Google’s Lighthouse is now recommending JavaScript library alternates

Using the notion of a lighthouse as a metaphor, Google Lighthouse steers developers away from the rocks by shining light at issues it discovers on an asset-by-asset basis. With specific feedback for performance and security improvements, reports include references to media that could use resizing with compression, new or different cache policies, and linked files that contain blocks of unused CSS and/or JavaScript.

Until September, however, Google wasn’t calling out JavaScript libraries themselves.

> *If JavaScript libraries are proving costly, replace them with smaller alternatives.*
>
> *Lighthouse in @ChromeDevTools now recommends smaller libraries that improve bundle size. pic.twitter.com/VFe8TFV9Y5*
>
> *— Addy Osmani (@addyosmani) September 12, 2020*

And now, the warnings have graduated to appear in Page Speed Insights.

A word about JavaScript

In the open-source JavaScript world, developers stand on the shoulders of the developers who came before them, especially those who solved something that otherwise would have to be solved by a new library author. The prevalent JavaScript packaging system NPM (Node Package Manager) eases the inclusion of preexisting libraries in your project. At the starting point, a given JavaScript project is always the tip of an iceberg made of much more JavaScript underneath, usually stored by NPM in the .node_modules directory.

It stands to reason that projects, especially those created using a sophisticated framework, would only ever make use of a

tiny percent of the code available to it, largely through numerous library dependencies. That's why there's an optimization process known as "tree shaking" for bundling only what's actively being used by a given project as much as possible. Tree shaking doesn't always work well with older libraries due to the moving goalposts of keeping up with modern syntax and coding patterns.

About Frameworks

Frameworks make life easier for developers by removing the complexity of making your library choices, configuring bundlers, and writing scripts to automate optimization processes for production. With a "quick start" recipe found in most documentation, developers can get up and running prewritten command line interpreter scripts that come as part of most packaged frameworks. An example of this is the Create React App which scaffolds up blank React application code ready for you to develop further into a web application.

```
$ npx create-react-app my-app
```

Running the above command creates a "my-app" directory and spawns the entire React app directory tree into it, complete with all the required files and library dependencies. Optimizing your production bundle, which can contain code from several Javascript libraries, is perhaps the most important reason modern framework packages have tools and steps that weed out unused blocks of code and minimize output for production.

When you choose to use a framework you accept, perhaps without knowing all the details, the difficult decisions of architecture, configuration, and library dependencies of that framework. Here is the guide for optimizing React for production from the makers of the popular frontend library currently used by many projects and frameworks such as NextJS.

It’s all too common that a handful of older libraries (highly useful in their day) have found their way into project bundles as dependencies. Websites, libraries, and frameworks authored prior to JavaScript milestones show their age when using deprecated code as JavaScript fundamentally progresses at such a breakneck pace. Lighthouse now serves as a beacon for warning you when your project includes old and or vulnerable code.

MomentJS

A notable library, MomentJS (with 12 million downloads per week as of September 2020), is the first one Lighthouse points out as having a few better options. Google’s logic here is irrefutable and rather well known. In response, Moment’s own homepage and documentation now mirrors the advice provided in Lighthouse’s reporting. Moment is feature frozen, with only stability updates planned.

Other libraries that Google has under the microscope are Lodash and possibly Underscore. There have been legitimate negative feelings expressed about this throughout the developer community, with more than one developer calling it “toxic” or “harmful” to the open-source community. The complaints have to do with Google “shaming” libraries without giving enough context and promoting alternatives that can harm the discovery of smaller and lesser-known library alternatives.

While all that is true, it’s also true that you have to break eggs in order to make an omelet. Progress often will hurt some folks. Google relies on a third-party reference (BundlePhobia) for collecting alternate library lists. They further vet choices based on a “high bar for equivalency” and “ease of migration” as determined by the Lighthouse team.

Lesser-known library authors can get in the mix by submitting their library to the service. Additionally, a developer

commented that since Google is now officially recommending libraries, they should help finance open-source by donating to the project and a team member on Twitter has agreed to begin doing this starting in 2021.

> *This is a really great idea. Fully agree these teams need support. Chrome currently has a yearly web performance and frameworks fund for sponsoring work like this. We've been working with OpenCollective to expand on this for 2021 and will have more funding to announce soon.*
>
> *— Addy Osmani (@addyosmani) September 12, 2020*

What this means for TechSEOs

It's true that TechSEO practitioners do not have to be developers in order to be excellent at their job. It's also true that the more familiar you are with the struggle developers undergo, perhaps by learning a little about coding and at least understanding the details of Google's Lighthouse as much as possible, the better you will be able to communicate problems and practical solutions to developers.

Replacing MomentJS wholesale may be anywhere from super easy, to frighteningly complex for the recipient of the news that it needs replacing. Unless you are a developer yourself, or have at least tried dabbling in a little web development using modern JavaScript libraries and frameworks, it will be difficult for you to know just how painful switching out a library like Moment for a smaller alternative may be for a particular project.

It has to do with how much that library has been integrated into the codebase. Code patterns may have to be completely rewritten throughout an application. The bigger and more interconnected the application codebase is, the harder it will be to accomplish the replacement. It isn't always as simple as

saying “Just use one of the alternate libraries that Google is telling you to use instead.”

One of the daunting thoughts that can occur when learning one has to replace Moment comes from the fact that Objects in the library are written to be mutable (changeable). Keeping this pattern is seen as necessary for backward compatibility and this really complicates replacing Moment with another library. Whole blocks of code may need to be written in order for your application to accommodate the fact that variable values assigned using Moment somewhere in a call chain can’t be counted on as immutable values inside your application codebase.

A modern pattern which relies on immutable Objects from a newer library is more stable. To get there can require a lot of rewriting every instance where Moment gets used.

SEO for Developers

Optimizing JavaScript for production as a task is very much in the wheelhouse of the TechSEO working on code, or providing feedback to a developer who may be unfamiliar with SEO, as developers should understand performance optimization in the first place. It can’t be presumed that developers know about BundlePhobia, Lighthouse, or SEO for that matter.

If you made it this far and wish to know more about coding in order to provide a better service to your clients, then you’re in luck. This year, we’re going to be producing the SEO for Developers Workshop as an optional add-on to the SMX conference series. The information presented will be targeted at guiding your journey from wherever you are as a TechSEO practitioner, to wherever our collective paths lead us into coding. Given how fast things are progressing the sky’s the limit!

About The Author

Detlef Johnson is the SEO for Developers Expert for Search Engine Land and SMX. He is also a member of the programming team for SMX events and writes the SEO for Developers series on Search Engine Land. Detlef is one of the original group of pioneering webmasters who established the professional SEO field more than 20 years ago. Since then he has worked for major search engine technology providers, managed programming and marketing teams for Chicago Tribune, and consulted for numerous entities including Fortune 500 companies. Detlef has a strong understanding of Technical SEO and a passion for Web programming.
http://feeds.searchengineland.com/~r/searchengineland/~3/Ldw7tfRq8Js/googles-lighthouse-is-now-recommending-javascript-library-alternates-344940