Technical SEO Best Practices

by Ian Lurie

PORTENT

Legal Stuff

Reduce Ulcers: Read This Document

When they read this story, Portent's clients will say "Ian, is that me?" Don't feel bad. It's you. It's also every site my team's worked on in the last 15 years:

Your new site is ready to go. You're justifiably proud of it. It kicks butt. It looks great. It works well. It's got all manner of useful, built-in tools. The CMS is flawless. Then, you turn it over to the SEO team for a review.

Then the wheels come off.

Suddenly, you've got a laundry list of changes you have to make to page templates, server configuration, and editable page elements.

That SEO team is the pain in the rear.

These changes will take hours. You have to launch tomorrow. You do what you can and put the rest on the list for another day. You dread meetings with the marketing and SEO folks. You've got one more source of those daily "when will this be done?" e-mails.

Eventually, the CEO searches Google for some relevant phrase. Your site doesn't rank. She yells at the marketing team. The marketing team points at you…

You can avoid all this stress. Before you start your project, read this document. Every time you update staging, read it again.

More important: Use it as your requirements doc. Before you start work, show it to your SEO expert(s). Make sure it covers everything they're worried about. Get them to sign off on it. If they add stuff, OK—now you know. If they don't, great! Either way, you can build an SEO-ready site the first time and skip the panicky fixes twelve hours before launch.

Preface: Don’t Skip This!!!

The first time you read this guide, read the preface!!! It’s got important stuff. “Preface” doesn’t mean “ignore me.”

That’s like ignoring The Hobbit because it comes before Lord of the Rings. Not the movies. The book. We don’t speak of The Hobbit movies here at Portent.

Who This Is For

Developers. Developers developers developers…

I had to say it once. Sorry, Steve.

This ebook is for the people who build websites, and the SEO teams they work with. It assumes you’ve got basic dev skills: You understand how a web page works, what a server response code is, and the difference between client- and server-side.

If you don’t, that’s OK! Take this and forward it to your dev team. Walk through it with them. Then go learn some of this stuff. Developers have to learn to work in your world. You should think about learning a bit about theirs.

How This Is Organized

SEO factors group like this:

- Visibility
- Authority
- User Experience
- Relevance

I’m betting that’s not your workflow, though. So, the first part of this guide talks about SEO factors and tries to tie them to phases of site development.

The bulk of the guide works by position in the technology stack and when you might work on them. I don’t know your project, though, so be sure to read the whole thing, first.

A Note About Google Announcements

When I’m doing development work, I hate ambiguity.

Google, though, often introduces ambiguity when they make pronouncements like, “There is no duplicate content penalty.”

They’re always **technically** accurate. For example: You don’t get penalized for duplicate content. You do, however, waste crawl budget, split link votes, lower site quality if pages are “thin,” and hurt your rankings. Penalty? No. Bad? Yep.

Another recent favorite: A Google engineer said they treat 302 and 301 redirects the same for rankings purposes. A few days later, they backtracked. A few days after that, the engineer threw up his hands in frustration and said, “Just use the right one at the right time!”

Which, of course, is exactly where we started.

My favorite: Google says they can crawl javascript just fine. Which they (sort of) can. What they **can’t** do is index all client-side javascript-delivered content. What they **won’t** do is pay attention to any client-side content that takes longer than a few seconds to deliver. What they **will** do is de-prioritize content that’s not visible when the page first loads. That all came out later on.

Take Google’s statements with a healthy grain of salt. When in doubt, think about what makes sense: 301 and 302 redirects aren’t the same. Use them appropriately. Then, when a Google engineer tries to explain how Google treats them, you don’t have to scramble to figure it out.

How Technology Affects SEO 1

Developers learn that SEO is about stuffing keywords into the right place or putting links in the footer, or that it's a complete fiction.

Unfortunately, none are true. **Technology, from infrastructure to HTML code, have more impact on rankings than any other single factor.**

What?!

If you're an SEO, you probably read that and are already booking a flight to Seattle to slap me. While you do that, you're saying, "Ian, you idiot. Links are most important! Title tags are most important! Technology? I don't think so."

Build all the links you want. If your site takes 10 seconds to load, or search engines can't find half your content, you're still not going to rank.

Write all the keyword-stuffed crap you want. If search engines get bound up in endlessly expanding URLs and duplicate content, you're *still not going to rank*.

I’m not saying this as a nerd. I was a writer long before I put on my propeller hat. SEO is about technology first. Then it’s about architecture. Then it’s about links and content.

Visibility

Can search engines find, crawl and index your site? Redirects can tell a search spider to ignore you. A javascript framework may hide content or generate duplication. An incorrect robots.txt file can completely shut you down. Getting the infrastructure right ensures visibility. Technology makes that happen.

This guide shows you how to maximize visibility through:

- Correct javascript framework implementation
- Response codes
- Canonicalization
- Correct use of subdomains and subfolders
- Sitemaps

User Experience

Google and Bing both look at site speed, user interface, mobile usability and a long list of little things that are the hallmarks of a good user experience. Technology influences all of those things.

Google and Bing may also look at behavior, like “bounce back” to search results pages. This guide deals with the technical details that impact user experience:

- Site performance
- Page rendering
- Content visibility on the page

Relevance

Yes, keywords do matter. As long as we all talk and write, we'll search using words. The right words in the right places drive relevance. Site and server setup make those words possible. Technology helps content creators ensure relevance. This guide talks about the stuff you manage that affects relevance:

- What's editable and what's not
- Structured data and markup
- Linking and site structure
- Page structure and elements

Authority

Yes, links matter a lot. As a developer, you need to ensure the site implementation doesn't hurt authority by reducing link value and validity. Consider:

- New and old URLs
- Redirection
- Response codes
- Canonicalization

Foundational Stuff

A few crucial factors don't fit into the Big Four, but handling them now will mean less work later:

- Server logging
- Analytics
- Tag managers

Laying the Foundation 2

Before you set up a repository or download a framework, do these things:

Server Logging

Most servers are set up correctly out of the box, but just in case, make sure:

- The server logs the referring domain and URL, date, time, response code, user agent, requested resource, file size and request type. That's the minimum. IP address helps, too.
- The server logs relevant errors
- The server won't delete log files. Archive them instead. If they're gigantic, delete them after a year, or make the SEO team pay for an Amazon Glacier account. Or the equivalent. What matters is that they pay for it

Analytics

Ensure the right analytics tags are on every page, and throughout any checkout or other funnels. I know: Duh. This seems like one of those things authors put into guides to pad the keyword count.

