

CHROME DEV TOOLS
FRONT

CHROME DEV TOOLS
TO BACK

Follow Along!



@katie_fenn



<http://tiny.cc/cdt-ftb>

Debug:

verb [T] (REMOVE MISTAKES)

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

to debug a program

Cambridge Dictionaries Online

Early debugging:

```
alert ("Debug");
```

Early debugging:

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

Early debugging:

- Monitoring program flow and state
- Debug messages are not interactive

Script Debugging

The Sources tab

Debugging Scripts:

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

Debugging Scripts:

debugger;

Developer Tools - http://localhost:9001/

Elements Network Sources Timeline Profiles Resources Audits Console

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

▶ (no domain)
▶ ajax.googleapis.com
▶ hello.myfonts.net
▼ localhost:9001
▶ css
▶ js
▷ (index)
▶ www.google-analytics.com

```
1 $('.select-navigation select').selectNavigation();
2
3 var scrollMax = $(document).height() - window.innerHeight,
4     windowScroll = getWindowScroll(),
5     ticking = false;
6
7 $(window).on('scroll', function onScroll () {
8     windowScroll = getWindowScroll();
9     updateFrame(windowScroll);
10});
11
12 /*function updateFrame(windowScroll) {
13     animateCloud(windowScroll);
14 }
15
16 function getWindowScroll() {
17     return (1 / $(document).height() - window.innerHeight) * w
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
29     ticking = true;
30 }
31
32 function getWindowScroll () {
33     return 1 / (scrollMax * window.pageYOffset);
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: function...
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 control icons: a play button, a refresh arrow, a dropdown arrow, a double up arrow, a double down arrow, a refresh symbol, and a pause button.

The main area is divided into several sections:

- Watch Expressions**: A list containing:
 - window.pageYOffset: 269
 - \$('h1').first().text(): "Katie F..."
- Call Stack**: A yellow-highlighted section containing:

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

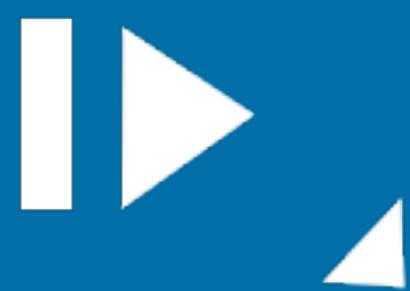
updateFrame	main.js:21
onScroll	main.js:9
- Scope Variables**: A section containing:
 - Local**:
 - this: Window
 - windowScroll: 0.00000188991973...

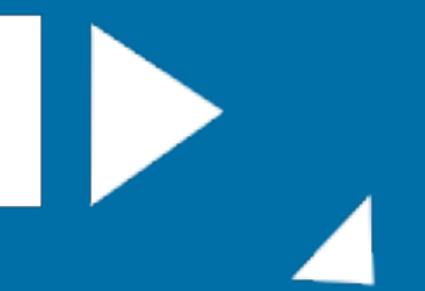


```
).selectNavigation();  
  
height() - window.innerHeight,  
scroll(),  
  
on onScroll () {  
  scroll();  
}  
  
Scroll) {  
};
```

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 main sections: "Watch Expressions" and "Call Stack". The "Watch Expressions" section contains two entries: `window.pageYOffset: 269` and `($('h1').first().text(): "Katie F...`. The "Call Stack" section has a checkbox labeled "Async" followed by the text "2 stack frames are hidden (black-boxed). [Show](#)". Below this, the call stack is listed with two frames: "updateFrame" at main.js:21 and "onScroll" at main.js:9.

Call Stack	File	Line
updateFrame	main.js	21
onScroll	main.js	9

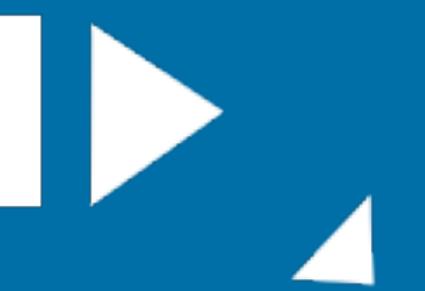




Resume script execution

```
    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);  
}
```

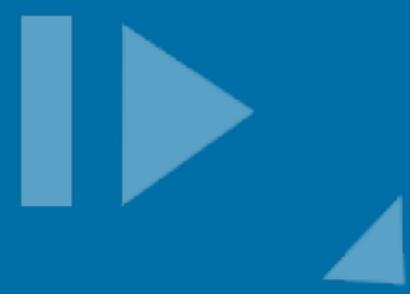


Resume script execution

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

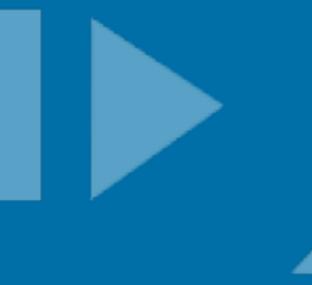
```
this.save = function (options) {  
    this.status = this.get('willPublish')  
    return get(this, keyName);
```

