Manage SEO in JavaScript applications

June 28, 2022 • Jan Cerman • 5 min read

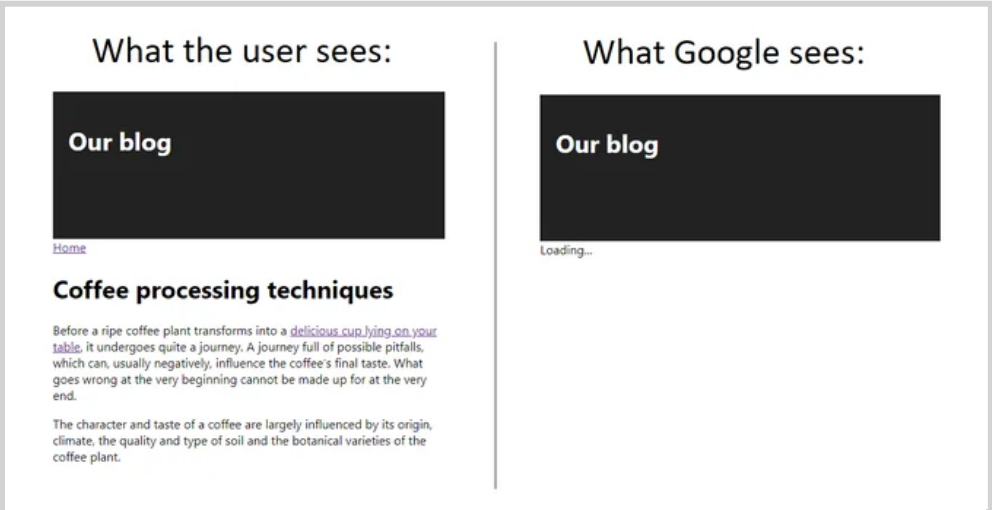
When developing any single page application or a website that uses JavaScript to fetch content from APIs, you need to think carefully about search engine optimization.

Key points

- Search engines don't index AJAX responses from Delivery API.
- For the best SEO of your JavaScript app, serve fully rendered HTML to the search engines.
- Use SEO applications to test how your app gets indexed by search engines.

Indexing by search engines

Content fetched from Delivery API via client-side AJAX requests **will not** be indexed by Google, Bing, and other search engines. By default, the API doesn't allow it. Most engines apart from Google don't even try to index JavaScript calls. If you need the API responses to be indexable, discuss the requirement with our support.



The best practice is to make sure search engines are served fully rendered HTML with all the content you wish to be indexed.

This can be achieved in several ways, each of them presenting a trade-off between performance, complexity, and cost. In general, you have three options:

- Server-side rendering
- Prerendering
- Static site generators

Below you will find a brief introduction to each approach and links to more resources. There is no single best solution; you always need to consider the specific requirements of your project.

Server-side rendering

Server-side rendering is the practice of using a server to render content and serve the rendered HTML on the initial page load. The full JavaScript bundle loads later and subsequent requests are handled by the client. This solves the SEO problem and can improve perceived performance by reducing the initial load time of your single-page app.

These kinds of apps are sometimes called isomorphic applications because their code can run on both client and server.

On the downside, server-side rendering increases the complexity of your application and can hinder performance in some situations, for example, when the server is under a heavy load.

- Read more about server-side rendering to decide if it makes sense for your project.

All major front-end frameworks support server-side rendering or provide dedicated server-side rendering frameworks:

- React: Read about React server-side rendering or check out Next.js.
- Angular: See Angular Universal.
- Vue: Read the Vue server-side rendering guide or check out Nuxt.js.

Prerendering

Rather than using a web server to compile HTML *on the fly* per request, prerendering simply generates static HTML files for specific routes *at build time*. The advantage is that setting up prerendering is simpler and it allows you to keep your frontend as a static site without the need for a node server.

Any single page application using *webpack* can be prerendered using the Prerender SPA Plugin (often used for Vue apps).

For React specifically, there is react-snapshot and react-snap:

- `react-snapshot` tutorial on Medium
- `react-snap` tutorial on Medium

Prerendering is also provided by several **3rd-party services**, for example:

- Netlify provides basic prerendering even for free its free plan.
- Prerender.io is a dedicated prerendering service and is also open-source.

Static site generators

Static site generators are tools that take your source code, assets, content from APIs, and so on, and generate your website as a collection of static files. You can host the static files anywhere and SEO is not a problem. If your website doesn't require authentication or display real-time or user-specific data, consider using a static site generator.

- Staticgen.com provides an overview of all static site generators.
- Gatsby.js is a React-based static site generator with a Kontent.ai source plugin.
- Nuxt.js works as a Vue-based static site generator and has a Kontent.ai module.
- Next.js is an open-source React framework with a Kontent.ai corporate starter.

Some tools combine more than one approach, for example, Next.js and Nuxt.js support both server-side rendering and static site generation. The lines between server-side rendering, static site generators, and prerendering can be blurry.

Testing

You can use the URL inspection Tool by Google to see the index status of your website.

What's next?

This article serves as a starting point to get your JavaScript application indexed correctly. SEO for JavaScript apps is a rapidly evolving area of the JavaScript landscape so we encourage you to research the best approach for your specific requirements.

- Have a look at our JavaScript sample apps and tools.
- If you have any questions, get in touch with us using the chat button in the bottom-right corner.