Not I. You need to think about:

- Cross-domain and subdomain tracking
- Multiple accounts
- IP filtering
- Funnel tracking
- Page event tracking
- On-page tag location: Top of the page slows things down, bottom of the page means you may miss bounces
- Onsite search logging

Told ya.

Tag Managers

I love and hate tag managers. They're great because they let marketing teams add and edit third party widgets, JSON-LD for structured markup, and analytics code.

They're also horrible because they let marketing teams add and edit third party widgets, JSON-LD for structured markup, and analytics code.

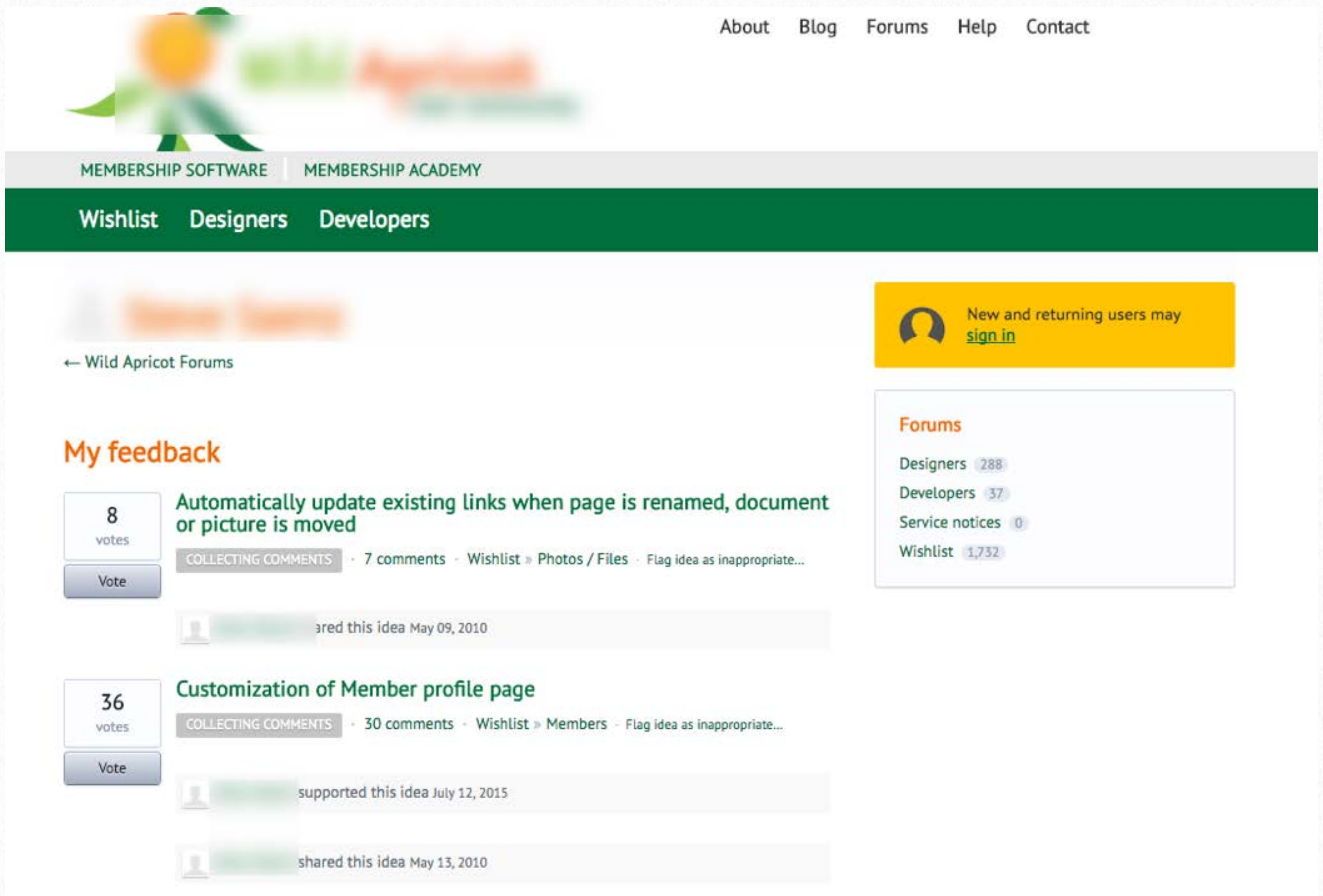
My recommendation: Find out exactly what they want to do. Figure out if they're going to break the site or kill performance. Provide clear guidelines. Be ready to fix things when they break the site or kill performance.

"Thin" Content

I'm not sure this is a foundational issue. It's hard to fix later on, though, so think about it now.

Google and Bing do not like sites where a large percentage of pages (or a large number) have very little content. If that content's the same, even worse. They consider that "thin" content.

The most common example is a site that has some kind of member profile page. If you have many members, and none of them fill out their profile, you have many near-duplicate pages with little or no content.



[redacted]

← Wild Apricot Forums

New and returning users may sign in

Forums
Designers 288
Developers 37
Service notices 0
Wishlist 1,732

My feedback

8 votes Vote
Automatically update existing links when page is renamed, document or picture is moved
COLLECTING COMMENTS · 7 comments · Wishlist » Photos / Files · Flag idea as inappropriate...
[redacted] ared this idea May 09, 2010

36 votes Vote
Customization of Member profile page
COLLECTING COMMENTS · 30 comments · Wishlist » Members · Flag idea as inappropriate...
[redacted] supported this idea July 12, 2015
[redacted] shared this idea May 13, 2010

This is a classic "thin content" page. This site has hundreds of members.
Their pages have, at most, one or two "feedback" listings. That's thin content.

If you think your site might have these, do one of three things:

- Don't let people create empty profiles. I don't recommend this option, but if you must, it's available
- Use robots.txt to disallow all profile pages
- Have your CMS check for profile length, and if it's zero, add a meta robots noindex tag to the page

Upload sites like public photo galleries and review sites are susceptible to thin content. So are sites that have any kind of "e-mail this to a friend" forms, if the forms are separate pages.

Response Codes

Response codes tell Google and Bing how to treat a page. Use the right ones:

200	The resource is OK. Google and Bing will index it
404	The resource doesn't exist. Google and Bing ignore it. 410 is another version of this that, after much debate in the SEO world, appears to have the same impact
301	The resource is gone forever, and there's a new one. Google and Bing will index the new page and pass or link authority to it
302	The resource is temporarily gone. Google and Bing will return later to see if it's back. They won't index the other page or pass link authority to it
501	The server had a problem. They ignore the resource for now

Most response code problems I've seen are on .NET-based content management systems and sites. I don't know why, but out of the box they're configured to serve a 302 redirect when a visiting browser attempts to load a missing resource. Instead of a 404 response, the server delivers 302 redirect then sends the visitor to a page with a “page not found” message. That’s bad.