```
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);  
}
```

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



Katie Fenn

localhost:9001

Should disable UI?

Posted 2013-08-11 10:45

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

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

Some are claiming JavaScript is a functional need of the web, while others believe capabilities can be used to enhance the user experience. One person even suggested that users should be able to execute scripts in my browser.

I'm sure that 95% of the content on the internet is porn, and I'm not a proponent of the power of JavaScript. I think custom-designed select boxes should be removed from the web.

However, I'd argue that most of the websites on the web today are designed for the user experience. Sure, they're not always well-designed, but they are designed to be user-friendly.

JavaScript is and should be a functional need of the web. It's an option from the consumer's perspective, supporting the absence of functionality specified in the context of the page. I don't deny that there are use cases where JavaScript is not needed, but they are now rare.

Custom-design select boxes should be removed from the web.

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

RECORDS

Summary Layers

Katie Fenn

localhost:9001

Should disable UI?

Posted 2013-08-11 10:45

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

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

Some are claiming JavaScript is a functional need of the web, while others believe capabilities can be used to improve the user experience. One person even suggested that users should be able to execute scripts in my browser.

I'm sure that 95% of the content on the internet is porn, and I'm not a proponent of the power of JavaScript. I think custom-designed select boxes should be removed from the web.

However, I'd argue that most of the websites on the web today are designed for the user experience. Sure, they're not always well-designed, but they are designed to be user-friendly.

