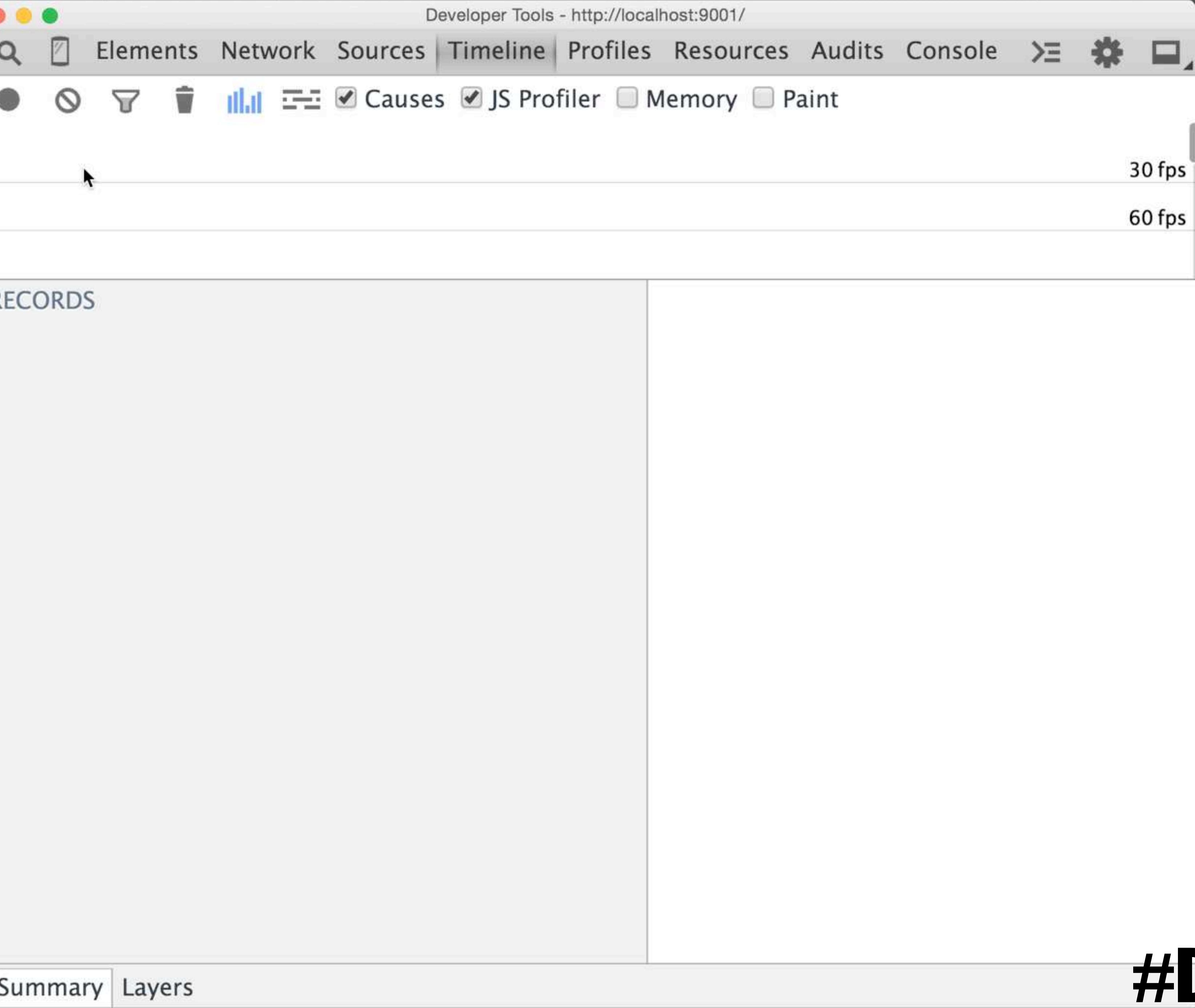
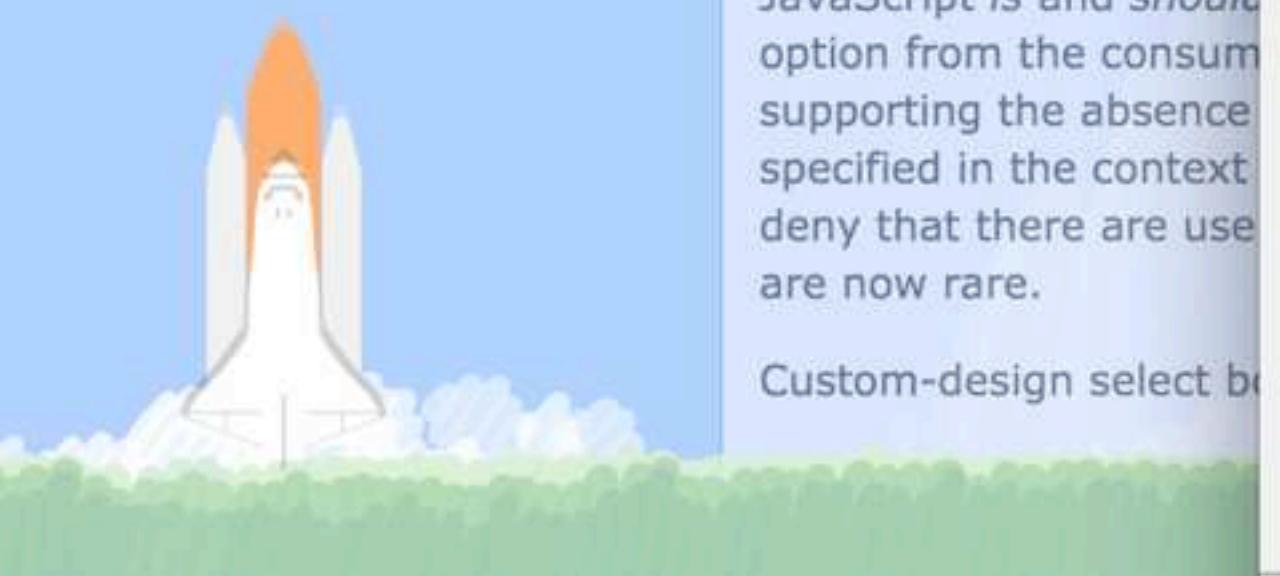
Some are claiming JavaScript is useful for functional needs of about:config. Others say that capabilities can be used to compromise user experience. One person claims that "I can't even execute scripts in my browser."