Even worse, some .NET servers come configured to deliver a 200 response for broken links and missing resources. The visiting web browser tries to load a missing page. Instead of a 404 response, the server delivers a 200 “OK” response and keeps you on that page. That page then displays a “page not found” message. Crawlers then index every instance of that message, creating massive duplication.

That’s a canonical issue I talk about in the next section.

Canonicalization 3

Nothing trashes site indexing like duplicate content. Nothing creates duplicate content faster than canonicalization issues. Every resource on your site should have a single valid address. One. Address.

Canonicalization problems can cause duplicate content that in turn wastes crawl budget, reduces authority and hurts relevance.

Don’t take my word for it. Read Google’s opinion:

https://support.google.com/webmasters/answer/139066?hl=en

If you follow these recommendations, you’ll avoid 90% of canonicalization problems:

Home Page Has a Single URL

Ensure your home page is “/”.

- It shouldn’t be www.foo.com/index.html, www.foo.com/default.aspx, /index.php or any-thing else
- It should be www.foo.com
- www.foo.com is canonically different from foo.com. Use one or the other
- If you’re using SSL, redirect from all “http” addresses directly to the canonically correct https home page

Don’t depend on rel=canonical or 301 redirects for this. Make sure all internal site links point to the same canonical home page address. No site should ever require a 301 redi-rect from internal links. You might need to redirect clicks on external links—it’s amazing how many ways people find to link to your home page. You control internal home page links. Make sure they’re canonically consistent.

There’s an exception to every rule. Some sites *supposedly* require redirection to something like /foo/bar/home.do. I’ve never, in 20 years, seen an instance of this that couldn’t be fixed. Anything’s possible, I guess.

Pagination Has One Start Page

Make sure that the link to page one of a pagination tunnel always links to the untagged URL.

For example: If you have paginated content that starts at /tag/foo.html, make sure that clicking “1” in the pagination links takes me back to /tag/foo.html, not /tag/foo.html?page=1.

No Hard-Coded Relative Links

Friends don't let friends create links like this:

```
<a href="~">
```

Those can create infinitely-expanding URLs:

```
/en-us/
/en-US/US-Distribution
/en-US/~/link.aspx?_id=6F0F84644AC94212ACA891D5AE1868C9&_z=z
/en-US/~/~/link.aspx?_id=B682300BEAD24C0ABC268DB377B1D5A0&_z=z
/en-US/~/~/~/link.aspx?_id=6F0F84644AC94212ACA891D5AE1868C9&_z=z
/en-US/~/~/~/~/link.aspx?_id=B682300BEAD24C0ABC268DB377B1D5A0&_z=z
/en-US/~/~/~/~/~/link.aspx?_id=6F0F84644AC94212ACA891D5AE1868C9&_z=z
/en-US/~/~/~/~/~/~/link.aspx?_id=B682300BEAD24C0ABC268DB377B1D5A0&_z=z
/en-US/~/~/~/~/~/~/~/link.aspx?_id=6F0F84644AC94212ACA891D5AE1868C9&_z=z
/en-US/~/~/~/~/~/~/~/~/link.aspx?_id=B682300BEAD24C0ABC268DB377B1D5A0&_z=z
```

Never hard-code relative links, unless you want to be comic relief in an SEO presentation.

No Query Attributes For Analytics

Don't use query attributes to tag and track navigation.

Say you have three different links to /foo.html. You want to track which links get clicked. It's tempting to add "?loc=[value]" to each link. Then you can look for that attribute in your analytics reports and figure out which links get clicked most.

You don't need to do that. Instead, use a tool like Hotjar. It records where people click, then generates scroll, click and heat maps of your page.

If you absolutely must use tags, then use an # instead of a ? and change your analytics software to interpret that, so that ?loc=[value] becomes #loc=[value]. Web crawlers ignore everything after the hash sign.

Use the Right Response Codes

Response codes tell visiting browsers and bots how to treat pages. I already talked about this, but it bears repeating: *Response codes are really important*.

- If a search engine bot gets a 200 response, it will try to index that page
- If it gets a 301 response, it will index the page to which it's redirected
- If it gets a 302 response, it will check for that page later
- If it gets a 404 (or 410) response, it stops attempting to index

The wrong response code can create massive duplication. If your server, say, returns a 200 response for broken links or missing resources, search engines will index the "page not found" message for every broken link on the site. That's a lot of "oops" messages showing up in the search results.

Things to Do

Whether you have canonicalization issues or not, make sure you:

- Set the preferred domain in Google Search Console and Bing Webmaster Tools (last time I checked, you could do this in both)
- Set rel=canonical for all pages. Might as well handle it ahead of time
- Set the canonical http header link

Quick fixes

It's best to fix canonicalization issues by doing it right: Build your site to have a single address for every page.

If you can't do that, though, use these:

- rel=canonical points search engines at the preferred page. It doesn't fix crawl budget issues, but it's something. Make sure you use it right! Incorrect rel=canonical setups can hurt more than help
- Use the URL Parameters tool in Google Search Console to filter out parameters that cause duplication. Be careful. This tool is fraught with peril

NEVER

Never ever under any circumstances:

- Use robots.txt or meta robots to hide duplicate content. This completely screws up the site's link structure, doesn't hide the content, and costs you authority
- Point rel=canonical for one set of duplicates at different target pages
- Use either Google Search Console or Bing Webmaster Tools to remove the URLs of duplicate pages

In other words, no funny business.