JavaScript is and should be a functional need of the web. It's a tool for supporting the absence of features that are specified in the context of the page. It's not a tool for denying that there are user needs that are now rare.

Custom-design select boxes should be removed from the web.

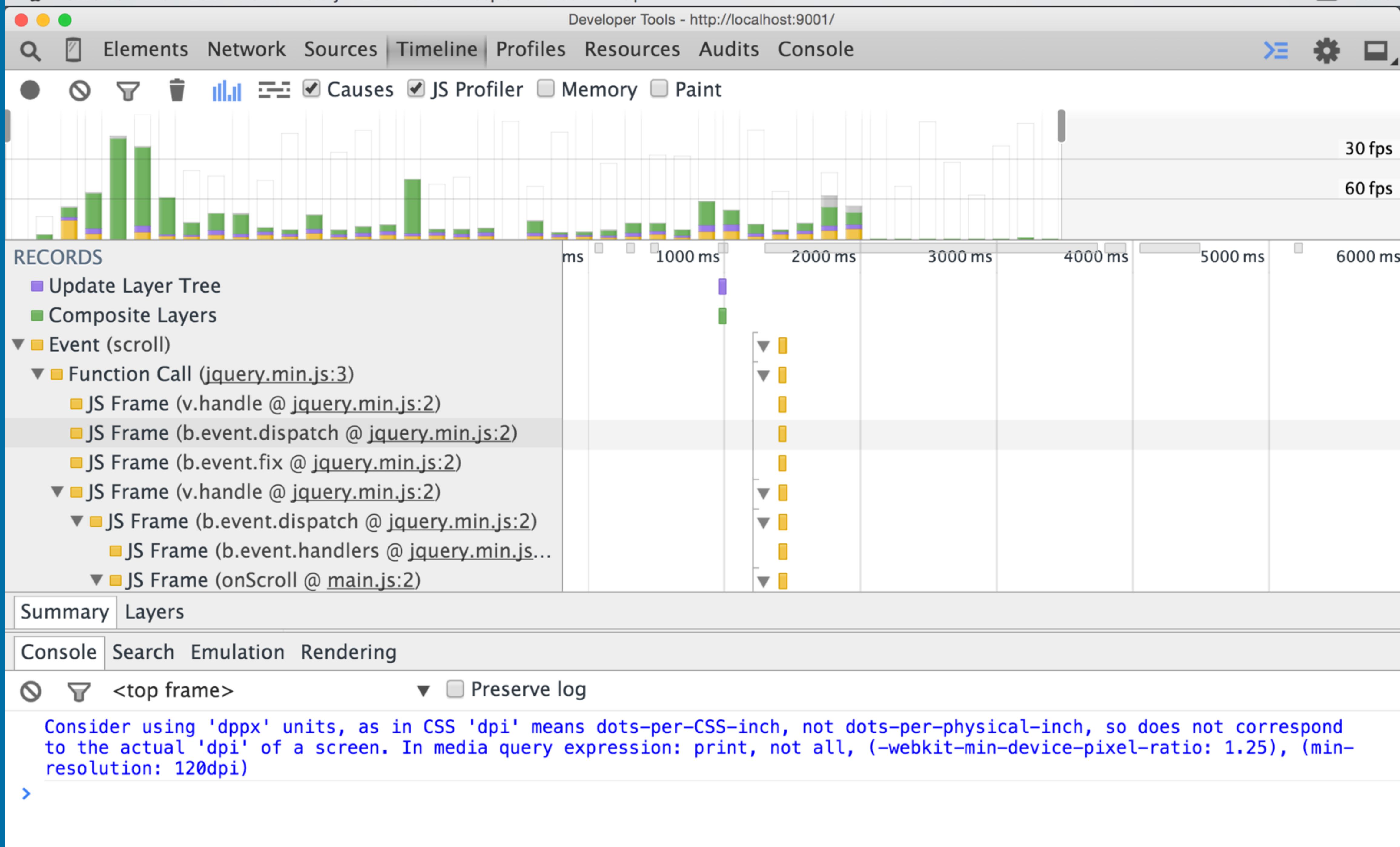
Developer Tools - http://localhost:9001/

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

200 ms 400 ms 600 ms 800 ms 1000 ms

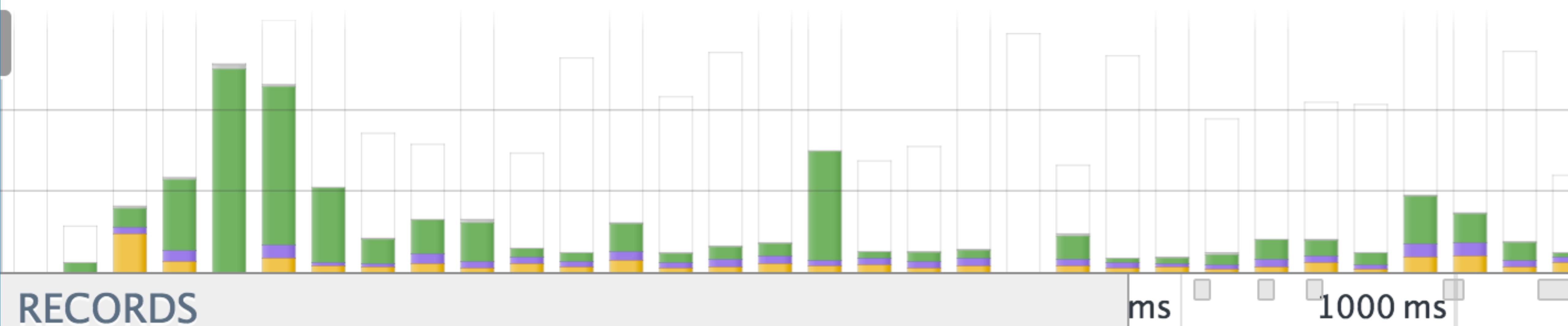
RECORDS

Summary Layers



Elements Network Sources Timeline Profiles Resources Audits

● ⚡ ⚏ ⚗ ⚑ ⚒ ⚓ Causes JS Profiler Memory Paint



Update Layer Tree

Composite Layers

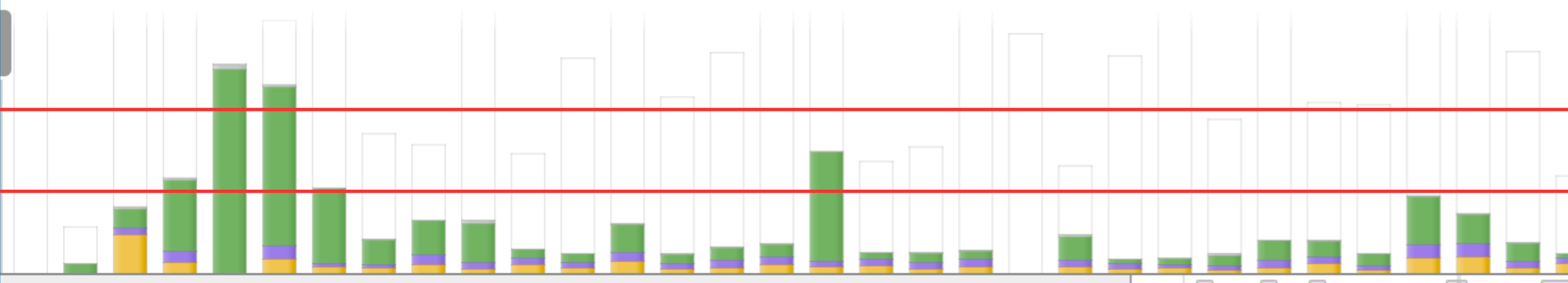
Event (scroll)

Function Call ([jquery.min.js:3](#))

JS Frame ([v.handle @ jquery.min.js:2](#))

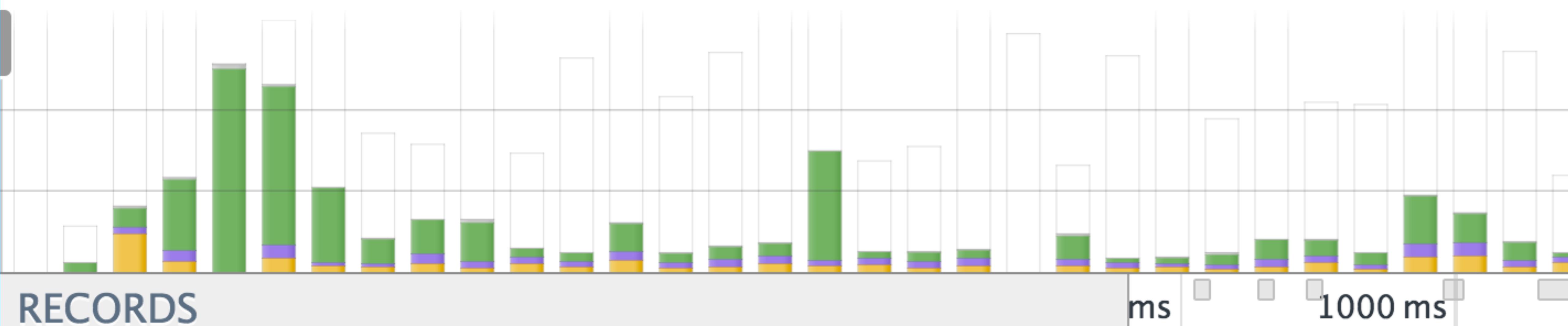
Elements Network Sources Timeline Profiles Resources Audits

● ⚡ ⚏ 🗑️ 📈 📜 Causes JS Profiler Memory Paint



Elements Network Sources Timeline Profiles Resources Audits

● ⚡ ⚏ ⚗ ⚑ ⚒ ⚓ Causes JS Profiler Memory Paint



Update Layer Tree

Composite Layers

Event (scroll)

Function Call ([jquery.min.js:3](#))

JS Frame ([v.handle @ jquery.min.js:2](#))

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

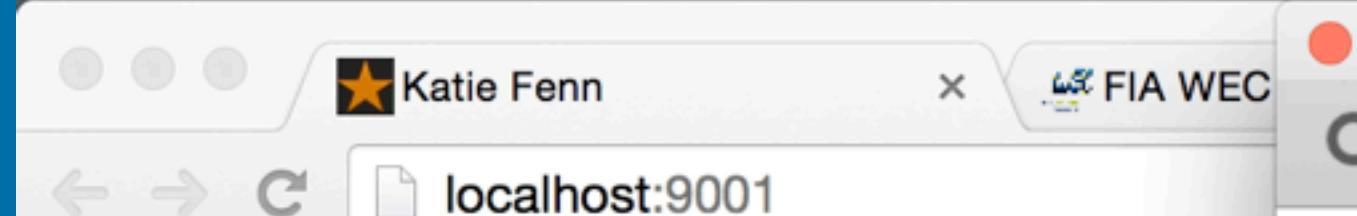
Search Emulation Rendering



<top frame>



Preserve log



Developer Tools - http://localhost:9001/

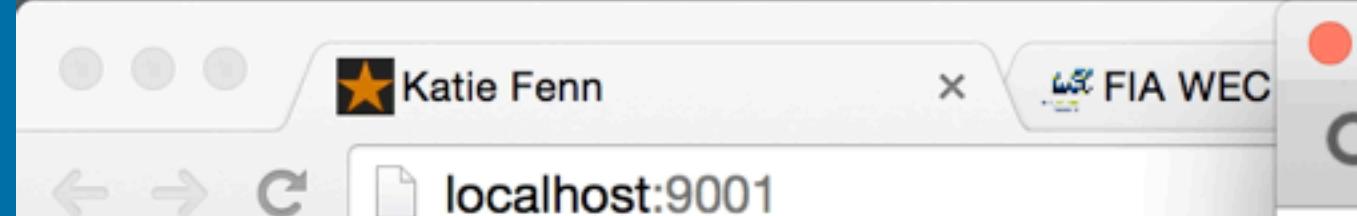
Elements Network Sources Timeline Profiles Resources Audits Console Causes JS Profiler Memory Paint

30 fps
60 fps

RECORDS

Summary Layers

The developer tools interface is shown in the background. The "Timeline" tab is active. The timeline shows several network requests and responses. A legend at the top indicates that blue bars represent "Causes" and orange bars represent "JS Profiler". The timeline has markers for "30 fps" and "60 fps". The "RECORDS" section below the timeline is mostly empty. At the bottom, there are tabs for "Summary" and "Layers".



