

SEO Tracking Technical Guide

Sitecore Discover

v 2.0.3 20210220

Table of Contents

Chapter 1	Introduction	3
1.1	AJAX calls and SEO - Things to consider.....	4
Chapter 2	Basics for developing JavaScript websites	5
2.1	Important events from JavaScript that have influence on SEO	7
2.2	Errors that you should avoid with JavaScript.....	8
Chapter 3	Use Schema.org Specifications for Rich Text identification by search engines.....	9
Chapter 4	Verifying Content for SEO	10

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2022 Sitecore. All rights reserved.

Chapter 1 Introduction

Google announced in the webmaster central blog in October 2015 that the search engine is able to render and understand websites and modern browsers, including JavaScript and CSS. In the updated webmaster guidelines, it is recommended that no important site components, including those in JavaScript or CSS, are excluded from crawling via the robots.txt. In the same blog post, Google indicates that it prefers the principle of progressive enhancement, an approach involving the successive improvement of HTML websites. However, JavaScript would still be crawled and rendered.

At the Google developer conference in May 2018, Tom Greenaway of Google announced that the Googlebot has its own renderer that can handle JavaScript as soon as the bot detects JS on a website. However, since rendering JavaScript requires many resources, Greenaway says Google has stopped rendering JS resources until Googlebot is equipped with the necessary resources. Therefore, it's possible that Google indexes a website before the rendering is complete. Only when the final rendering process has been completed will the Googlebot start a new attempt to render the content of the client-side.

1.1 AJAX calls and SEO - Things to consider

Consider the following when using AJAX calls and SEO:

- The content must be indexable within the load event so that JavaScript content makes sense for search engines.
- Content dependent on user events is not indexable.
- JS sites require an indexable URL with server-side support.
- The same SEO best practices used for HTML sites apply to rendering JS sites.
- There should be no contradiction between HTML and JavaScript versions of the website.

Chapter 2 Basics for developing JavaScript websites

JavaScript is often more complex than HTML. To realize what is most important for SEO, you should first try to understand how JavaScript works:

Initial Request: The browser and the search engine bot start a GET request for the HTML code of the website and its affiliated assets.

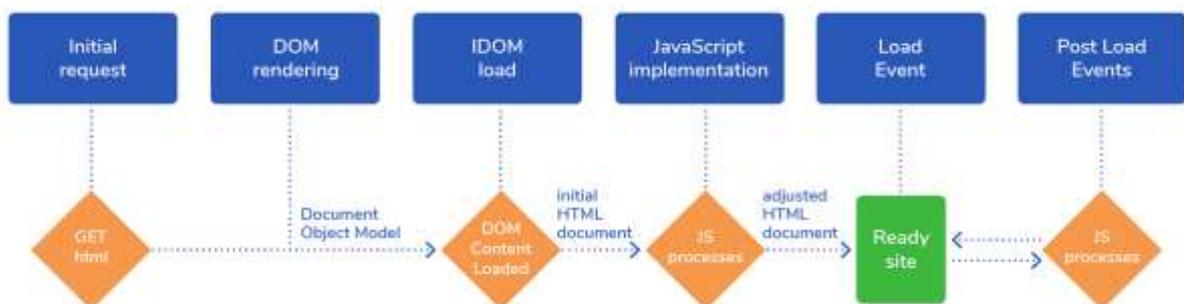
DOM rendering: The JS script site delivers the DOM (Document Object Model) to the browser or the bot. The document shows how the content will be formed on the website and what the connections are between the individual elements on the site. The browser renders this information and makes it visible and usable for the user.

IDOM load: While the target site is being processed, the browser triggers events that belong to the DOMContentLoaded. The initial HTML document is then loaded and stored. The browser or the bot is now ready to implement JavaScript elements.

JavaScript implementation: JavaScript elements can now change the contents or functions of the website without the HTML source code having to be changed, removed, or expanded.

Load Event: As soon as the resources and the JS resources dependent on these are loaded, the browser starts the load, and the site is finished.

Post Load Events: After the JS site has been loaded, further content or functional elements can be changed or adapted by the user.



Sitecore Discover

Search engines such as Google use so-called headless browsers in order to simulate access to a traditional browser. In contrast to the "normal" browser, the headless browser calls up the code via the DOM to render a website from it. In this way, the Googlebot can, for example, check which elements JavaScript inserts in order to modify the HTML site. After the rendering, the Googlebot can analyze and index the rendered elements like an HTML source.

With JavaScript, there are two versions for crawlers, the pre-DOM HTML code and the rendered post-DOM HTML code.