Editable Elements 4

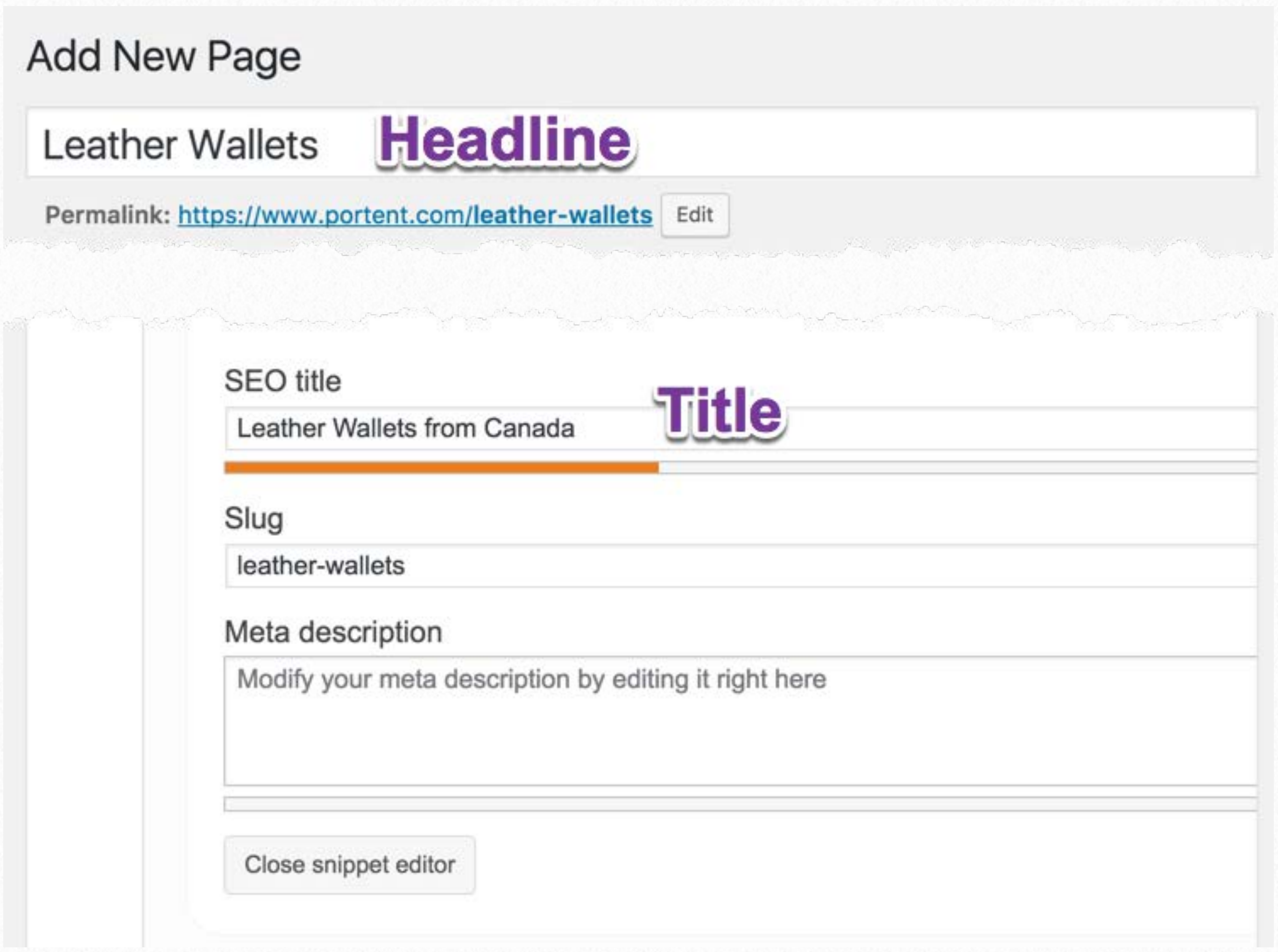
SEO nerds like me make many little changes to every page on a site. We observe the result. Then we change stuff again. Unless you want me e-mailing you every half-hour, make sure these things are editable:

Have One, Editable Title on Each Page

The title element is the strongest on-page ranking signal. There must be one `<title></title>` element on each page.

Make it a separate, editable field. If I write a page with the headline “Leather Wallets,” I might want to make the title tag “Leather Wallets from Canada.”

Have the title element default to the page headline, but make it separately editable. In the example below, if I didn’t edit the SEO Title to be “Leather Wallets from Canada,” The CMS would automatically use the headline “Leather Wallets.”


Add New Page

Leather Wallets

Permalink: https://www.portent.com/leather-wallets

SEO title

Leather Wallets from Canada

Slug

leather-wallets

Meta description

Modify your meta description by editing it right here

A separate title element and headline

Make Meta Tags Editable in the CMS

First: The meta keywords tag is utterly useless and has been since, oh, 2004. Remove it. If your SEO protests, find a new SEO.

With that out of the way, make sure each page has the following editable META tags:

Description

Every page should have an editable description meta tag. The description tag doesn't affect rankings. It does, however, affect clickthrough rate, which can mean organic traffic growth even if rankings don't improve.

Like the title tag, make the description tag a separate, editable field.

If the page is a product page, have the description tag default to the short product description. If the page is a longer descriptive page, have the description tag default to the first 150 characters of the page content.

Never have a blank meta description! If you do, Google and Bing will choose what they think is best. Don't rely on them.

Open Graph Protocol (OGP)

Facebook uses OGP tags to build the text, image, and title of shared content. Without it, Facebook may use the title and meta description tag, and pick an image. It may pick something else.

OGP tags let the content creator control what will appear on Facebook and, like the meta description tag, they can boost clickthru.

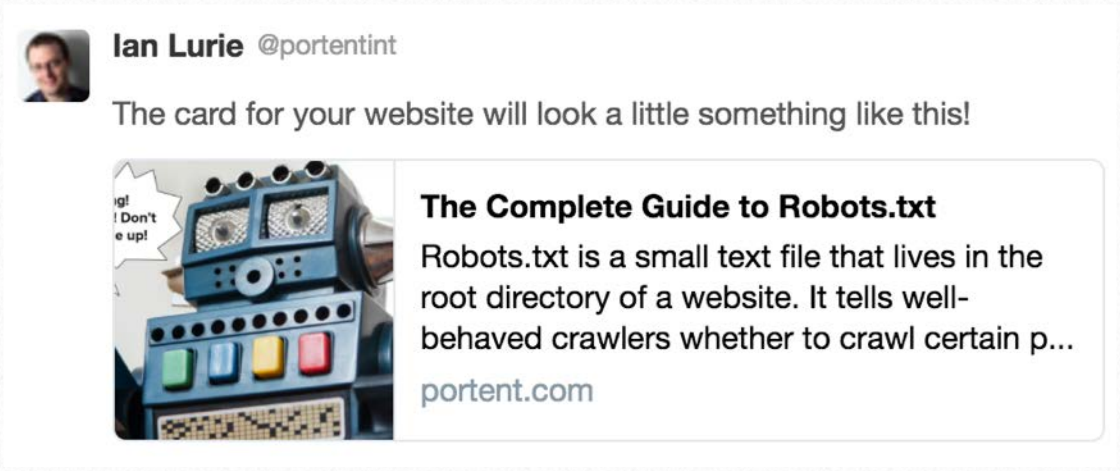
Have the OGP tags default to the page's title, meta description and featured image. Then let the author edit them.

