What is a reliable way to calculate actual (web) page loadtime

Asked 15 years, 9 months ago Modified 4 years, 10 months ago

Viewed 24k times

14

I'm interested in knowing the actual average page loadtime for my webapplication.

Simplistically, how log does my average visitor wait before they can start using a page on my site. From when they click the link to my site until the site is finished rendering & ready to accept input.

The standard solution seems to be to use Javascript to compare the time from a script in the until a script in the window.onload() event.

(See: http://www.dreamincode.net/code/snippet1908.htm)

This doesn't seem like a very acturate measure to me, as it ignores the time taken to resolve my domain & receive enough HTML content to begin Javascript parsig.

It also looks like Safari fires window.onload before the page has actually finished loading (http://www.howtocreate.co.uk/safaribenchmarks.html).

Any ideas?

Is it possible to get the time a the current request was initiated via Javascript?
What event fires after everything is ready reliably across all browsers?

javascript performance pageload

Share
Improve this question
Follow

asked Mar 20, 2009 at 14:38
David Laing
7,665 ● 10 ● 35 ● 44

DNS resolution time is something that is outside your control, there is no way to measure that reliably. – Ian Kemp Mar 20, 2009 at 14:43

11 Answers

Sorted by: Highest score (default)

8

http://www.webpagetest.org/ is an excellent resource for measuring load time

Also google chorme dev tools has a Timeline panel where you can record events, Here is a 2.5minute video showing you how Timeline in google chrome works http://www.youtube.com/watch?v=RhaWYQ44WEc

Share Improve this answer
Follow

edited Apr 27, 2012 at 14:03

answered Apr 27, 2012 at 13:51

zadubz
1,301 ● 2 ● 21 ● 36

3 Is any API available for the above ? Is there any other API available which can be used to calculate reliable load time of a website ? – SilentAssassin Mar 22, 2013 at 7:58

For reason I do not know the two tools provide a very different loading time in many pages – Lorenzo Belli Jun 10, 2017 at 16:51

3

FireBug has a "network timing mode" where you can see how long it took to download each resource which makes up your web page.

Plus you should measure the time your server needs to prepare the request. Since you can't influence the browser and the network, rendering time on your server should be as small as possible.

Share Improve this answer Follow

answered Mar 20, 2009 at 14:58

Aaron Digulla
328k ● 110 ● 622 ● 834

3

Firebug is a great resource for this and loads of other information about your page loads. Additionally, Firebug with YSlow goes one step further. YSlow has a hadnful of checks that it runs against your page and grades it's performance based on certain rules (are you using a CDN, is your CSS and JS compressed, etc.). I've found it invaluable to make some major improvements (JS compression is a great one) to my sites.

Share Improve this answer Follow

answered Mar 20, 2009 at 15:05

Milner
638 ● 7 ● 19

1 I agree, Firebug is a useful tool. However, it only measures the load time for MY PC/ browser / network; not an average time for the actual users of my site (with their weird and wonderful combinations of PC speed, network location, browser type etc etc) – David Laing Mar 20, 2009 at 15:54

1

Try Yahoo's YSLOW, it will answer your post of the questions, but it works only with FireFox (actually its a plugin for firebug)

Share Improve this answer Follow

answered Oct 21, 2010 at 10:44

user482873
11 ● 1

0

The most accurate way to calculate load times is on the server side: once the page is built, how much it takes to display on the user's browser will depend on:

- Current network traffic;
- User's computer specs;
- Which browser he's using.

So, using JavaScript is not a great measure because there are a lot of factors you can't change in there.

The best thing you can do is measure the time each page takes to be generated on your server - *that* you can improve.

Needless to say, that will depend on which language you're coding in.

Share Improve this answer Follow

answered Mar 20, 2009 at 14:43

Seb
25.1k ● 5 ● 68 ● 85

1 Actually, there are plenty of things you can do to improve client side render time; from including less / smaller images, improving js render time, putting stuff on a CDN etc etc. But I agree; having a breakdown of where the delays lie would be very useful. – David Laing Mar 20, 2009 at 15:57

Granted; you're right. You can tweak JS as well; it's only I took your question server-side :) – Seb Mar 20, 2009 at 18:09

0

One thing that can be done is to use Javascript to grab the current time when a client-side event occurs that triggers a post-back to your server. Passing this value back to your server side will allow you to render it back to the client as your initial 'trigger' time that you can compare against.

