An SEO's survival guide to Single Page Applications (SPAs)

SEOs beware: if you haven't heard of Single Page Applications (SPA for short), or if you have been resistant to learning about these JavaScript methods for creating websites, the time for hiding your head in the sand is over.

Check out this tweet from Google Webmaster Trends Analyst John Mueller:

> *The web has moved from plain HTML – as an SEO you can embrace that. Learn from JS devs & share SEO knowledge with them. JS's not going away.*
>
> *– John .o(≧▽≦)o. (@JohnMu) August 8, 2017*

John Mueller is correct. *It's not going away.*

A quick search on Google Trends for "Single Page Application" reveals the sharp rise in popularity and awareness of SPAs over time:

Some developers are positively enamored with using JavaScript frameworks and libraries to create websites, and SPA popularity has been steadily growing.

Take Angular (also known as AngularJS and Angular.js), for example.

Here’s a Google Trends search for the Angular JavaScript framework showing the past 5 years, and Trends even recognizes the application platform – you can see popularity has increased greatly over the last couple of years:

The React JavaScript library shows a similar up and to the right trend:

In my role as a professional SEO, I can’t say that Single Page Applications are the rule and not the exception when it comes to how businesses choose to develop websites these days, but I am running across more and more SPAs, and so are my colleagues.

- Yes, it’s true that JavaScript was never intended for web page content delivery.
- Yes, it’s true that SPAs to date have not been great for SEO.
- Yes, it’s true that many developers who had fun quickly creating websites using SPAs had to later spend more time fixing SEO problems than the time they would have spent if they just coded the site to deliver content via

HTML5 in the first place.

But, none of that matters, my SEO friends.

Like it or not, SPAs look like they're here to stay.

It's time to stop thinking bad thoughts about SPAs and trying to wish them into the cornfield.

Single-Page Applications: Resistance is futile

I admit it – for a while there I was hoping I could ignore Single Page Applications, and maybe eventually SPAs would end up in the trash heap of obsolete website trends such as the <blink> tag, and web page content that's free of annoying and intrusive advertising interruptions.

Programming and coding languages live and die by developer adoption. For example, if, by some weird turn of events, developers across the world suddenly decided they hated PHP and fell in love with some super-cool new server-side scripting language, then PHP withers, maybe even dies.

It's just that simple.

That's why, for example, Google has been pushing Accelerated Mobile Pages (AMP) super hard – because they need major and widespread developer adoption for AMP to succeed and not wind up as the <blink> tags' roommate.

Talk to developers who've created sites using Angular, React, or other JavaScript frameworks or libraries. See if they light up as they talk about the ease and speed of development and how debugging was not as hard as the rumors have it.

SPAs are popular with developers, and that popularity is not showing any sign of slowing down.

Dipping a toe into the SPA

Looking “under the hood” of SPAs, a distinguishing characteristic is that there’s a lot less back and forth between the server and the browser making requests to the server.

After the initial JavaScript framework download to the browser and first page view, there is no page reloading going on when navigating to a second, third, and fourth (etc.) page, hence the “single page” part of Single Page Application.

After that initial JavaScript framework download and first page view, subsequent pages viewed load very quickly, exactly because of the lack of back and forth requests between the server and browser that “traditional” web sites require.

And this means a very good user experience because no extra page load means no extra wait time. And, as we all know, everyone prefers fast-loading pages.

The main aspect to remember here is that with an SPA there is far less back and forth between the browser and the server.

But JavaScript was never intended for web page content delivery

Before JavaScript started being used commonly in website development, web pages were static and created using HTML.

Using JavaScript enabled website developers to add interactivity to their web pages such as pop-up dialog boxes when a user is filling out a form, expandable content when a user clicks on text or a button, or a drop-down menu when the user hovers their mouse over a navigation element.

These and other user interactive features JavaScript allows can be executed in the browser without requiring a call to the

server.

And thus, for many years, website developers used HTML for content delivery, CSS for layout and styling, and JavaScript for adding user interactivity.

It's a fair generalization that JavaScript has become vital to websites and to a developer's resume; JavaScript is pretty much ubiquitous. JavaScript is not that difficult to learn compared to full-blown programming languages such as Java and C++. The "J" in AJAX and jQuery is – you guessed it – JavaScript.

