



HOW TO IMPROVE SITE SPEED

WHAT MAY BE SLOWING YOU DOWN AND HOW TO FIX IT



CONTENTS

INTRODUCTION.....3

LIGHTHOUSE REPORT.....4

 WHAT IS THE ‘LIGHTHOUSE REPORT’?.....4

 RUNNING THE REPORT.....5

CONCLUSION.....16



INTRODUCTION

Speed has been an increasingly important ranking signal over the last few years – from the specific Speed update announced by Google at the turn of 2018 to the incorporation of speed in to the ‘core web vitals’ that they [announced in May](#) would be part of their ranking calculations. Google was able to report in April 2019 that, among slower sites, there had been a 15-20% improvement in speed, but there’s a long way to go.

With consumer expectation for rapidly loading, quickly accessible content growing and tolerance for lagging sites diminishing, it’s not only likely that their expectations will continue in the same direction, but also that Google will increase their [‘nudge theory’](#) speed weighting to gently guide webmasters where they want them to go.

This eBook will take a trip through a Lighthouse Report, and break down the meaning behind the various instances of jargon and technical language and offer some advice on how a website can improve its speed in accordance with the factors that Google deems important.



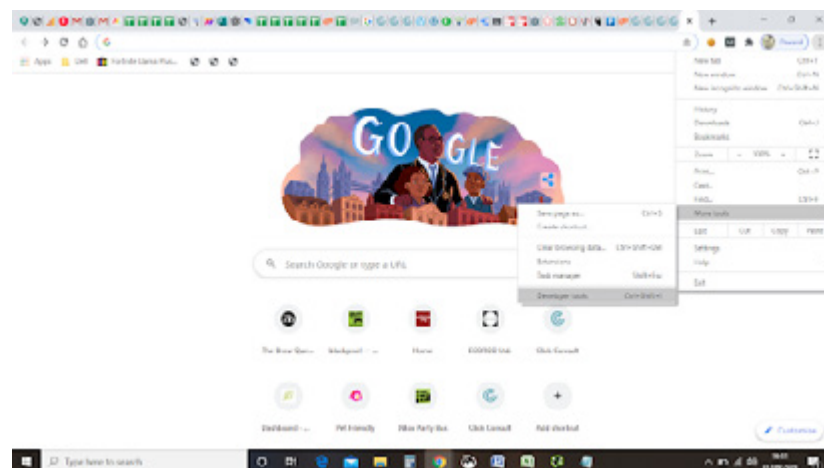
LIGHTHOUSE REPORT

WHAT IS THE 'LIGHTHOUSE REPORT'?

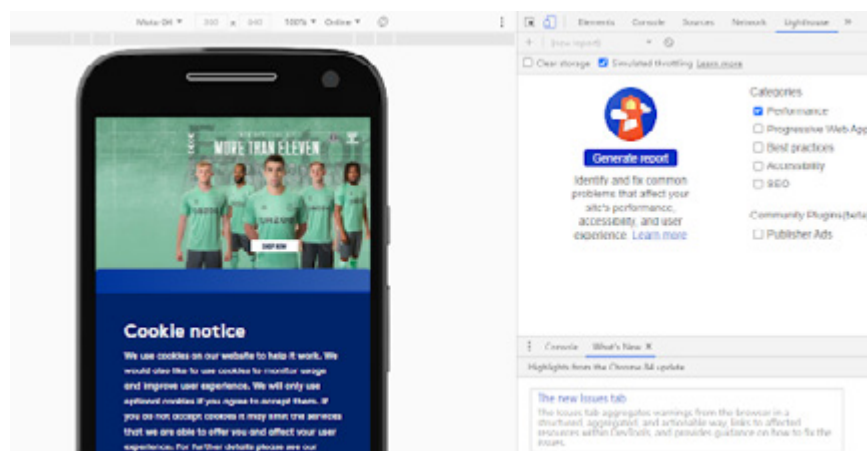
The tool we're going to be using to dissect a site is a free, Google tool called 'Lighthouse' – an open source platform to help analyse the quality of websites. It operates on a traffic light system and measures a host of different aspects of a site from performance through to SEO and accessibility (it can also test your PWA if you're so inclined).

How to access 'Lighthouse Report'

You can find the tool by clicking on the hamburger menu in chrome, then 'More tools' and then 'Developer tools'.