2.1 Important events from JavaScript that have influence on SEO

Load events and user events can clearly influence your SEO. This is why:

- **Load Event:** The load event is "fired" by the browser when a site is completely loaded. Search engine robots imitate common browsers when rendering JavaScript. The load event is therefore important for them - it makes it possible for them to take a snapshot of the rendered content. Events loaded after the load event has been fired will not be considered by JavaScript crawling, and therefore won't be indexed, because JavaScript can quickly change site content. This is especially the case for news sites or social networks feeds, such as Twitter and Facebook.

The time frame of the DOMContentLoaded can be measured with the Google development tool:

- **User Events:** After the load event, further events can be triggered via JavaScript. Among these, for example, are the so-called "onClick-Events." These are user-triggered events, such as the restriction of site content or interactive navigation. However, this content, which is created or changed by user events, is usually not indexed by search engines, because it takes place after the load event.

2.2 Errors that you should avoid with JavaScript

If you use JavaScript on your website, Google can now render the elements after the load event quite well, and can finally read and index the snapshot like a traditional HTML site.

Most problems with JavaScript and SEO result from improper implementation. Many common SEO best practices can therefore also be used for JavaScript SEO. These are a few of the most common errors that can occur:

- **Indexable URLs:** Every website requires unique and distinctive URLs so that the sites can be indexed at all. A pushState, as is created with JavaScript, however, does not generate a URL. Therefore, your JavaScript site also requires its own web document that can give a status code 200 OK as a server answer to a client or bot inquiry. Every product presented with JS (or each category of your website realized with JS) must therefore be equipped with a server-side URL so that your site can be indexed.
- **pushState errors:** With the pushState method, JavaScript URLs can be changed. Therefore, you must absolutely ensure that the original URL is relayed with server-side support. Otherwise, you risk duplicate content.
- **Missing Metadata:** With the use of JavaScript, many webmasters or SEO forget the basics and do not transfer meta data to the bot. However, the same SEO standards hold for JavaScript content as for HTML sites. So, think about title and meta description of ALT tags for images. We recommend you to follow - <https://schema.org/Product>. More on this later.
- **a href and img src:** The Googlebot requires links that it can follow so that it can find further sites. Therefore, you should also provide links with href- or src-attributes in your JS documents.

Create unified versions: Through the rendering of JavaScript, preDOM and postDOM versions arise. Ensure that, if possible, no contradictions slip in and, for example, that canonical tags or paginations can be correctly interpreted. In this way, you will avoid [cloaking](#).

Create access for all bots: Not all bots can deal with JavaScript like the Googlebot. It is therefore recommended to place title, meta information, and social tags in the HTML code.

Don't disable JS over robots.txt: Ensure that your JavaScript can also be crawled by the Googlebot. For this, the directories should not be excluded in the robots.txt.

Use a current sitemap: In order to show Google any possible changes in the JS contents, you should always keep the attribute "lastmod" current in your XML sitemap.

Chapter 3 Use Schema.org Specifications for Rich Text identification by search engines

We recommend you follow schema.org in rendering HTML. Following is an example:

```
<div itemscope itemtype="http://schema.org/Product">
  
  <span itemprop="name">Dell UltraSharp 30" LCD Monitor</span>
  <div itemprop="aggregateRating"
    itemscope itemtype="http://schema.org/AggregateRating">
    <span itemprop="ratingValue">87</span>
    out of <span itemprop="bestRating">100</span>
    based on <span itemprop="ratingCount">24</span> user ratings
  </div>
  <div itemprop="offers" itemscope itemtype="http://schema.org/AggregateOffer">
    <span itemprop="lowPrice">$1250</span>
    to <span itemprop="highPrice">$1495</span>
    from <span itemprop="offerCount">8</span> sellers
    Sellers:
    <div itemprop="offers" itemscope itemtype="http://schema.org/Offer">
      <a itemprop="url" href="save-a-lot-monitors.com/dell-30.html">
        Save A Lot Monitors - $1250</a>
    </div>
    <div itemprop="offers" itemscope itemtype="http://schema.org/Offer">
      <a itemprop="url" href="jondoe-gadgets.com/dell-30.html">
        Jon Doe's Gadgets - $1350</a>
    </div>
  </div>
  ...
</div>
```

Chapter 4 Verifying Content for SEO

There are several commercial tools available for verifying SEO quality of your html pages/content. You may use whatever tools you use in a normal course to verify content that we serve.

You may use the following free tools from Google to verify how Google search engine sees your pages.

Google Search Console: <https://search.google.com/search-console>

Google Rich Text: <https://search.google.com/test/rich-results>