I'm sure that 95% of the content on the internet is porn, and I'm not sure that a proponent of the power of JavaScript would design select boxes so poorly.

However, I'd argue that Facebook, Twitter, Google+, and other sites improve the user experience. Sure, they're not perfect, although they are also designed to be.

JavaScript *is* and *should* be disabled by default as an option from the consumer's perspective, supporting the absence of security specified in the context of the browser. I deny that there are use cases where JavaScript is needed; they are now rare.

Custom-design select boxes are a good example.



Find out more!

[DevTools Paint Profiler and Frame Viewer](#)

Performance Profiling

2. CPU Profiler



Developer Tools - http://localhost:9001/

Elements Network Sources Timeline Profiles Resources Audits Console

Profiles

Select profiling type

Collect JavaScript CPU Profile

CPU profiles show where the execution time is spent in your page's JavaScript functions.

Take Heap Snapshot

Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.

Record Heap Allocations

Record JavaScript object allocations over time. Use this profile type to isolate memory leaks.

Start Load



Developer Tools - http://localhost:9001/

Elements Network Sources Timeline Profiles Resources Audits Console

Profiles

Select profiling type

Collect JavaScript CPU Profile

CPU profiles show where the execution time is spent in your page's JavaScript functions.

Take Heap Snapshot

Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.

Record Heap Allocations

Record JavaScript object allocations over time. Use this profile type to isolate memory leaks.