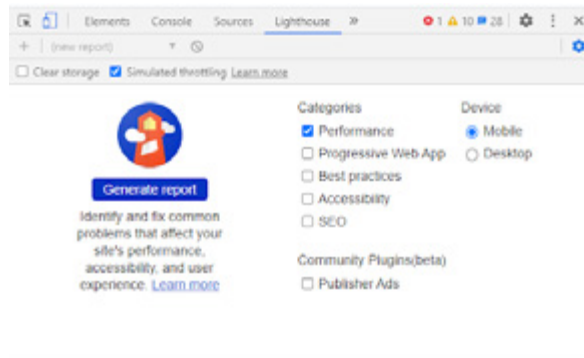


You'll then see a representation of the site you're looking at and it will either load the Lighthouse tab, or you'll need to select it so you see something like this:

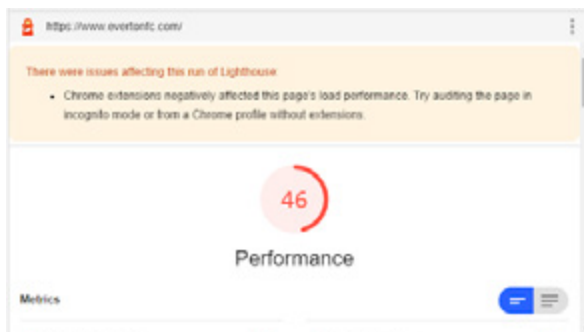


RUNNING THE REPORT

For the purposes of this, we're going to un-tick anything other than the performance option, but running the tool is as simple as clicking 'Generate report' once you've selected the options you want the report to include.



One thing to remember is that to get a true reflection of the performance, you'll need to run it in an incognito window as the various plugins you're using will impact the performance of the review and you'll see a message like this:



Once you've run it in an incognito window, your next scroll down will show you a set of ranked metrics (green to red), which look something like this:

Metrics					
●	First Contentful Paint	1.9 s	▲	Time to Interactive	11.3 s
■	Speed Index	5.4 s	▲	Total Blocking Time	8,270 ms
■	Largest Contentful Paint	3.2 s	▲	Cumulative Layout Shift	0.316
Values are estimated and may vary. The <u>performance score is calculated</u> directly from these metrics. <u>See calculator.</u>					

First Contentful Paint (FCP)

The FCP score is based on the amount of time, immediately after navigation to the site, when the browser renders the first bit of content from the DOM (Document Object Model – the tree structure that represents the HTML of the website). This is an important milestone for users as it provides feedback that the page is loading.

Speed Index

Speed Index is a page load performance metric which shows you how quickly the content of a page is visibly populated.

Largest Contentful Paint (LCP)

LCP measures how long it takes for the largest content element in the viewport to fully become visible and serves as a proxy for the time it takes for the page to have properly loaded.

Time to Interactive (TTI)

A page's loading is not a moment but an experience that no single metric can fully capture. There are multiple single points during the load that can affect whether a user perceives it as “fast” or “slow”. The TTI metric measures how long it takes a page to become interactive. “Interactive” is defined as the point where:

- The page has displayed useful content, which is measured with First Contentful Paint.
- Event handlers (the program or part of the program which will deal with interactions) are registered for most visible page elements.
- The page responds to user interactions within 50 milliseconds.

Some sites optimise content visibility at the expense of interactivity. While this can seem a good idea – as the consumer is only ever interacting with the page at its best aesthetically, this can create a frustrating user experience as the site appears to be ready, but, when the user tries to interact with it, nothing happens.

Total Blocking Time (TBT)

TBT tracks the amount of time a site blocks interaction – mouse clicks, screen taps etc. and is (as an approximate – the actual calculation is more complicated) the time between your FCP and TTI. There's a more in-depth description [here](#).

Cumulative Layout Shift (CLS)

CLS is a measure of the total time a site experiences unexpected shifts between one rendered frame (snapshot) and another. This is things like text suddenly shifting to make way for an asynchronously loaded element.

Issues and solutions

What follows is a run through the issues log for The Guardian – including passed tests – so while the order may vary, the same tests are run for any site which runs the performance element of the report.

Third party code

First up in The Guardian report is third party code – as you’ll see in the following image, this is external code you’ve placed somewhere on your site, such as Google Analytics or various advertising, shopping and tracking tools which require a third party call to enable them.