At a minimum, include `og:title`, `og:type`, `og:image` and `og:url`.

You can read more about OGP tags at http://ogp.me/.

Twitter Card Markup

Twitter cards are more niche. Twitter will use OGP tags as a fallback, so these aren't required. If you can add them, though, it gives content creators even more control over what Twitter shows for shared content.


Ian Lurie @portentint

The card for your website will look a little something like this!

The Complete Guide to Robots.txt

Robots.txt is a small text file that lives in the root directory of a website. It tells well-behaved crawlers whether to crawl certain p...

portent.com

Twitter cards can double clickthrough and other engagement. They're worth the effort.

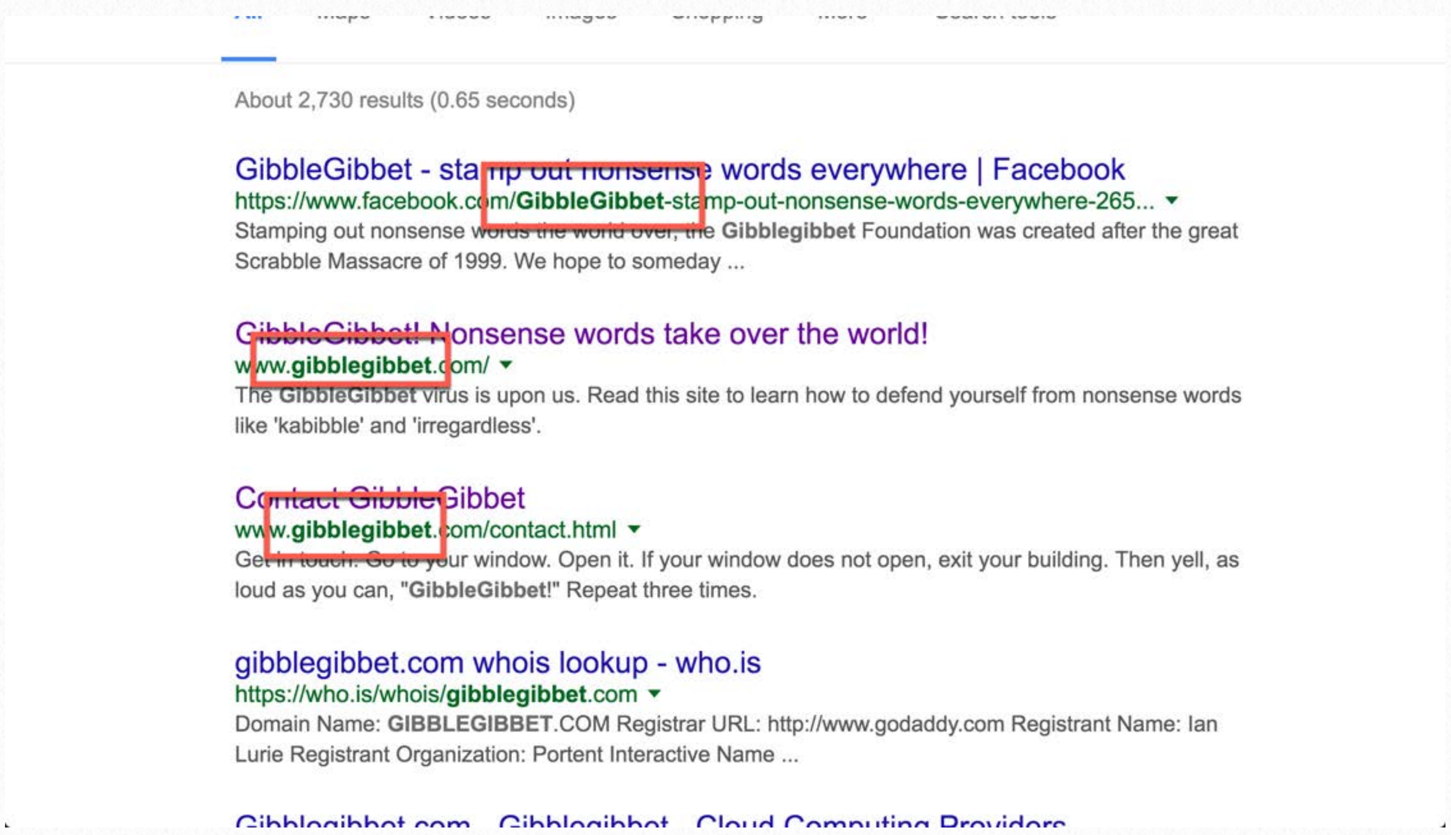
See https://dev.twitter.com/cards/overview for more information.

Image ALT Attributes Editable in the CMS

The ALT attribute is another strong ranking signal. Every image uploaded as part of page content must be editable when the user uploads it. If your CMS lets users write an image caption, have the ALT attribute default to the image caption, but let the user edit it separately if they want to.

URLs

URL keywords don't directly affect rankings, but Google and Bing bold keywords in URLs if those keywords match the query. The right URL can improve click thru from search results.



About 2,730 results (0.65 seconds)

GibbleGibbet - stamp out nonsense words everywhere | Facebook
https://www.facebook.com/**GibbleGibbet**-stamp-out-nonsense-words-everywhere-265... ▾
Stamping out nonsense words the world over, the **Gibblegibbet** Foundation was created after the great Scrabble Massacre of 1999. We hope to someday ...

GibbleGibbet! Nonsense words take over the world!
www.**gibblegibbet**.com/ ▾
The **GibbleGibbet** virus is upon us. Read this site to learn how to defend yourself from nonsense words like 'kabibble' and 'irregardless'.

Contact GibbleGibbet
www.**gibblegibbet**.com/contact.html ▾
Get in touch. Go to your window. Open it. If your window does not open, exit your building. Then yell, as loud as you can, "**GibbleGibbet**!" Repeat three times.

gibblegibbet.com whois lookup - who.is
https://who.is/whois/**gibblegibbet**.com ▾
Domain Name: **GIBBLEGIBBET**.COM Registrar URL: http://www.godaddy.com Registrant Name: Ian Lurie Registrant Organization: Portent Interactive Name ...

Search engines bold matching keywords in URLs. That improves clickthru

Let the user edit the URL. However, make sure the CMS filters for duplicates! I've seen this bite more than one site squarely on the rear end.

Page & Site Structure 5

Use Standard Page Structure

We've already dealt with title elements and such, so this is a lot easier. Every page should:

Have a Single H1

While heading tags don't necessarily affect rankings, page structure as evidenced by rendering does. H1 is the easiest way to represent the top level in the page hierarchy.