I only bring this up because in retrospect, and hindsight is always 20/20, we SEOs all should have seen the rise of Single Page Applications looming on the horizon.

But viewed glass-half-full, the rise of SPAs presents an opportunity for technically-minded SEOs to gain experience and become even more valuable now and in the future.

If SPAs can cause SEO issues, then why do developers create SPA websites?

If you've never done any coding, then you might not realize what it's like to be in a developer's mindset.

Think about it this way: if you were going to have to sit down and write code to create a certain web page functionality and you could either write 10 lines of code to achieve that, or write 1,000 lines of code, which would you choose? You'd opt for the expedience of 10 lines of code, right?

Developers are not lazy; they simply prefer efficiency and elegance when it comes to writing code. I've seen developers frame code and hang it on their office wall. Ever heard the

saying “code is poetry?”

If you’re trying to get somewhere the fastest way possible, you take the shortest route, correct?

Single Page Application frameworks and libraries, in crude summary, provide building blocks that allow developers to create a website quickly and efficiently.

Consider the fact that SPAs allow developers to efficiently create modern-looking websites that load pages quickly, which makes for a great user experience, and you can see why you might choose an SPA over coding a website from scratch in HTML, CSS, and JavaScript, or hassling with the constraints of a Content Management System.

SPAs present a fast-loading user experience because they don’t need to reload most resources such as HTML, CSS, and scripts with each user interaction like a “traditional” website does. These files only require initial loading and then after that only new data is retrieved and downloaded from the server.

SPAs reduce response times primarily by moving the heavy-lifting of data processing from the server to the browser.

SEO may be a lesser consideration given the SPA developments upsides, an afterthought, or perhaps not a consideration at all during the website development process. Any SEO pro who has been in digital marketing for very long has seen the all-too-common situation where a company develops a website, only later to ask the question “how do we SEO this thing?”

Not everyone realizes that SEO should be baked-in at the beginning and not sprinkled-on at the end, or that their website development choices can have definite downstream negative impacts with respect to SEO.

Share this article

Related articles

JavaScript libraries vs. JavaScript frameworks

Untangling the technology behind SPAs eventually leads us to the topic of JavaScript libraries and frameworks.

Ask a developer *“what’s the difference between a library and a framework”* and you’ll get a lot of interesting answers.

One overriding distinction you hear repeatedly goes something like this:

The code you write calls a library, whereas a framework calls the code you write.

React and Angular are both SPAs, but React is technically a library, whereas Angular is technically a framework. However, you will hear often people refer to SPA technology generally as “JavaScript frameworks”.

Frameworks can be thought of as a structure, like a pre-fab home which comes with the framing, drywall, plumbing, and electrical wiring and all you have to do is add the appliances, windows and coverings, flooring, paint, etc.

A library can be thought of as a place that contains a set of ready to use pre-built tools and functionalities. You’d call a library in your code for a specific function.

You can see that starting a web development project using frameworks and/or libraries can streamline the process, as opposed to writing from scratch all the necessary code to create a website.

Common SEO problems of Single Page Applications

There’s a lot of talk about how well Google can handle JavaScript when it comes to crawling and indexation.

Crawling and indexing is critical to ranking.

Google discovers web pages using software called Googlebot during a very fast process often called “crawling” or “spidering”, during which it downloads an HTML file it finds, extracts the links and visits them simultaneously, and then sends the downloaded resources to the indexer.

But when it comes to a JavaScript-based single page application website, the process gets a bit more complicated.

It’s like the process noted above, but there’s a delay and extra step involved because part of the indexer must do some heavy lifting by parsing and executing the JavaScript, and the new links found then must be passed back to the crawler to look at and then sent back to the indexer; you can see that this is less efficient because of the JavaScript.

SEO is more than just having “great content” and earning high-quality links; it’s also about making your web pages easy to discover by search engines like Google and making it simple for them to know which pages are more important than other pages via internal linking.

A “traditional” HTML-based site is far easier to crawl and index, and by extension, rank. Google can get all the links easily and see what the importance of pages are via internal linking.

A JavaScript-based SPA website makes Google’s life more difficult, and some testing would seem to indicate that there may be downsides when relying on JavaScript for purposes of

indexation.