The screenshot shows a performance report for the URL https://www.theguardian.com/uk. It highlights a critical issue: 'Reduce the impact of third-party code' with a sub-issue 'Third-party code blocked the main thread for 609 ms'. Below this, a table lists the third-party providers and their impact on the main thread.

Third Party	Transfer Size	Main-Thread Blocking Time
Google Analytics	0 KB	373 ms
Fastly	0 KB	257 ms
Integral Ad Science	0 KB	54 ms

Below the table, other issues are listed: 'Avoid an excessive DOM size' (4,090 elements) and 'Reduce JavaScript execution time' (18.9 s).

The incredible measurability of digital marketing is, in this instance a bit of a rod we built for our own backs, as the expectations for thorough reporting often necessitates such code snippets which, if placed inexpertly, will slow down the load times of your site.

How to resolve the issue

For any external call that is not integral to the proper function of your site (those in the <head> of the page, at least), you can use asynchronous loading – which essentially tells the parser that these code snippets can wait a while – so your script calls will look like the following:

```
<script async src="script.js"></script>
```

DOM size

Your site's Document Object Model (DOM) is the branching tree diagram that represents your website, showing the hierarchical structure of the site from your home page to the deepest page (the page that would require the most clicks to reach).

According to Google, Lighthouse will fail pages with DOM trees that:

- Have more than 1,500 nodes total.
- Have a depth greater than 32 nodes.
- Have a parent node with more than 60 child nodes.



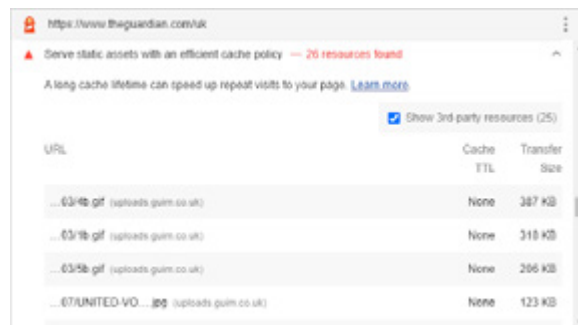
Statistic	Element	Value
Total DOM Elements		4,090
Maximum DOM Depth	<code><path d="M69.587,30.1842 15.556,3.89 31.316,4.708 48.2037,043c3.07-16.784 8.391-32.544 17.602-48.1h14.942a932.949,5c2.047 15.556-4.404 31.316-4.932 48.1h.203.866 32.236 8.592 16.456 17.538.9815.352"></code>	21
Maximum Children	<code><div class="facinome" data-linkname="Front 1 /uk" role="main"></code>	39

How to resolve the issue

This is naturally going to be difficult for a news site, but is easily achievable for most SMEs. Simply put, your site's hierarchy determines how easy it is for a consumer to get from one end of your site to the other and you want to ensure that it's as easy for them as possible. You can read a little more about internal linking and site structure [here](#).

Efficient cache policy

When requesting a resource from a server, the browser will receive not only the resource, but also instructions as to how long to store that resource locally. An ‘efficient’ cache policy is one that takes in to account how regularly certain resources change and assigns a cache term accordingly.



The screenshot shows the Chrome DevTools 'Performance' tab with the 'Network' sub-tab selected. A message at the top states: 'Serve static assets with an efficient cache policy --- 26 resources found'. Below this, a link says 'A long cache lifetime can speed up repeat visits to your page. Learn more'. A checkbox labeled 'Show 3rd party resources (25)' is checked. A table lists resources with columns for 'URL', 'Cache TTL', and 'Transfer Size'. The first four rows are truncated, showing URLs from 'uploads.guim.co.uk' and a 'Cache TTL' of 'None'.

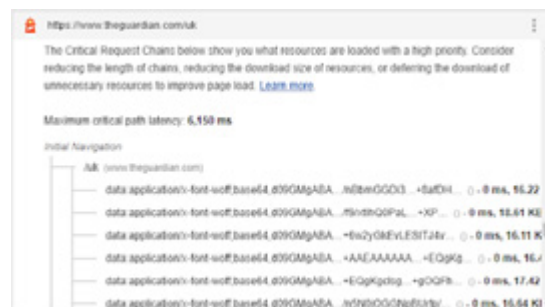
URL	Cache TTL	Transfer Size
...G3Hb.gif (uploads.guim.co.uk)	None	387 KiB
...G3Hb.gif (uploads.guim.co.uk)	None	318 KiB
...G3Hb.gif (uploads.guim.co.uk)	None	286 KiB
...G7UNITED-VO...jpg (uploads.guim.co.uk)	None	123 KiB

How to resolve the issue

Simply put, there needs to be lengthy cache limits for things like logos, CSS and JS files as well as downloadable assets (unless they're regularly updated). There's a really in-depth look at caching [here](#).