Start Load

#DZY

Elements Network Sources Timeline **Profiles** Resources Audits Console

Heavy (Bottom Up) ▾ eye x refresh

Profiles	Self	Total	Function
CPU PROFILES	5883.4 ms 93.34%	5883.4 ms 93.34%	(idle)
	383.3 ms 6.08%	383.3 ms 6.08%	(program)
	2.1 ms 0.03%	36.4 ms 0.58%	v.handle
	2.1 ms 0.03%	34.3 ms 0.54%	▶ b.event.dispatch
	0 ms 0%	27.8 ms 0.44%	▶ onScroll
	0 ms 0%	23.6 ms 0.37%	▶ updateFrame
	0 ms 0%	23.6 ms 0.37%	▶ animateCloud
	1.1 ms 0.02%	23.6 ms 0.37%	▶ b.extend.access
	1.1 ms 0.02%	20.3 ms 0.32%	▶ b.fn.extend.css
	1.1 ms 0.02%	18.2 ms 0.29%	▶ (anonymous function)
	15.0 ms 0.24%	17.1 ms 0.27%	▶⚠ b.extend.style
	0 ms 0%	4.3 ms 0.07%	▶ b.fn.(anonymous function)
	2.1 ms 0.03%	4.3 ms 0.07%	▶ (anonymous function)
	0 ms 0%	4.3 ms 0.07%	▶ getWindowScroll
	1.1 ms 0.02%	3.2 ms 0.05%	▶ b
	2.1 ms 0.03%	2.1 ms 0.03%	▶ b.event.fix
	1.1 ms 0.02%	2.1 ms 0.03%	▶ P
	0 ms 0%	2.1 ms 0.03%	▶ b.fn.b.init
	0 ms 0%	2.1 ms 0.03%	▶ b.fn.extend.find
	0 ms 0%	2.1 ms 0.03%	▶ b.extend._data
	2.1 ms 0.03%	2.1 ms 0.03%	▶⚠ st
	1.1 ms 0.02%	1.1 ms 0.02%	▶ get scrollHeight
	1.1 ms 0.02%	1.1 ms 0.02%	▶ b.extend.acceptData
	1.1 ms 0.02%	1.1 ms 0.02%	▶ (anonymous function)
	1.1 ms 0.02%	1.1 ms 0.02%	▶ b.extend.camelCase
	1.1 ms 0.02%	1.1 ms 0.02%	▶ get source_url

Profile 1 Save

#DZY

Elements Network Sources Timeline **Profiles** Resources Audits Console

Heavy (Bottom Up) ▾ Eye X Reload

Self	Total	Function
5883.4 ms 93.34%	5883.4 ms 93.34%	(idle)
383.3 ms 6.08%	383.3 ms 6.08%	(program)
2.1 ms 0.03%	36.4 ms 0.58%	v.handle
2.1 ms 0.03%	34.3 ms 0.54%	▶ b.event.dispatch
0 ms 0%	27.8 ms 0.44%	▶ onScroll
0 ms 0%	23.6 ms 0.37%	▶ updateFrame
0 ms 0%	23.6 ms 0.37%	▶ animateCloud
1.1 ms 0.02%	23.6 ms 0.37%	▶ b.extend.access
1.1 ms 0.02%	20.3 ms 0.32%	▶ b.fn.extend.css
1.1 ms 0.02%	18.2 ms 0.29%	▶ (anonymous function)
15.0 ms 0.24%	17.1 ms 0.27%	▶⚠ b.extend.style
0 ms 0%	4.3 ms 0.07%	▶ b.fn.(anonymous function)
2.1 ms 0.03%	4.3 ms 0.07%	▶ (anonymous function)
0 ms 0%	4.3 ms 0.07%	▶ getWindowScroll
1.1 ms 0.02%	3.2 ms 0.05%	▶ b
2.1 ms 0.03%	2.1 ms 0.03%	▶ b.event.fix
1.1 ms 0.02%	2.1 ms 0.03%	▶ P
0 ms 0%	2.1 ms 0.03%	▶ b.fn.b.init
0 ms 0%	2.1 ms 0.03%	▶ b.fn.extend.find
0 ms 0%	2.1 ms 0.03%	▶ b.extend._data
2.1 ms 0.03%	2.1 ms 0.03%	▶⚠ st
1.1 ms 0.02%	1.1 ms 0.02%	▶ get scrollHeight
1.1 ms 0.02%	1.1 ms 0.02%	▶ b.extend.acceptData
1.1 ms 0.02%	1.1 ms 0.02%	▶ (anonymous function)
1.1 ms 0.02%	1.1 ms 0.02%	▶ b.extend.camelCase
1.1 ms 0.02%	1.1 ms 0.02%	▶ get source_url