Have a single H1 that automatically uses the page headline, whether that's a product description, an article title some other unique page heading.

Do not put the logo, images or content that repeats from page to page in an H1 element.

Make H2, H3, H4 Available to the User

Allow multiple H2, H3, and H4 elements on the page. Let content creators use H2, H3 and H4. You can let them drill down even further, but I've found that leads to some, er, creative page structures.

Use <p> elements for paragraph content, not hard breaks or DIVs
Any developer knows this. Content creators sometimes don’t. I still see many writers insert double line breaks. It’s not easy, but if you can somehow enforce the use of <p> elements for paragraphs, it will make later tweaks to styles a lot easier.

Use Relevant Structured Data.
At a minimum, generate structured markup for:

- Places
- Products
- Reviews
- People

See schema.org for more information. Right now, JSON-LD is the most popular way to add structured data. It’s easiest, and if you (properly) use a tag manager, you can add structured data to the page without changing code.

Site Structure
I’m assuming you’re not the information architect on this project. If you are, let that part of your brain rest for a bit. In the technical context, “site structure” means URLs, content placement, subdomains and such. These are the dev-level things that maximize SEO potential:

Readable URLs
Where possible, create URLs that make sense.

/products/shoes/running

is better than

/products?blah-1231323

Readable URLs may not directly impact rankings. But they improve clickthrough, both because Google and Bing bold keywords in URLs, and because people are more likely to click on readable URLs.

Subdomain vs. Subfolder

All quality content should “live” on the same domain. Use subfolders. The blog should live at /blog. The store should live at /store or similar.

I always get pushback on this one. Google has said in the past that subdomains are OK. Yes, they’re OK. They’re not the best.

Google says subdomains are **sometimes** just as good. Not always. Apparently, when Googlebot comes across a subdomain, it decides whether to treat it as a subfolder or not.

> A guy named Michael Martinez - a downright brilliant SEO - has lambasted me about this. He hasn’t seen any data proving the subfolder vs. subdomain theory. He’s right. Like many things Google does and says, they’re unclear about it, and results differ. In **most** cases, moving content to a subfolder helps, if by “most” we mean “every site I’ve ever worked on.”

We can have a debate about this if you want. Regardless, my recommendation is: Use a subfolder. Here’s one company’s tale of woe:

https://iwantmyname.com/blog/seo-penalties-of-moving-our-blog-to-a-subdomain

There are two times when a subdomain might make more sense:

- If you’re doing reputation management, you need to control as many listings on the first page of a search result as possible. Google often separately ranks subdomain content. A subdomain can help you eat up an additional spot.
- If you’re having trouble with a large amount of low-quality content or thin content, move that to a subdomain, and you may see rankings improvements.

The most common reason folks use subdomains is the blog: The CMS, or server, or something else doesn't support a blog. So you set up a WordPress.com site. That ends up being blog.something.com.

If you have to do that, consider using a reverse proxy to put it all under one domain.

Of course, if you have no choice, use a subdomain. It's better than nothing.

Videos Have Their Own Pages

Video libraries are great, but having all of your videos on a single page makes search engines cry.

Put each video on its own page. Include a description and, if you can, a transcript. Link to each video from the library.

That gives search engines something to rank.

Don't Use Nofollow

Just don't.

Nofollow is meant to prevent penalties for links from comments and advertising. It doesn't help channel PageRank around a site. It does burn PageRank. It's a bad idea.

The **only** time to use nofollow is to avoid a penalty because you're linking by way of ads or other paid space on your site.

> A good rule of thumb: If you're doing something "just" for SEO, think carefully. Nofollow is a good example.

User Experience 6

Google uses a headless browser to render pages. That way, it can “see” how the page lays out and how it will behave when a real person visits.

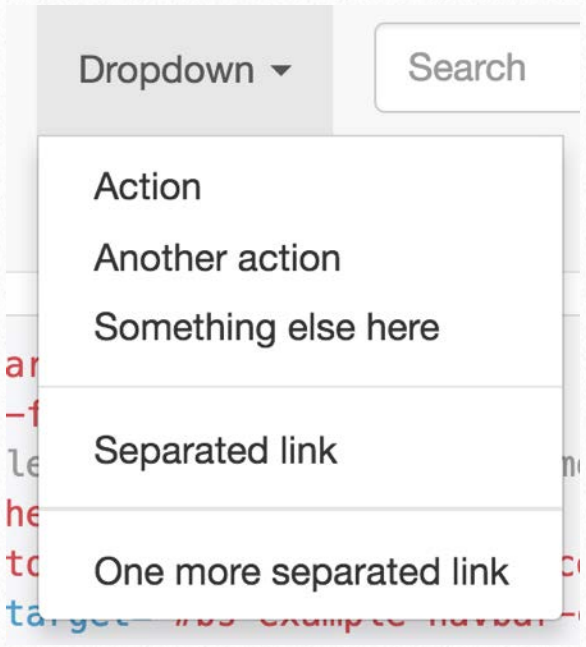
They also look at behavioral signals: High bounce back to search results, for example, may be a negative ranking signal.

These are things that boost direct and indirect ranking signals:

Navigation Is Clickable

Clicking the top level navigation should take me somewhere other than “#.”

In the dropdown below, “Dropdown” links to “#.” Rolling over it activates the menu but doesn’t do anything else.



In this example, clicking "Dropdown" doesn't do anything

That creates three problems:

- The site's primary navigation is a hidden rollover. Google and Bing will attribute less importance to it
- You lose what could be a top-level link to a single page on your site from every other page on your site. That's scads of internal authority gone to waste
- Users will click on "Dropdown" and get frustrated

Make sure clicking any visible navigation takes me somewhere.

All Content Can Be Reached Clicking Links

If you want a page indexed, I need to be able to reach it by clicking on links. Forms, javascript maps, etc., aren't enough.

For example: If you have a stores directory, keep the map and ZIP code search. Just make sure there's also a clickable index I can use to find stores. That means I can link to it, too.

This rule is particularly important when you work with javascript frameworks. See the next chapter for more about that.

Site Performance

Site speed is one of the easiest wins you'll find because it helps everything, from SEO to conversion rates.

No web page should take more than 2 seconds to load. "Load" means "content loaded and interface rendered." Progressive images and such can take longer.