Critical Request Chains

Critical request chains – as you might expect – are requests which are critical to the page load but which are forced to happen in a chain mixed in with non-critical requests, and which slow down the page load time as a result.



The screenshot shows the Chrome DevTools 'Performance' tab with the 'Network' sub-tab selected. A message at the top states: 'The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. Learn more'. Below this, a link says 'Maximum critical path latency: 6,458 ms'. A section titled 'Initial Navigation' shows a tree view of requests. The first request is 'AJK (www.theguardian.com)'. Subsequent requests are truncated, showing data application/font-woff/base64 and a '0 ms' latency.

Request	Latency
AJK (www.theguardian.com)	
data application/font-woff/base64...	0 ms
data application/font-woff/base64...	0 ms
data application/font-woff/base64...	0 ms
data application/font-woff/base64...	0 ms
data application/font-woff/base64...	0 ms
data application/font-woff/base64...	0 ms

This can be things like loading multiple font types in the middle of two critical requests, or – as mentioned earlier – loading a number of external tool scripts before being able to load the page elements.

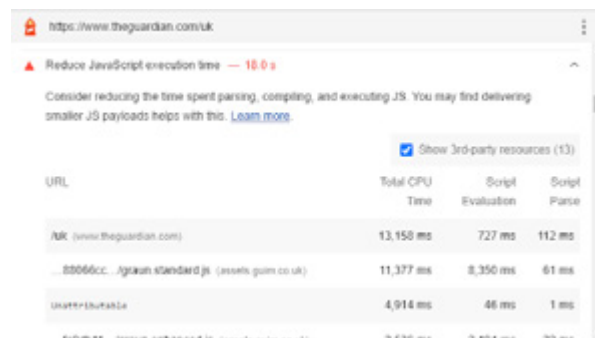
How to resolve the issue

As with the third party code problem, the answer here is to reduce the number of external calls you're making and to load anything you can asynchronously.

Reduce JavaScript Execution Time

This dovetails with both the Critical Request Chains and Third Party Code issues in that sites – especially legacy sites – will carry a lot of JavaScript, much of which will be carried by the <head> and will need to load fully before the site is useful for the user.

As a result, sites can end up with a long and growing time required to load the JS on the page.



URL	Total CPU Time	Script Evaluation	Script Parse
/uk (www.theguardian.com)	13,158 ms	727 ms	112 ms
...80066cc.../js/gaon-standard.js (assets.gum.co.uk)	11,377 ms	8,350 ms	61 ms
UNIDENTIFIABLE	4,914 ms	46 ms	1 ms
.../js/gaon-standard.js (assets.gum.co.uk)	9,630 ms	9,630 ms	99 ms

How to resolve the issue

The answer here is, again, to audit the JS on the site, remove anything which is not integral to it and to asynchronously load anything you can.

Counts and transfer sizes

This is an interesting one – not for the issue itself exactly, but for one of the tools for tackling it. Essentially this issue arises if the page is calling a lot of resources (images, video, widgets etc.) and the size of the resources is over large.

This touches on most of the other issues – if you fail against one or two of the other metrics, you'll likely not pass this test.



Resource Type	Requests	Transfer Size
Total	97	1,608.7 KB
Image	34	1,206.4 KB
Document	3	153.8 KB
Other	24	104.6 KB
Font	10	97.5 KB

How to resolve the issue

The recommendations stated elsewhere will aid with this one, too – but there's additional advice offered in the 'Learn more' link – to [create a budget](#). By installing LightWallet and creating a simple JSON file (java script object notation) with a budget of load times, you can have Lighthouse report deliver feedback on specific resource performance issues. The advice in the previous link is nice and easy to follow and should offer a nice additional report for webmasters.

Largest Contentful Paint element

The LCP element is, simply, the automatically detected resource which, once it has finished loading, can be said to indicate the completion of the page load. Therefore, regardless of performance elsewhere, if you have a particularly massive resource as one of the main features of your page, you may fail against this metric regardless.