Developer Tools - http://localhost:9001/

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

30 fps
60 fps

RECORDS

Summary Layers

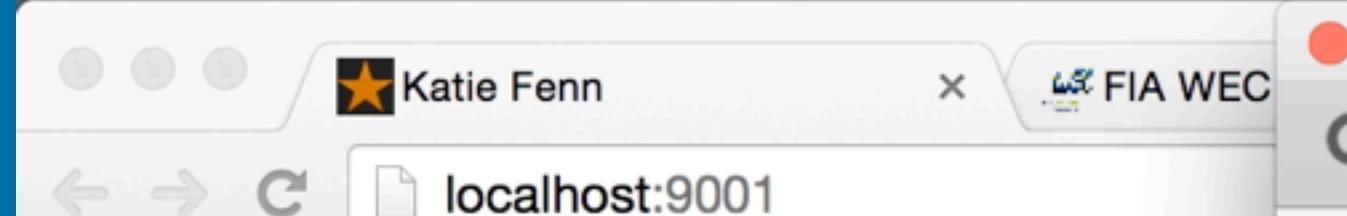
A screenshot of the Chrome Developer Tools Timeline tab. The title bar says "Developer Tools - http://localhost:9001/". The tabs at the top are Elements, Network, Sources, Timeline, Profiles, Resources, Audits, and Console. The Timeline tab is active. There are several icons below the tabs: a dot, a circle, a square, a funnel, a trash can, a bar chart, a grid, and a checkmark. To the right of these icons are checkboxes for "Causes" and "JS Profiler", both of which are checked. Below the timeline are two rows of text: "30 fps" and "60 fps". A large rectangular area labeled "RECORDS" is shown below the timeline. At the bottom of the developer tools window, there are tabs for "Summary" and "Layers", with "Summary" being the active tab.

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

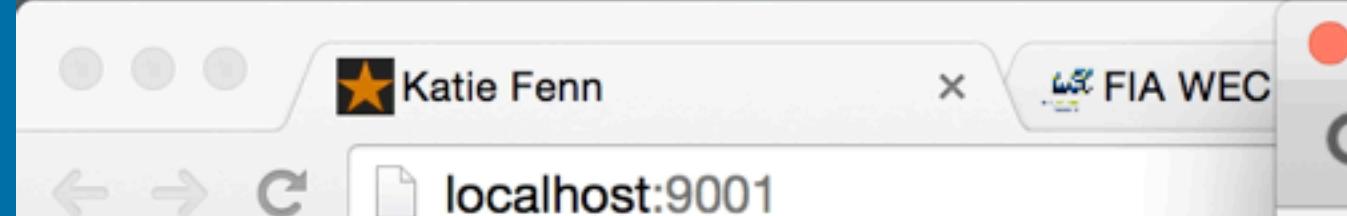
Profiles

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.



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.

	Self	Total	Function	
Profiles	5883.4 ms 93.34%	5883.4 ms 93.34%	(idle)	
CPU PROFILES	383.3 ms 6.08%	383.3 ms 6.08%	(program)	
	2.1 ms 0.03%	36.4 ms 0.58%	v.handle	jquery.min.js:2
	2.1 ms 0.03%	34.3 ms 0.54%	▶ b.event.dispatch	jquery.min.js:2
	0 ms 0%	27.8 ms 0.44%	▶ onScroll	main.js:2
	0 ms 0%	23.6 ms 0.37%	▶ updateFrame	main.js:7
	0 ms 0%	23.6 ms 0.37%	▶ animateCloud	main.js:15
	1.1 ms 0.02%	23.6 ms 0.37%	▶ b.extend.access	jquery.min.js:2
	1.1 ms 0.02%	20.3 ms 0.32%	▶ b.fn.extend.css	jquery.min.js:4
	1.1 ms 0.02%	18.2 ms 0.29%	▶ (anonymous function)	jquery.min.js:4
	15.0 ms 0.24%	17.1 ms 0.27%	▶⚠ b.extend.style	jquery.min.js:4
	0 ms 0%	4.3 ms 0.07%	▶ b.fn.(anonymous function)	jquery.min.js:4
	2.1 ms 0.03%	4.3 ms 0.07%	▶ (anonymous function)	jquery.min.js:4
	0 ms 0%	4.3 ms 0.07%	▶ getWindowScroll	main.js:11
	1.1 ms 0.02%	3.2 ms 0.05%	▶ b	jquery.min.js:2
	2.1 ms 0.03%	2.1 ms 0.03%	▶ b.event.fix	jquery.min.js:2