Google is evidently willing to do the extra heavy lifting here, and to my mind that indicates that they'll improve over time rather than announce to webmasters in the future that they have decided they don't want to bother with the extra work required to crawl and index JavaScript-based sites.

Another potential SEO problem related to the extra work to discover links is that Google may have issues with evaluating the link equity of those pages.

It's likely that in time, at least some of the SPA frameworks in popular use will evolve the rendering process to make it easier for Google to crawl and index, perhaps even making it on par with "traditional" HTML-based websites.

But in the meantime, we're where we are and those who've tested how well Google can handle JavaScript-based sites have shown that Google's ability is inconsistent, and we're also still in a place where those who have developed SPAs frequently must use workarounds, for example using prerender.io along with Angular to serve fully-rendered pages to the crawler.

Another solution is isomorphic JavaScript, sometimes called "Universal JavaScript", where a page can be generated on the server and sent to the browser, which can immediately render and display the page. This solves the SEO issues as Google doesn't have to execute and render the JavaScript in the indexer.

Headless Chrome is another option recently proposed as an easy solution by a Google engineer, who also mentions another solution called Preact, which ships with server-side rendering.

It's also a good idea to create a properly formatted XML Sitemap and submit that to Google Search Console.

Right now, there doesn’t appear to be any single solution or a paint-by-numbers approach to handing the problems you may encounter if you’re an SEO assisting a client with launching or redeveloping a website using an SPA.

It boils down to effectively communicating the correct end result that’s needed, and dealing with issues as they’re presented based on the library or framework being deployed.

Some important Single Page Application resources

Some super-sharp SEOs and developers have written helpful articles about Single Page Applications, and here are a few resources I have enjoyed that I think you will find helpful:

Final thoughts

Single Page Applications are evolving rapidly, as is the web technology landscape in general. It’s worth the effort for professional SEOs to be as conversant as possible with not only Single Page Applications, but also Accelerated Mobile Pages, Progressive Web Apps, Content Management Systems in general, and of course the tech behind how websites are coded from scratch.

My sense of the situation is that SPAs, and Google’s ability to handle JavaScript-based websites, will advance at a quickening pace because the stakeholders involved are aware that SPAs come with a definite SEO downside as it stands right now.

It’s entirely possible that in a year or so the most popular SPAs will ship with SEO solutions built in because awareness of the need for SEO friendly JavaScript-based websites is growing. But there’s no guarantee of that happening soon or at all, so my recommendation for today’s SEOs is to get excited

about and embrace this technology trend.

David Portney is Senior SEO Manager at 3Q Digital

Want to stay on top of the latest search trends?

Get top insights and news from our search experts.

Related reading

Artificial intelligence (AI) is a field of technology

that is surrounded by both hyped and misconceptions. It is predicted that $60 billio

n
will be spent by brands on AI technology by 2025, so this hype is having a direct impa

ct on where companies allocate their budgets.

There are many instances where

a website might be required to change its domain name. For example, your website

might be penalized heavily by Google, and you want a new domain, or you might

want to rebrand your business. In this article, we will focus on the tools and tech

niquesthatwillallowyoutomigrateWordPresswebsitedomainnamewithminimal

impact on SEO.

There’s been much talk recently about Google implementing a broad

core algorithm update. Google has suggested that this update has nothing to do wi

th
th
e
qu
al
it
y
of
co
nt
en
t,
an
d
in
st
ea
d
fo
cu
se
s
on
im
pr
ov
in
g
th
e
qu
al
it
y
of
th
e

SERPs. However, if Google wasn’t testing quality, what exactly were they testing? Wh

o
be
ne
fi
te
d
fr
om
th
e
up
da
te
,
an
d
wh
at
ca
n
we
le
ar
n
fr
om
it
?

Fo
r
ye
ar
s,
sh
or
t-

tail and transactional keywords were the primary focus of SEO campaigns. Howeve

r, there is a clear acknowledgement on both sides of the fence with regards to th

e
po
we
r
of
lo
ng
-
ta
il
ke
yw
or
ds
.
So
ho
w
ca
n
we
ad
ap
t
ke
yw
or
d
re
se
ar
ch
in
re
la
ti
on

to this focus on identifying great long-tail keywords?

https://searchenginewatch.com/2018/04/09/an-seos-survival-guide-to-single-page-applications-spas/