Profiles CPU PROFILES **Profile 1 Save**

#DZY

Elements Network Sources Timeline Profiles Resources Audits Console

Heavy (Bottom Up) ▾ eye x refresh

Self	Total	Function
5883.4 ms 93.34%	5883.4 ms 93.34%	(idle)
383.3 ms 6.08%	383.3 ms 6.08%	(program)
2.1 ms 0.03%	36.4 ms 0.58%	v.handle
2.1 ms 0.03%	34.3 ms 0.54%	▶ b.event.dispatch
0 ms 0%	27.8 ms 0.44%	▶ onScroll
0 ms 0%	23.6 ms 0.37%	▶ updateFrame
0 ms 0%	23.6 ms 0.37%	▶ animateCloud
1.1 ms 0.02%	23.6 ms 0.37%	▶ b.extend.access
1.1 ms 0.02%	20.3 ms 0.32%	▶ b.fn.extend.css
1.1 ms 0.02%	18.2 ms 0.29%	▶ (anonymous function)
15.0 ms 0.24%	17.1 ms 0.27%	▶⚠ b.extend.style
0 ms 0%	4.3 ms 0.07%	▶ b.fn.(anonymous function)
2.1 ms 0.03%	4.3 ms 0.07%	▶ (anonymous function)
0 ms 0%	4.3 ms 0.07%	▶ getWindowScroll
1.1 ms 0.02%	3.2 ms 0.05%	▶ b
2.1 ms 0.03%	2.1 ms 0.03%	▶ b.event.fix
1.1 ms 0.02%	2.1 ms 0.03%	▶ P
0 ms 0%	2.1 ms 0.03%	▶ b.fn.b.init
0 ms 0%	2.1 ms 0.03%	▶ b.fn.extend.find
0 ms 0%	2.1 ms 0.03%	▶ b.extend._data
2.1 ms 0.03%	2.1 ms 0.03%	▶⚠ st
1.1 ms 0.02%	1.1 ms 0.02%	▶ get scrollHeight
1.1 ms 0.02%	1.1 ms 0.02%	▶ b.extend.acceptData
1.1 ms 0.02%	1.1 ms 0.02%	▶ (anonymous function)
1.1 ms 0.02%	1.1 ms 0.02%	▶ b.extend.camelCase
1.1 ms 0.02%	1.1 ms 0.02%	▶ get source_url

Profiles CPU PROFILES Profile 1 Save

#DZY

Elements Network Sources Timeline **Profiles** Resources Audits Console

Heavy (Bottom Up) ▾ eye x refresh

Self	Total	Function
5883.4 ms 93.34%	5883.4 ms 93.34%	(idle)
383.3 ms 6.08%	383.3 ms 6.08%	(program)
2.1 ms 0.03%	36.4 ms 0.58%	v.handle
2.1 ms 0.03%	34.3 ms 0.54%	▶ b.event.dispatch
0 ms 0%	27.8 ms 0.44%	▶ onScroll
0 ms 0%	23.6 ms 0.37%	▶ updateFrame
0 ms 0%	23.6 ms 0.37%	▶ animateCloud
1.1 ms 0.02%	23.6 ms 0.37%	▶ b.extend.access
1.1 ms 0.02%	20.3 ms 0.32%	▶ b.fn.extend.css
1.1 ms 0.02%	18.2 ms 0.29%	▶ (anonymous function)
15.0 ms 0.24%	17.1 ms 0.27%	▶⚠ b.extend.style
0 ms 0%	4.3 ms 0.07%	▶ b.fn.(anonymous function)
2.1 ms 0.03%	4.3 ms 0.07%	▶ (anonymous function)
0 ms 0%	4.3 ms 0.07%	▶ getWindowScroll
1.1 ms 0.02%	3.2 ms 0.05%	▶ b
2.1 ms 0.03%	2.1 ms 0.03%	▶ b.event.fix
1.1 ms 0.02%	2.1 ms 0.03%	▶ P
0 ms 0%	2.1 ms 0.03%	▶ b.fn.b.init
0 ms 0%	2.1 ms 0.03%	▶ b.fn.extend.find
0 ms 0%	2.1 ms 0.03%	▶ b.extend._data
2.1 ms 0.03%	2.1 ms 0.03%	▶⚠ st
1.1 ms 0.02%	1.1 ms 0.02%	▶ get scrollHeight
1.1 ms 0.02%	1.1 ms 0.02%	▶ b.extend.acceptData
1.1 ms 0.02%	1.1 ms 0.02%	▶ (anonymous function)
1.1 ms 0.02%	1.1 ms 0.02%	▶ b.extend.camelCase
1.1 ms 0.02%	1.1 ms 0.02%	▶ get source_url