	1.1 ms 0.02%	2.1 ms 0.03%	▶ P	jquery.min.js:2
	0 ms 0%	2.1 ms 0.03%	▶ b.fn.b.init	jquery.min.js:2
	0 ms 0%	2.1 ms 0.03%	▶ b.fn.extend.find	jquery.min.js:3
	0 ms 0%	2.1 ms 0.03%	▶ b.extend._data	jquery.min.js:2
	2.1 ms 0.03%	2.1 ms 0.03%	▶⚠ st	jquery.min.js:3
	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	jquery.min.js:2
	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	jquery.min.js:2
	1.1 ms 0.02%	1.1 ms 0.02%	▶ get source_url	

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

Heavy (Bottom Up) ▾ Eye X Reload

Profiles

CPU PROFILES

Profile 1 Save

	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

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

Elements Network Sources Timeline Profiles Resources Audits Console

Heavy (Bottom Up) ▾ eye x refresh

Profiles	Self		Total		Function
	Time	Percentage	Time	Percentage	
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

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

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 Firefox 21, Mozilla has removed the ability to disable JavaScript from the user interface. Mozilla has added a new preference to the browser's about:config interface, `javascript.enabled`, which controls whether or not users can disable JavaScript in the browser's settings. This is a problem.

A [debate is unfolding at Ars Technica](#) over whether or not it's a good idea to disable JavaScript on the web.

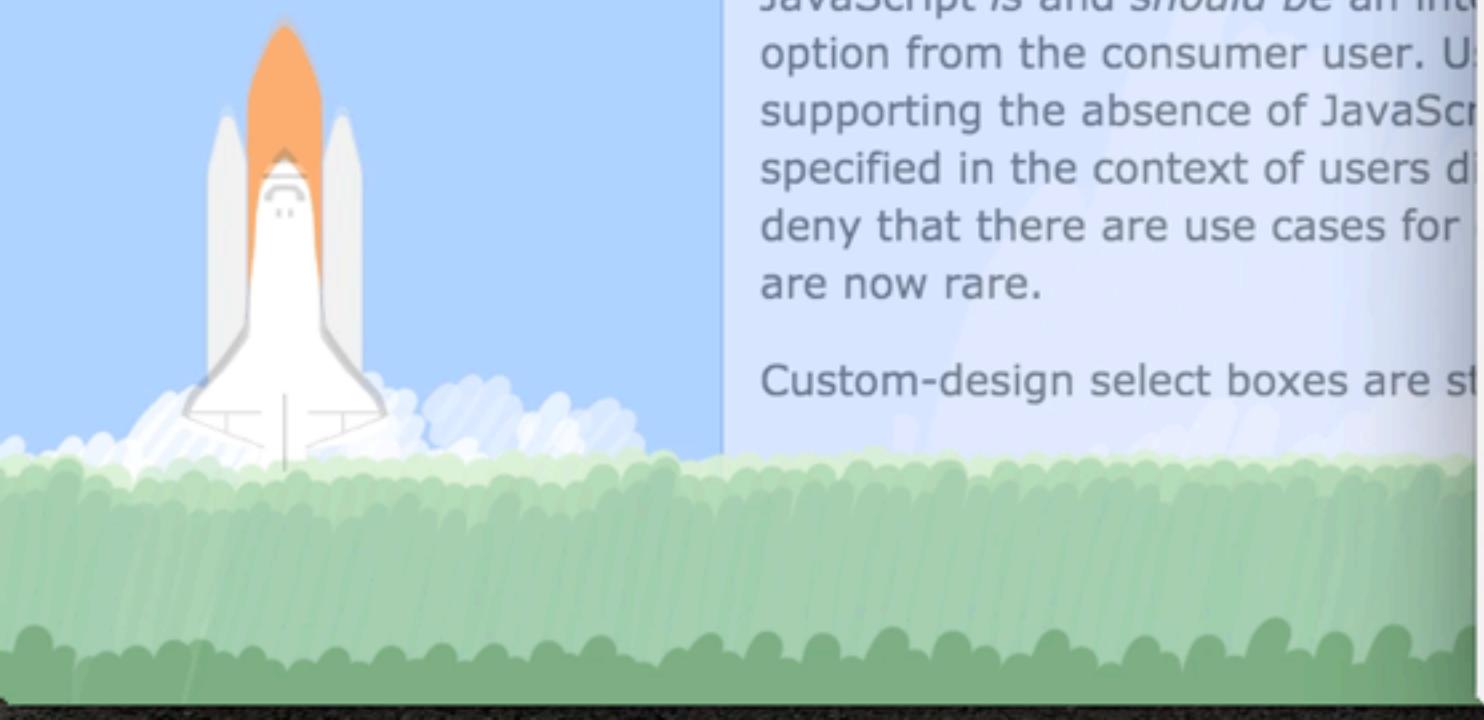
Some are claiming JavaScript is unnecessary for most web pages. They point to the functional needs of about 95% of the web. Most of the remaining 5% of capabilities can be used to design web pages that are better for users. One person even slam-dunked the argument by saying "I don't execute scripts in my browser" that's like saying "I don't eat meat" because I don't eat meat.

I'm sure that 95% of the internet is porn, and that certain people are proponent of the power of the web. But there are still some well-designed select boxes sure as hell.