How to resolve the issue

Obviously, this will depend on what the resource is that is taking the most time, but the general advice for load times will be to compress where possible, delete where necessary and to always ensure that file sizes and dimensions are tailored to their place on a page. Resizing takes time, buffering takes time, so attempt to reduce the need for extraneous work on behalf of the browser.

Large layout shifts

As you may guess from the heading and description in the image, this is one of the main measurements that feed into your score for the CLS metric.

This, too, is a measure that fits nicely with its predecessor here – we discussed the CLS and what it means, and the reason for it is often loading the secondary characteristics of a resource – resizing etc. as mentioned previously.



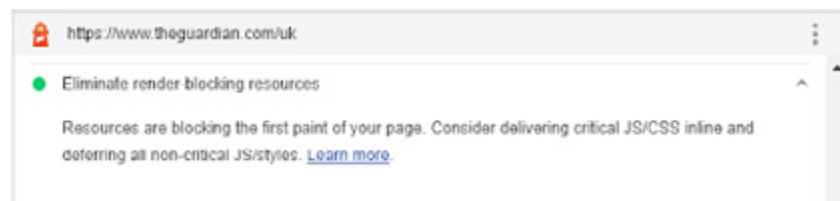
How to resolve the issue

The answer is, again, to ensure that you're correctly using the right sizes and positions for resources. Not only is a layout shift a negative as far as UX is concerned, but it also impacts a growing ranking signal. So make the effort to properly size your divs and tables, your images and videos and make sure that your pages reach their final layout as quickly as possible.

Eliminate render-blocking resources

This is one that brands can fall foul of for no other reason than CSS and JS files are just easier to maintain than inline styling. If you're using multiple fonts and styles to present your page, then you may find yourself impacting your score for this measure.

Again, you want your page to achieve its final form as fast as possible, and if you're making multiple external calls for fonts and style sheets, then you'll hurt rather than help your page performance.

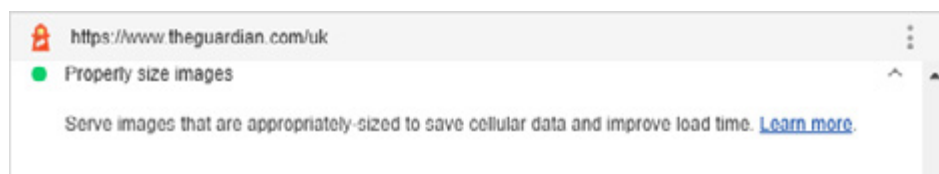


How to resolve the issue

Advice previously to cache CSS and JS files will help with this enormously – while it may not go as far as complete inline styling, the reduction in time will almost always be enough.

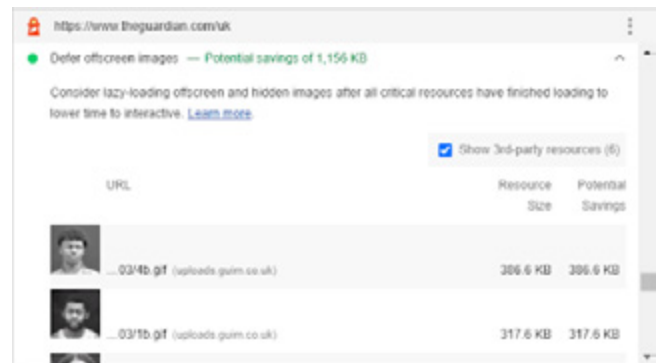
Properly size images

The cause of and solution to this one (and many of the later tests in this sequence) are fairly self evident, so we're going to break from the format for these. The cause of a fail in this is generally when sites use over large images to prevent blurring or pixilation. While providing high quality images should be the target, correct sizing (using 600 x 300 for a gap that size rather than 1200 x 600 to improve quality) should always be followed.



Defer offscreen images

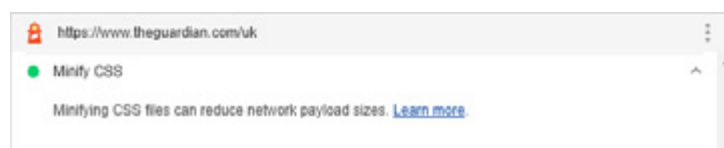
Another problem with the solution in the title – offscreen images, or images below the fold, can take up valuable load time. That doesn't need to be the case, however, and lazy loading is now a native attribute with Google, meaning that there is no real coding required to implement it – instead, simply adding `` is now supported by most Chromium powered browsers.