These are the high-level best practices:

- Compress all images, and use the correct format: PNG for line art, icons and other images with few colors; JPG for photographs
- Turn on disk caching. If that causes weird site behaviors, try to track down the problem—disk caching is performance 101, and you can't afford to ignore it. If you want to get fancier, use memory caching, or look at a technology like Varnish
- Use GZIP compression
- Put any CSS or javascript longer than 10-20 lines in a separate file. That lets visiting browser cache the file locally
- Minify JS and CSS
- Set far-future expires headers for files that won't often change. That might include icons, images in carousels, CSS files, and javascript files. Consider how often these files will change, first. Set the expires header accordingly. For example, if you know you'll change CSS every week, don't set a six-month expires header

Here are a few other things that don't fit a nice, neat numbered list:

Third-Party Scripts

Chances are, someone else will add a bunch of third party scripts and clobber site performance. You can get off to a good start:

- Defer loading of third-party scripts, where you can
- Ask the service provider for the compressed version of the script. They often have one
- Use CDN versions wherever possible. For example, you can use the Google CDN version of jquery.

Defer Blocking Javascript and CSS

Google wants immediate page rendering. If you have a monster javascript or CSS file near the top of the page, then all rendering stops while the visiting browser downloads and parses that file.

In mobile search this is a negative ranking signal. Google specifically checks for render-blocking files.

Wherever possible, move blocking scripts and CSS to the bottom of the page, or defer loading.

Use DNS Prefetch

If you're loading assets from a separate site, consider using DNS prefetch. That handles the DNS lookup ahead of time:

```
<link rel="dns-prefetch" href="//foo.com">
```

That reduces DNS lookup time.

Use Prefetch
Find the most popular resources on your site and use prefetch (not to be confused with DNS prefetch, above). That loads the asset when the browser is idle, reducing load time later:

```
<link rel=”prefetch” href=”fonts.woff”>
```

Be careful with prefetch. Too much will slow down the client. Pick the most-accessed pages and other resources and prefetch those.

Use Prerender
Find the most popular pages on your site and prerender those:

```
<link rel=”prerender” href=”products.html”>
```

That loads the page in advance. Same warning as above.

They aren’t saying. If I’m a search engine, though, I’m going to assume that the text behind the tab is less important than the text that’s immediately visible. If you’re trying to optimize for a specific term, don’t bury it.

Other Things
I’m assuming you know about CDNs and such. If not, here’s a little blatant self-promotion. We wrote a series of blog posts called The Ultimate Guide to Page Speed:

http://portent.co/page-speed-guide

Have a look.

Don’t Hide Content (If You Want To Rank for It)
Until, oh, last week (seriously, Google just changed this last week), Google said they wouldn’t consider content that only appeared after user interaction. Content behind tabs, loaded via AJAX when the user clicks, etc., got zero attention.

Last week, the big G said they **do** examine this content, and they do consider it when determining relevance.

I believe them, but as always, they've left out some details:

- Do they assign the same weight to content that requires user interaction?
- Do they differentiate between hidden content (like tabs) and content that doesn't load without user interaction (like asynchronous content)?

Oh, also: The old tiny-content-at-the-bottom-of-the-page trick still doesn't work. That's not what they meant.

Javascript & Frameworks 7

I decided to write a whole separate section about frameworks.

The developer in me loves javascript frameworks. They can speed up content delivery. They let me do all sorts of cool interactive stuff. They can provide an app-like experience to content otherwise constrained by web browsers.

The keep-it-all-above-the-fold guy in me loves things like tabbed content because I can fit more content into a smaller space.

The SEO and old-school UX person in me hates both. Client-side frameworks and things like tabs hide content. They're often an unnecessary layer of complexity. Also, no matter what they say, Google is awful at crawling javascript. We've tested this time and again.

And, the one-page websites created by many client-side frameworks break the clickable, linkable URL rule from the last chapter.

Javascript frameworks and SEO are in flux. This is another place where you need to be very careful when interpreting Google's statements. They've said things like they can "generally" crawl javascript, but that we should use progressive frameworks.

Basically: We'll give it a shot, but build as if we can't.

Also, Bing, Facebook, Twitter, et al. are light years behind.

What I write now might be wrong in two weeks. My recommendation is if you're using a framework, read up and consider the SEO implications. Otherwise, you may end up completely rebuilding your site.

Ask Yourself Why

First, before you get into complicated ways to mitigate the SEO problems caused by many frameworks and javascript widgets, ask yourself, "Why am I building my site this way?"

If there's no compelling argument–if using a framework doesn't offer essential features–consider doing something else.

Understand the Consequences

If you use a javascript framework or otherwise render content via javascript:

- Google rarely indexes content that depends on user events for delivery (behind tabs, etc.)
- Google devalues content that's only revealed after a user event (again, tabs, etc.)
- Any content delivered or rendered after the page's load event will probably not get indexed at all

Put all these together, and you're badly limiting your potential organic search traffic. If that's OK (and it might be), then don't worry about it. If it's not OK, here are some alternatives and approaches that can work:

Only Hide Content When Essential

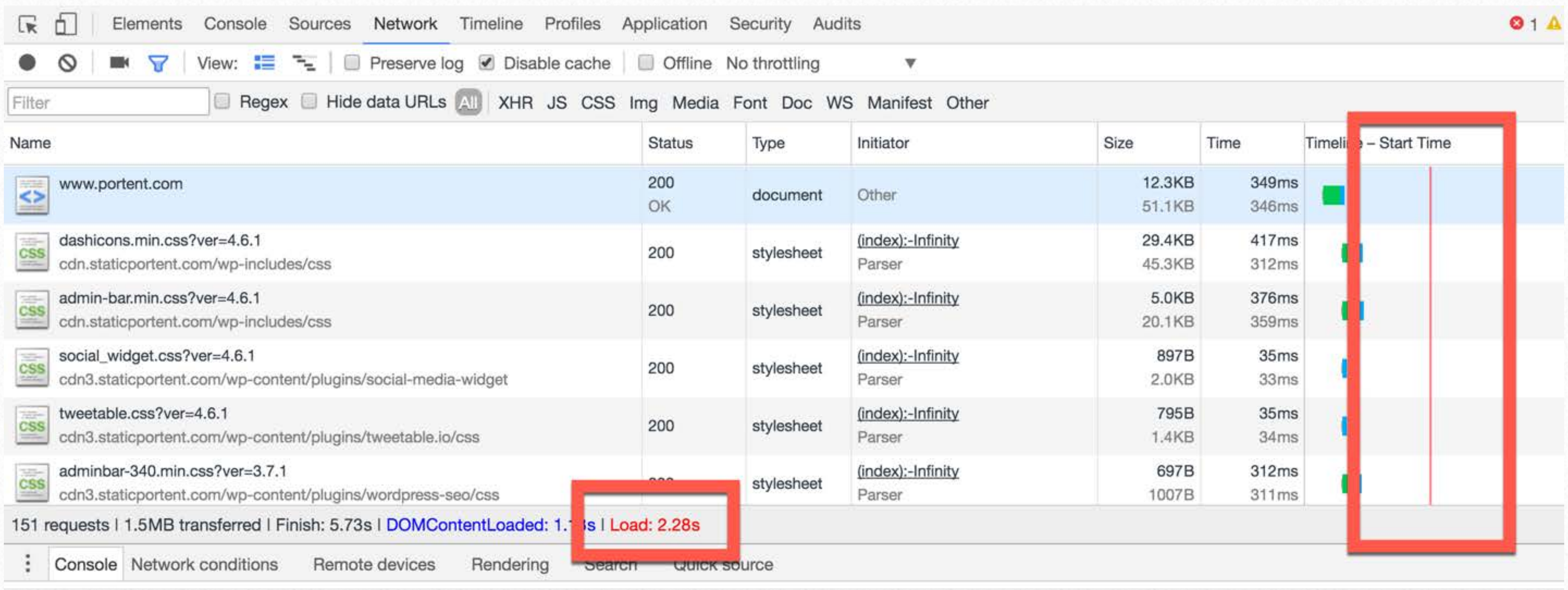
This is the easy part: If you've got content on the page for which you want to rank, don't hide it behind a tab, an accordion, or whatever else. On a well-designed page, people who want to see everything will scroll down. If they don't want to see it, they weren't going to click the tab anyway.