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

JavaScript *is* and *should be* an integral part of the web. It's a tool, not a weapon. Supporting the absence of JavaScript is a bad idea. It's not specified in the context of users disabled JavaScript. It's not clear if users deny that there are use cases for JavaScript. But the use cases 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**

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 Firefox 21, Mozilla has removed the ability to disable JavaScript from the user interface. Mozilla has added a new preference to the browser's about:config interface, `javascript.enabled`, which controls whether or not users can disable JavaScript in the browser's settings. This is a problem.

A [debate is unfolding at Ars Technica](#) over whether or not it's a good idea to disable JavaScript on the web.

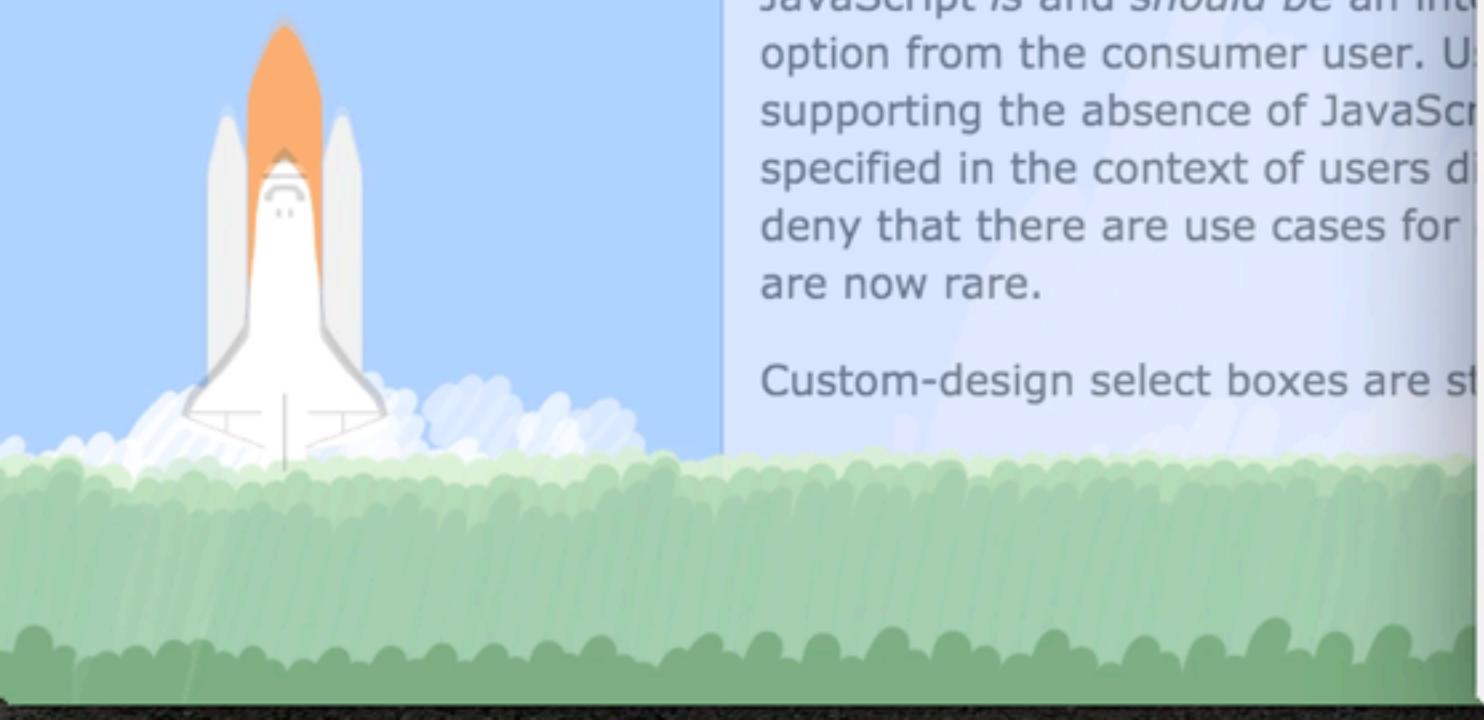
Some are claiming JavaScript is unnecessary for most web pages. They point to the functional needs of about 95% of the web. Most of the remaining 5% of capabilities can be used to design web pages that are better for users. One person even slam-dunked the argument by saying "I don't execute scripts in my browser" that's like saying "I don't eat meat" because I don't eat meat.

I'm sure that 95% of the internet is porn, and that certain people are proponent of the power of the web. But there are still some well-designed select boxes sure as hell.

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

JavaScript *is* and *should be* an integral part of the web. It's a tool, not a weapon. Supporting the absence of JavaScript is a bad idea. It's not specified in the context of users disabled JavaScript. It's not clear if users deny that there are use cases for JavaScript. But the use cases 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**

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

Debugging the Server

Using Node-Inspector to debug Node.JS scripts

Installing Node-Inspector:

Installing Node-Inspector:

```
npm install -g node-inspector
```

bash

vim

10:49:19 \$>

katie Tombo ~/Sites/website-ghost

10:49:43 \$> █

}

bash

vim

10:49:19 \$>

katie Tombo ~/Sites/website-ghost

10:49:43 \$> █

}

—debug-brk

Lewis - Katie Fenn Editor - Katie Fenn Node Inspector - file:///User... 127.0.0.1:8080/debug?ws=127.0.0.1:8080&port=5858

Sources Profiles Console

Sour... Cont... Snip... index.js x frontend.js node.js

▶ (core modules)
▼ file:///
 ▼ Users/katie/Sites/we...
 ▶ content/themes
 ▶ core
 ▶ node_modules
 JS Gruntfile.js
 JS config.example.js
 JS config.js
 JS index.js

```
1 (function (exports, require, module, __filename, __dirname) { //  
2 // Orchestrates the loading of Ghost  
3 // When run from command line.  
4  
5 var express,  
6   ghost,  
7   parentApp,  
8   errors;  
9  
10 // Make sure dependencies are installed and file system permissions  
11 require('./core/server/utils/startup-check').check();  
12  
13 // Proceed with startup  
14 express = require('express');  
15 ghost = require('./core');  
16 errors = require('./core/server/errors');  
17  
18 // Create our parent express app instance.  
19 parentApp = express();  
20  
21 ghost().then(function (ghostServer) {  
22   // Mount our ghost instance on our desired subdirectory path  
23   parentApp.use(ghostServer.config.paths subdir, ghostServer.r...  
24  
25   // Let ghost handle starting our server instance.  
26   ghostServer.start(parentApp);  
27 }).catch(function (err) {  
28   errors.logErrorAndExit(err, err.context, err.help);  
29 });  
30  
31 });
```

{} Line 5, Column 1

▶ Watch Expressions + C
▼ Call Stack □ Async
 (anonymous function) index.js:5
 Module._compile module.js:460
 module.js:478
 Module._extensions..js
 Module.load module.js:355
 Module._load module.js:310
 Module.runMain module.js:501
 listOnTimeout timers.js:110

▼ Scope Variables

▼ Local

- __dirname: "/Users/katie/Sit...
__filename: "/Users/katie/Si...
errors: undefined
- ▶ exports: Object
- ▶ express: undefined
- ▶ ghost: undefined
- ▶ module: Module
- ▶ parentApp: undefined
- ▶ require: function require(pa...
▶ this: Object

▶ Global Object

▼ Breakpoints

Katie Fenn Node Inspector - file:///User/Katie/Fenn

127.0.0.1:8080/debug?ws=127.0.0.1:8080&port=5858

Sources Profiles Console

Sour... Cont... Snip... index.js frontend.js node.js

▶ (core modules)
▼ file:///
 ▼ Users/katie
 ► .nvm/versions/node
 ▼ Sites/website-ghost
 ► content/themes
 ► core
 ► node_modules
 JS Gruntfile.js
 JS config.example.js
 JS config.js
 JS index.js

author: function (req, res, next) {
 // Parse the page number
 var pageParam = req.params.page !== undefined ? parseInt(req.params.page, 10) : 1;
 options = {
 page: pageParam,
 author: req.params.slug
 };
 // Get url for tag page
 function authorUrl(author, page) {
 var url = config.paths.subdir + '/' + config.routeKeywords.page + '/' + author + '/' + page;
 if (page && page > 1) {
 url += config.routeKeywords.page + '/' + page + '/' + page;
 }
 return url;
 }
 // No negative pages, or page 1
 if (isNaN(pageParam) || pageParam < 1 || (req.params.page < 1)) {
 return res.redirect(authorUrl(options.author));
 }
 return getPostPage(options).then(function (page) {
 // If page is greater than number of pages we have,
 if (pageParam > page.meta.pagination.pages) {
 return res.redirect(authorUrl(options.author, page));
 }
 setReqCtx(req, page.posts);
 // If page meta file has author, r

{} Line 240, Column 1

Watch Expressions + C
Call Stack □ Async
Not Paused
Scope Variables Not Paused
Breakpoints No Breakpoints

Katie Fenn Node Inspector - file:///User/Katie/Fenn

127.0.0.1:8080/debug?ws=127.0.0.1:8080&port=5858

Sources Profiles Console

Sour... Cont... Snip... index.js frontend.js node.js

▶ (core modules)
▼ file:///
 ▼ Users/katie
 ► .nvm/versions/node
 ▼ Sites/website-ghost
 ► content/themes
 ► core
 ► node_modules
 JS Gruntfile.js
 JS config.example.js
 JS config.js
 JS index.js

author: function (req, res, next) {
 // Parse the page number
 var pageParam = req.params.page !== undefined ? parseInt(req.params.page, 10) : 1;
 options = {
 page: pageParam,
 author: req.params.slug
 };
 // Get url for tag page
 function authorUrl(author, page) {
 var url = config.paths.subdir + '/' + config.routeKeywords.page + '/' + author + '/' + page;
 if (page && page > 1) {
 url += config.routeKeywords.page + '/' + page + '/' + page;
 }
 return url;
 }
 // No negative pages, or page 1
 if (isNaN(pageParam) || pageParam < 1 || (req.params.page < 1)) {
 return res.redirect(authorUrl(options.author));
 }
 return getPostPage(options).then(function (page) {
 // If page is greater than number of pages we have,
 if (pageParam > page.meta.pagination.pages) {
 return res.redirect(authorUrl(options.author, page));
 }
 setReqCtx(req, page.posts);
 // If page meta file has author, r

{} Line 240, Column 1

Watch Expressions + C
Call Stack □ Async
Not Paused
Scope Variables Not Paused
Breakpoints No Breakpoints

▼ Scope Variables

▼ Local

▼ req: Object

▼ params: Object