In addition to this, the latest version of WordPress – which powers a large proportion of all websites – includes this as a default setting, so if you haven't upgraded to at least version 5.5, then do so now to solve this issue.

Minify CSS

Minification is simply the process of removing line breaks and additional spacing from your CSS files – parsers will treat each character as worthy of inspection, even if that is simply a blank space. While for most sites this won't make a tremendous difference to the size of a CSS file (especially if we keep on top of the next point), it makes sure that we are only using as much space as we really need which is a net positive (even if it makes it harder for us to peek at your styling when we see something nice).



Remove unused CSS and JavaScript

If you have a legacy site, have undertaken a rebrand or any other design shift that didn't include a specific rebuild of the website, then the chances are you have chunks of styling and script that the site no longer requires.

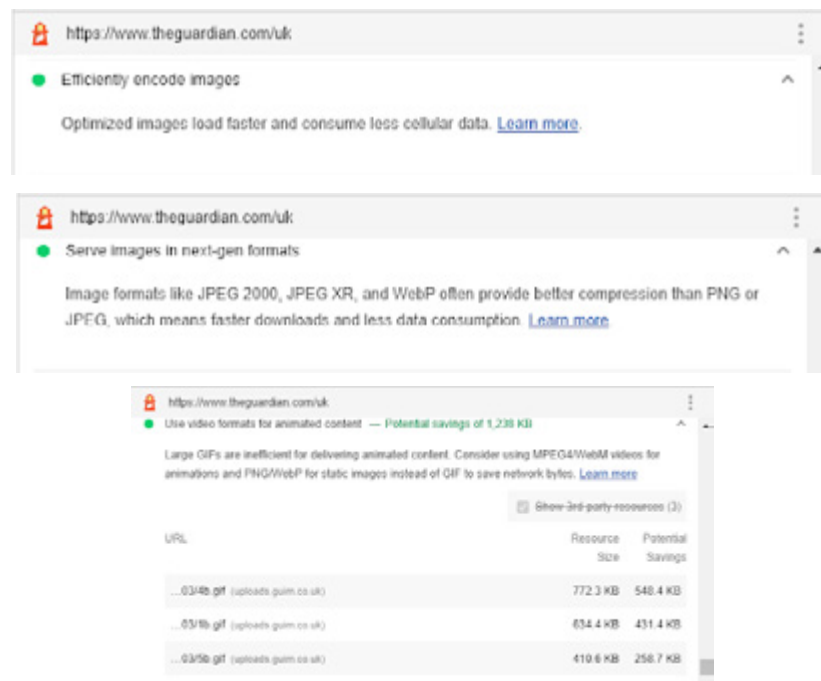


There are ways to check this using the coverage report in Chrome Dev tools, which will trigger each section of code necessary for the page and report any unused elements. Removing it is slightly trickier – but there are tools available that can automate the process.



Efficiently encode images and serve in next-gen formats

Encoding here is shorthand here for reducing load times – this can mean using CDNs for images, or using compression as well as using videos in place of gifs, lazy loading, responsive images and more. The threshold set by Lighthouse is level 85 compression which, if it delivers a saving of 4KiB will cause the tool to flag the image.

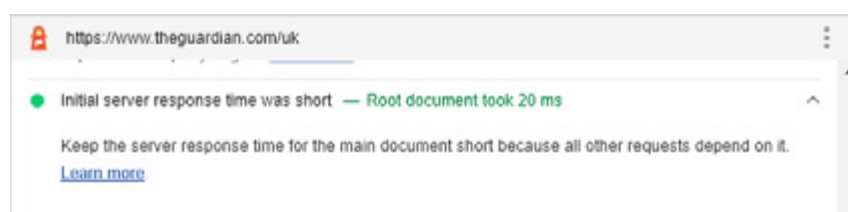


Similarly, serving next-gen formats feeds in to this, with the JPEG2000, JPEG XR and WebP file types capable of greater compression without quality loss and are, therefore, far faster than standard PNG and JPEG files.

Using video for animated content is also one of the measures which fits with the encoding and files message.