Profiles
CPU PROFILES
Profile 1 Save

#DZY

Total	▼	Function	
5883.4 ms	93.34 %	(idle)	
383.3 ms	6.08 %	(program)	
36.4 ms	0.58 %	v.handle	jquery.min.js:2
34.3 ms	0.54 %	► b.event.dispatch	jquery.min.js:2
27.8 ms	0.44 %	►onScroll	main.js:2
23.6 ms	0.37 %	►updateFrame	main.js:7
23.6 ms	0.37 %	►animateCloud	main.js:15
23.6 ms	0.37 %	► b.extend.access	jquery.min.js:2
20.3 ms	0.32 %	► b.fn.extend.css	jquery.min.js:4
18.2 ms	0.29 %	►(anonymous function)	jquery.min.js:4
17.1 ms	0.27 %	►⚠ b.extend.style	jquery.min.js:4
4.3 ms	0.07 %	► b.fn.(anonymous function)	jquery.min.js:4
4.3 ms	0.07 %	►(anonymous function)	jquery.min.js:4

#DZY

Katie Fenn FIA WEC LIVE

localhost:9001

Should disabling UI?

Posted 2013-08-11 10:54:00 +0100

In its latest release of its web browser, Mozilla has removed the ability to disable JavaScript from the user interface. In the browser's about:config interface, there is no longer a setting to disable JavaScript. This is a problem.

A [debate is unfolding at Ars Technica](#) over whether Mozilla's decision to remove the ability to disable JavaScript on the web is a good idea.

Some are claiming JavaScript is unnecessary for most web pages. They argue that functional needs of about 95% of web pages can be met with CSS and HTML alone. One person even slam-dunked Mozilla's decision by saying "I can't imagine anyone executing scripts in my browser" that don't have a functional purpose.

I'm sure that 95% of the internet is porn, and that certain proponent of the power of the web is referring to the fact that custom-designed select boxes sure as hell aren't functional.

However, I'd argue that JavaScript is useful for many things. Facebook, Twitter, Google Mail. JavaScript can enhance the user experience. Such enhancements, although they are also often paid for by the user, are often well worth it.

JavaScript *is* and *should be* an integral part of the web. Supporting the absence of JavaScript is a bad idea. It's not supported in the context of users disabled JavaScript. We deny that there are use cases for JavaScript that are now rare.

Custom-design select boxes are still a good idea.



Developer Tools - http://localhost:9001/

Elements Network Sources Timeline Profiles Resources Audits >   

Profiles

Select profiling type

Collect JavaScript CPU Profile

CPU profiles show where the execution time is spent in your page's JavaScript functions.

Take Heap Snapshot

Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.

Record Heap Allocations

Record JavaScript object allocations over time. Use this profile type to isolate memory leaks.

Start **Load**

#DZY

Katie Fenn FIA WEC LIVE

localhost:9001

Should disabling UI?

Posted 2013-08-11 10:54:00 +0100

In its latest release of its web browser, Mozilla has removed the ability to disable JavaScript from the user interface. In the browser's about:config interface, there is no longer a setting to disable JavaScript. This is a problem.

A [debate is unfolding at Ars Technica](#) over whether Mozilla's decision to remove the ability to disable JavaScript on the web is a good idea.

Some are claiming JavaScript is unnecessary for most web pages. They argue that functional needs of about 95% of web pages can be met with CSS and HTML alone. One person even slam-dunked Mozilla's decision by saying "I can't imagine anyone executing scripts in my browser" that don't have a functional purpose.

I'm sure that 95% of the internet is porn, and that certain proponent of the power of the web is referring to the fact that custom-designed select boxes sure as hell aren't functional.

However, I'd argue that JavaScript is useful for many things. Facebook, Twitter, Google Mail. JavaScript can enhance the user experience. Such enhancements, although they are also often paid for by the user, are often well worth it.

JavaScript *is* and *should be* an integral part of the web. Supporting the absence of JavaScript is a bad idea. It's not supported in the context of users disabled JavaScript. We deny that there are use cases for JavaScript that are now rare.

Custom-design select boxes are still a good idea.



Developer Tools - http://localhost:9001/

Elements Network Sources Timeline Profiles Resources Audits >   

Profiles

Select profiling type

Collect JavaScript CPU Profile

CPU profiles show where the execution time is spent in your page's JavaScript functions.

Take Heap Snapshot

Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.