Don't Deliver Content Based on User Events

If you want content indexed, don't deliver it based on a user event. Period. Google won't index anything delivered after someone clicks, rolls over or similar.

Show Content Before the Load Event

Look in your site's HAR. Anything that appears after the "load" event is probably not going to get indexed:


Elements Console Sources Network Timeline Profiles Application Security Audits
View: Preserve log Disable cache Offline No throttling
Filter Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
Console Network conditions Remote devices Rendering Search Quick source

Name	Status	Type	Initiator	Size	Time	Timeli e – Start Time
www.portent.com	200 OK	document	Other	12.3KB 51.1KB	349ms 346ms	
dashicons.min.css?ver=4.6.1 cdn.staticportent.com/wp-includes/css	200	stylesheet	(index):-Infinity Parser	29.4KB 45.3KB	417ms 312ms	
admin-bar.min.css?ver=4.6.1 cdn.staticportent.com/wp-includes/css	200	stylesheet	(index):-Infinity Parser	5.0KB 20.1KB	376ms 359ms	
social_widget.css?ver=4.6.1 cdn3.staticportent.com/wp-content/plugins/social-media-widget	200	stylesheet	(index):-Infinity Parser	897B 2.0KB	35ms 33ms	
tweetable.css?ver=4.6.1 cdn3.staticportent.com/wp-content/plugins/tweetable.io/css	200	stylesheet	(index):-Infinity Parser	795B 1.4KB	35ms 34ms	
adminbar-340.min.css?ver=3.7.1 cdn3.staticportent.com/wp-content/plugins/wordpress-seo/css	[illegible]	stylesheet	(index):-Infinity Parser	697B 1007B	312ms 311ms	

151 requests | 1.5MB transferred | Finish: 5.73s | DOMContentLoaded: 1.[illegible]s | Load: 2.28s

The Load event, in an HAR

Make sure whatever you want indexed appears before then.

Use Indexable URLs

See Make Content Clickable.

URLs with #! and similar won't get crawled. Google deprecated that as an indexing method.

Use Server-Side Rendering

React, Angular 2.0 and a few other frameworks now support server-side rendering. Use one of these the right way and you don't have to worry about load events or any other complications.

> I usually use Express as my web framework. But I'm a complete amateur at this entire style of web development. I learned to build sites on Python and PHP. COBOL. Old-school.

This isn't perfect. Plain old, generate-and-deliver HTML content delivery is still better, because we 100% know how Google and Bing handle it. Server-side rendering is a clean solution that lets you preserve much of the features you want, build on a javascript framework, and still get ranked.

Pre-render Content

Chances are, though, you just inherited an older site built on Angular 1.x or an older version of React, or something else.

In that case, you'll want to pre-render your content. Use a headless browser (Phantomjs is my favorite) to render site pages and save snapshots to your server. Detect visiting user agents and, if the visiting browser is a search engine, show them the snapshot.

You can use something like prerender.io, instead, if necessary.

Google approves this method, as long as the snapshot matches the normal page. Do **not** deliver different content. That's cloaking, and search engines will penalize you.

Consider Progressive Web Apps

Google's all aflutter about progressive web apps. Trendiness aside, if you have a site that's "app-like," consider building it as a progressive app. Done right, they're SEO-friendly, and they offer other advantages.

Read up about them here: https://developers.google.com/web/progressive-web-apps/

Other Considerations 8

HTTPS?

Use HTTPS if you can. Google says it's a ranking factor. In testing we haven't seen any rankings movement. If you can, though, why not? HTTPS can slow site performance though. You also need to think about canonicalization, and how you'll redirect from the non-SSL to SSL addresses. Plan your move carefully.

HTTP/2?

HTTP/2 is faster and more secure (hopefully). It means we can stop mucking about with sprites and other silliness. Right now, Google Chrome supports it. So does Apache. But that's it. It's still an emerging standard.

I'm a late adopter. So I say hold off for a while longer.

AMP?

Google and Bing are now pushing Accelerated Mobile Pages. I'm not a fan, but we're starting to implement them anyway, for three reasons:

- AMP content often gets a featured position in mobile search results. They don't rank higher. They just get preferential treatment with things like content carousels
- AMP listings sometimes get a little icon next to them
- AMP pages are screaming fast

In spite of all that, AMP really bugs me. It limits deliverable content. It ignores those of us who toiled to make our real sites really, really fast. It puts our content on a CDN and therefore gives Google control over our content. It's more code to write.

Make your own choice. When AMP for e-commerce rolls out, though, we'll recommend it. The ranking benefits are hard to ignore.

Regardless, your main site must provide a great mobile experience. AMP is an additional way to deliver content fast. It's not the bottom-line experience for 99% of your users. It strips out much of the user interface. Build responsive first, AMP second.

That’s It

As promised, this is a long list of little things. There are many other little things. They always come up. If you run into something, let us know, and we’ll try to help.

Follow this list, though, and you’ll save yourself round trips and last-minute changes. It’ll make you more efficient and save you from some long nights. It’ll also win you some fans in the marketing department, which can never hurt.

Time to start coding.

Portent wrote this. You can find more about us, and more stuff like this, at www.portent.com.

PORTENT