Server response time

More of a procurement issue than a web development or SEO issue, but having a potentially large impact regardless, your server needs to be able to rapidly process the requests made of it. There is enormous competition in this space, however, so brands really need to be ensuring that they're getting the service they're paying for or move on to another provider.



Avoid multiple page redirects

Another issue that compounds over time, redirect chains also add time to a load by forcing the browser to go the long way around to your page. For that reason, it's important to make sure that if your site moves or site structure changes, you can end up redirecting page that had already been redirected. You need to ensure that you're updating your URLs on third party sites and your directory entries. You can also use tools such as Screaming Frog to detect redirect chains and look to cut out the chain by redirecting each link in it to the right page.



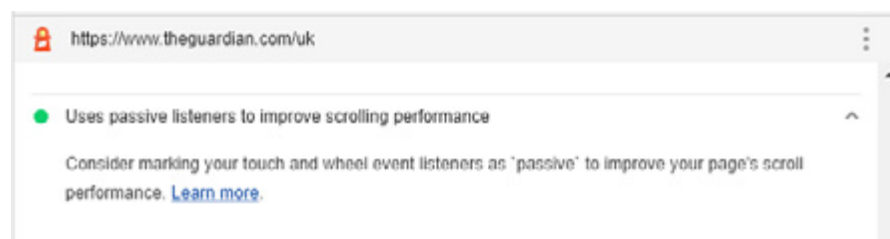
Use HTTP/2

Requiring HTTPS to qualify, HTTP/2 delivers content far faster than its predecessor. By now this shouldn't be an issue for most brands, but if you are coming up against a red flag here, then this should be a real priority.



Use passive listeners

Listeners to various user events will sometimes prevent scrolling on a page – which will wait to complete loading until the listener has loaded. Using passive listeners communicates to the page that the listener will never stop scrolling – allowing the page to continue loading without having to pause.



CONCLUSION

Speed may not currently be the most important ranking signal, but it is growing in importance – but more importantly than ranking (if we're allowed to speak the unspeakable) is the fact that many of the improvements we can make to a site's speed have the dual benefit of being beneficial to the user. For that reason, not only does the performance report we've just walked through impact site speed, it can serve as a conversion rate optimisation task – improving the user experience of a site in order to improve outcomes.

One thing is for sure, there are few confirmed ranking signals which are better correlated to improving conversions on a website and, in an industry with very few definitive statements it is well worth ensuring that we make the most of the areas in which we can act with a high level of confidence.

There are much more in-depth articles on each individual issue on Google's [Web.Dev](https://web.dev) site if you want to deep dive on any particular issue.



GET IN TOUCH WITH US TODAY

ABOUT US

Click Consult is a multi award-winning digital and search marketing agency with a focus on organic (SEO) and paid search (PPC). Part of global consumer brand business Ceuta Group, we have a team of 70-plus specialists and a portfolio of more than 60 clients worldwide.

Our complementary services include content marketing, outreach, social media, conversion rate optimisation (CRO) and international/multilingual search marketing. We can also offer training and consultation to support your teams or existing strategy.

Click was named Organic Search (SEO) Team of the Year 2020, adding to our long list of other awards and accolades, and also ranks within Econsultancy's 'Top 100 Digital Agencies', and Prolific North's 'Top 50 Digital Agencies'. We're also a Google Premier Partner, a Bing Select Partner and feature in The Drum Recommends.

HOW CAN WE HELP YOU?

CONTACT US



Follow us on [Facebook](#), [Twitter](#) or [LinkedIn](#), or to find out what Click Consult can do for you, call:

0845 205 0292

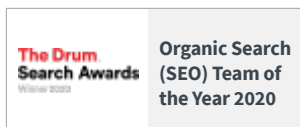
OUR ACCREDITATIONS



**The Drum
Recommended**



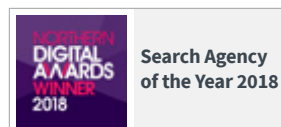
OUR AWARDS



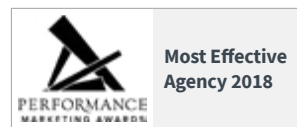
**Organic Search
(SEO) Team of
the Year 2020**



**SEO Agency of
the Year 2019**



**Search Agency
of the Year 2018**



**Most Effective
Agency 2018**

OUR TECHNOLOGIES

monitor **TRAX** rank **TRAX** feed **TRAX** link **TRAX** pro **TRAX** page **TRAX**