Instead of using onLoad, I believe you can put script inline at the end of the document so that it runs as soon as the browser renders that portion of the script. This will allow you to compare the current time when the inline script runs against the trigger time that was captured when the user initiated the call.

However, as mentioned by Seb, since you can only reliably control the server-side of the loadtime, it would be best to include in your metrics the server page generation time. If you have both metrics, you can at least see how much of the total time is taken by page generation, and how much is dependent on the various delays that could occur on the client-side.

Share Improve this answer Follow

answered Mar 20, 2009 at 14:50

Jay S

7,994 ● 2 ● 41 ● 53

You can't rely on client-side time. It could be set to 8:00 AM on January 1969. – Josh Stodola Mar 20, 2009 at 15:08

It doesn't matter what the client-side time is. Your server doesn't use it, you simply pass it back to the server so it can include it in the response. All the calculations are done client-side, so the client's time settings are used for both the before and after. – Jay S Mar 20, 2009 at 15:25

0

I've always found the [Pingdom Tools full page test](https://tools.pingdom.com) very useful. It's not an in-code solution, but it does give you a good idea of how quickly (or not) your page loads.

Share Improve this answer

answered Mar 20, 2009 at 14:55

Follow

Philip Morton
132k ● 38 ● 90 ● 97

0

psuedo-code.

```
server marks start of processing the request.
server sends the output.
    script tag, marks start time.
    rest of html markup.
    client script , with server processing time in ms
    client script with window.load event + server-side
    client script which sends the total back via an aj
```

Share Improve this answer

Follow

answered Mar 20, 2009 at 21:33

Tracker1
19.3k ● 12 ● 85 ● 106

0

The method I use is to create a session variable (wrapped in an if to check that it hasn't already been set) as the first thing in my index.php file (which every script runs through). Then, I have a pageLoad event in javaScript that posts back to a different script on the server that gets the session varaible time and subtracts it from the current time. This gives you the time it took for the request to hit your server, process, respond, and render.

just make sure you unset the session variable in the script you post back to so that the next time the page loads it sets a new session variable (as one doesn't exist because we unset it) and starts all over again. The only thing you may want to do is find the time the ajax took to request the server and subtract that, but it should be milliseconds.

Share Improve this answer

Follow

answered Oct 21, 2010 at 11:16

Alex
7,374 ● 1 ● 20 ● 31

I should add, I use YUI and an onDomReady method, which is a valid check for the page having completely loaded and the DOM being ready :-) – Alex Oct 21, 2010 at 11:16

0

• Figure out how many bits you are transferring. (this converter helps: http://www.matisse.net/bitcalc/)

• Estimate or determine download speed (you can use this thing: http://speedtest.net)

• divide

Share Improve this answer

Follow

answered Sep 1, 2013 at 16:29

starsinmypockets
2,294 ● 4 ● 35 ● 45

0

tl;dr

Use a headless browser to measure the loading times. One example of doing so is Website Loading Time.

Long version

I ran into the same challenges you're running into, so I created a side-project to measure actual loading times. It uses Node and Nightmare to manipulate a headless ("invisible") web browser. Once all of the resources are loaded, it reports the number of milliseconds it took to fully load the page.

One nice feature that would be useful for you is that it loads the webpage repeatedly. So, you could load the page multiple times and then average the values for a nice, round value. Also, since this script runs on the command-line, it's trivial to install on different machines and get actual loading times from various locations. (Load time depends not just on the server, but also the client and intermediaries)

Example usage:

```
website-loading-time rinogo$ node website-loading-time
1657
967
1179
1005
1084
```

```
1076
...
```

https://github.com/rinogo/website-loading-time

Disclosure: I am the author of this project.

Share Improve this answer Follow

edited Jun 20, 2020 at 9:12
Community Bot
1 ● 1

answered Jan 29, 2020 at 21:50
rinogo
9,093 ● 12 ● 68 ● 107