Record Heap Allocations

Record JavaScript object allocations over time. Use this profile type to isolate memory leaks.

Start **Load**

#DZY

Elements Network Sources Timeline Profiles Resources Audits Console

Heavy (Bottom Up) ▾ eye x refresh

Profiles	Self		Total		Function
	Time	Percentage	Time	Percentage	
CPU PROFILES	5930.9 ms	96.06%	5930.9 ms	96.06%	(idle)
	214.4 ms	3.47%	214.4 ms	3.47%	(program)
	2.1 ms	0.03%	18.6 ms	0.30%	⚠ b.event.add.v.handle
	4.1 ms	0.07%	16.5 ms	0.27%	▶ b.event.dispatch
	0 ms	0%	10.3 ms	0.17%	animateFrame
	3.1 ms	0.05%	10.3 ms	0.17%	▶ animateCloud
	3.1 ms	0.05%	7.2 ms	0.12%	▶ P
	0 ms	0%	6.2 ms	0.10%	▶ b.extend._data
	4.1 ms	0.07%	4.1 ms	0.07%	▶ b.extend.acceptData
	0 ms	0%	4.1 ms	0.07%	▶ b.fn.b.init
	0 ms	0%	4.1 ms	0.07%	▶ b.fn.extend.find
	0 ms	0%	4.1 ms	0.07%	▶ b
	1.0 ms	0.02%	3.1 ms	0.05%	▶ b.event.fix
	0 ms	0%	3.1 ms	0.05%	▶ b.fn.extend.css
	1.0 ms	0.02%	3.1 ms	0.05%	▶ b.extend.access
	2.1 ms	0.03%	2.1 ms	0.03%	▶ ⚠ st
	0 ms	0%	2.1 ms	0.03%	▶ (anonymous function)
	1.0 ms	0.02%	2.1 ms	0.03%	▶ onScroll
	2.1 ms	0.03%	2.1 ms	0.03%	▶ b.fn.b.pushStack
	2.1 ms	0.03%	2.1 ms	0.03%	▶ b.Event
	1.0 ms	0.02%	1.0 ms	0.02%	▶ ⚠ b.extend.style
	1.0 ms	0.02%	1.0 ms	0.02%	▶ b.extend.camelCase
	1.0 ms	0.02%	1.0 ms	0.02%	▶ updateFrame

4.1 ms	0.07 %	► b.fn.b.init	jquery.min.js:2
4.1 ms	0.07 %	► b.fn.extend.find	jquery.min.js:3
4.1 ms	0.07 %	► b	jquery.min.js:2
3.1 ms	0.05 %	► b.event.fix	jquery.min.js:2
3.1 ms	0.05 %	► b.fn.extend.css	jquery.min.js:4
3.1 ms	0.05 %	► b.extend.access	jquery.min.js:2
2.1 ms	0.03 %	►⚠️ st	jquery.min.js:3
2.1 ms	0.03 %	►(anonymous function)	jquery.min.js:4
2.1 ms	0.03 %	►onScroll	main.js:6
2.1 ms	0.03 %	►b.fn.b.pushStack	jquery.min.js:2
2.1 ms	0.03 %	►b.Event	jquery.min.js:2
1.0 ms	0.02 %	►⚠️ b.extend.style	jquery.min.js:4
1.0 ms	0.02 %	►b.extend.camelCase	jquery.min.js:2
1.0 ms	0.02 %	►updateFrame	main.js:19

Find out more!

[ng-inspector](#)

[Ember Inspector](#)

[React DevTools](#)

[The Breakpoint](#)

Find out more!

Firefox Developer Tools

Firebug

Internet Explorer Developer Tools

Remote Debugging for Mobile

Summary

- Elements tab for debugging HTML and CSS
- Network tab for debugging HTTP
- Sources tab for debugging flow and state
- Timeline for debugging performance broadly



“Everyone keep cool... Let's solve the problem and not make it any worse by guessin’” Gene Kranz



Thank you



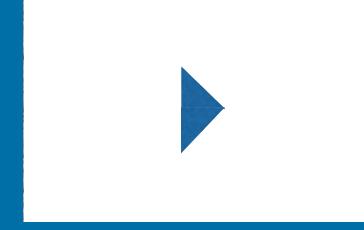
Thank you



@katie_fenn



tiny.cc/kf-wc-2016



tiny.cc/kf-wc-2016-videos