slug: "katie"

► __proto__: Object

► __proto__: Object

► this: Window

► Global

Window

▼ Scope Variables

▼ Local

▼ req: Object

▼ params: Object

slug: "katie"

► __proto__: Object

► __proto__: Object

► this: Window

► Global

Window

Lewis - Katie Fenn × Node Inspector - file:///User... × Editor - Katie Fenn ×

127.0.0.1:8080/debug?ws=127.0.0.1:8080&port=5858

Sources Profiles Console

Sour... Cont... Snip... index.js frontend.js × node.js

▶ (core modules)
▼ file:///
 ↳ Users/katie
 ↳ .nvm/versions/node
 ↳ Sites/website-ghost
 ↳ content/themes
 ↳ core
 ↳ node_modules
 JS Gruntfile.js
 JS config.example.j
 JS config.j
 JS index.j

author: function (req, res, next) {
 // Parse the page number
 var pageParam = req.params.page !== undefined ? parseInt(req.params.page, 10) : 1;
 options = {
 page: pageParam,
 author: req.params.slug
 };

 // Get url for tag page
 function authorUrl(author, page) {
 var url = config.paths.subdir + '/' + config.routeKeywords.page + '/' + page + '/';

 if (page && page > 1) {
 url += config.routeKeywords.page + '/' + page + '/';
 }

 return url;
 }

 // No negative pages, or page 1
 if (isNaN(pageParam) || pageParam < 1 || (req.params.page < 1)) {
 return res.redirect(authorUrl(options.author));
 }

 return getPostPage(options).then(function (page) {
 // If page is greater than number of pages we have,
 if (pageParam > page.meta.pagination.pages) {
 return res.redirect(authorUrl(options.author, page));
 }

 setReqCtx(req, page.posts);
 if (page.meta.filters.author) {
 return res.redirect(authorUrl(options.author, page));
 }
 }).catch(function (err) {
 if (err.message === 'Not Found') {
 return res.redirect('/');
 }
 return res.status(500).end();
 });
};

{} Line 240, Column 1

_maxListeners: undefined
► _parsedOriginalUrl: Url
► _parsedUrl: Url
► _passport: Object
_pendingIndex: 0
► _pendings: Array[0]
► _readableState: ReadableStat
_remoteAddress: "127.0.0.1"
► _startAt: Array[2]
► _startTime: Sun May 03 201...
baseUrl: ""
► body: Object
► client: Socket
complete: true
► connection: Socket
domain: null
► headers: Object
httpVersion: "1.1"
httpVersionMajor: 1
httpVersionMinor: 1
method: "GET"
► next: function next(err) {
 originalUrl: "/author/kati...
► params: Object
 slug: "lewis"
 ► __proto__: Object
► query: Object
► rawHeaders: Array[14]
► rawTrailers: Array[0]
readable: true

Lewis - Katie Fenn × Node Inspector - file:///User... × Editor - Katie Fenn ×

127.0.0.1:8080/debug?ws=127.0.0.1:8080&port=5858

Sources Profiles Console

Sour... Cont... Snip... index.js frontend.js × node.js

▶ (core modules)
▼ file:///
 ↳ Users/katie
 ↳ .nvm/versions/node
 ↳ Sites/website-ghost
 ↳ content/themes
 ↳ core
 ↳ node_modules
 JS Gruntfile.js
 JS config.example.j
 JS config.j
 JS index.j

author: function (req, res, next) {
 // Parse the page number
 var pageParam = req.params.page !== undefined ? parseInt(req.params.page, 10) : 1;
 options = {
 page: pageParam,
 author: req.params.slug
 };

 // Get url for tag page
 function authorUrl(author, page) {
 var url = config.paths.subdir + '/' + config.routeKeywords.author + '/' + author + '/';

 if (page && page > 1) {
 url += config.routeKeywords.page + '/' + page + '/';
 }

 return url;
 }

 // No negative pages, or page 1
 if (isNaN(pageParam) || pageParam < 1 || (req.params.page > config.pagination.maxPages)) {
 return res.redirect(authorUrl(options.author));
 }

 return getPostPage(options).then(function (page) {
 // If page is greater than number of pages we have,
 if (pageParam > page.meta.pagination.pages) {
 return res.redirect(authorUrl(options.author, page));
 }

 setReqCtx(req, page.posts);
 if (page.meta.filters.author) {
 return res.redirect(authorUrl(options.author, page));
 }
 }).catch(function (err) {
 if (err.message === 'Not Found') {
 return res.redirect('/');
 }
 return res.status(500).end();
 });
};

{} Line 240, Column 1

_maxListeners: undefined
► _parsedOriginalUrl: Url
► _parsedUrl: Url
► _passport: Object
_pendingIndex: 0
► _pendings: Array[0]
► _readableState: ReadableStat
_remoteAddress: "127.0.0.1"
► _startAt: Array[2]
► _startTime: Sun May 03 201...
baseUrl: ""
► body: Object
► client: Socket
complete: true
► connection: Socket
domain: null
► headers: Object
httpVersion: "1.1"
httpVersionMajor: 1
httpVersionMinor: 1
method: "GET"
► next: function next(err) {
 originalUrl: "/author/kati..."
► params: Object
 slug: "lewis"
 ► __proto__: Object
► query: Object
► rawHeaders: Array[14]
► rawTrailers: Array[0]
readable: true

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

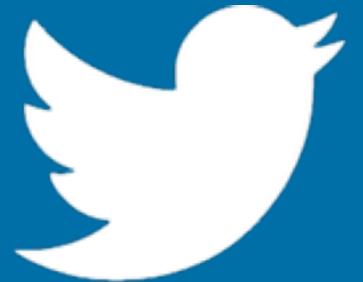
- DevTools has a huge roster of features
- Sources tab for debugging flow and state
- Timeline for debugging performance broadly
- Profiles for debugging specific script performance
- Node-Inspector for debugging Node.JS scripts



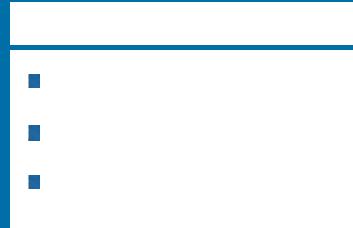
INVIQATM

We're hiring!

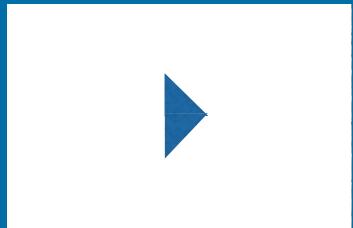
Thank You



@katie_fenn



<http://tiny.cc/cdt-ftb>



<http://tiny.cc/cdt-ftb-videos>