

Adding Sitemaps to Gatsby Websites

Daniel Stout



It is widely accepted that having a sitemap file in your website is helpful for SEO purposes. They are essentially a “roadmap” that search engine crawlers use to index pages of websites. Thankfully, adding a sitemap to a [Gatsby.js](#) project is incredibly simple! Let’s go over how in the post below.

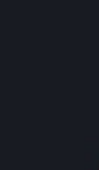
stay in the loop with our monthly web dev newsletter:

you@awesome.com

Subscribe!

follow us @alligatorio

site sponsors



SNIPCART

Add custom e-commerce to any site in minutes.

learn more →

[about sponsors](#)

Installation

This post assumes that you already have a Gatsby project set up, and that you are ready to edit. If you need help getting to that point, please follow along with the intro tutorial in the [Your First Steps with Gatsby v2](#) post.

Alligator.io recommends ↴

Fullstack Advanced React & GraphQL by Wes Bos

ⓘ About this affiliate link

Install the plugin

As you will probably notice sooner rather than later, there are a **lot** of useful official Gatsby plugins! It’s always a wise idea to look through [the plugins section on their official site](#) when building out your projects. For our sitemap, we will make use of [gatsby-plugin-sitemap](#).

This is the only new dependency we are adding to our project, so let’s add it like this:

```
$ yarn add gatsby-plugin-sitemap
```

Note: It’s *not* ok to install using npm, if you prefer using that!

Add the plugin to your Gatsby config:

All that’s left to do is add the sitemap plugin to our `gatsby-config.js` file, and optionally configure it.

Before adding the plugin, we need to make sure that the `siteMetadata` object has a valid `siteUrl` value set:

↴ [gatsby-config.js](#)

```
siteMetadata: {
  siteUrl: 'https://www.your-site-here.com',
  //... other values, e.g.
  // title: 'Alligator Bytes',
  // description: 'Chomp! Coding mini-posts from Alligator.io'
},
```

⚠ Important: Don’t skip this step! By default, this plugin uses the ‘siteMetadata’ value to build out the urls for your sitemap.

Next, a little further down inside the `gatsby-config.js` file, we add the plugin into our `plugins` array.

The easiest way to add the plugin involves no extra configuration, and will automatically create sitemap items for every page in our site. [This ‘no-config method’ even includes programmatically generated pages, such as blog posts from markdown!](#) Let’s do that now:

↴ [gatsby-config.js](#)

```
//...
plugins: [
  'gatsby-plugin-sitemap',
  //... other plugins here
]
```

Adding configuration options is no problem, though! We just use an object inside our `plugins` array, like this:

↴ [gatsby-config.js](#)

```
// ... siteMetadata above here
plugins: [
  {
    resolve: 'gatsby-plugin-sitemap',
    options: {
      output: './path/to/sitemap-alternate.xml',
      exclude: ['/some-page']
    }
  },
  // ... other plugins here
]
```

And that’s it, we are done! The plugin will automatically create our sitemap file with every `production` build.

Configuration options

According to the [plugin documentation](#), there are several options available for customizing our sitemap generation.

► **query** : (GraphQL query) This is a [GraphQL query used](#) to fetch the data needed to generate the sitemap. It’s used by default to get the `site.siteMetadata.siteUrl` value we set above, and all available pages. The default query looks like this, and you will rarely need to add your own:

```
query: `
  {
    site {
      siteMetadata {
        siteUrl
      }
    }
    allSitePage {
      edges {
        node {
          path
        }
      }
    }
  }`
```

► **output** : (string) Set the output path and filename here. The default value is `/sitemap.xml`, and that is also what I recommend unless you have special reasons.

► **exclude** : (array of strings) You can use this to exclude pages from the sitemap, using an array of strings with [glob](#) pattern formatting. For example, to exclude a single `pineapple` page and all page routes beginning with `winter`, we could set this value to `["/pineapple", "/winter/*"]` The default value for this is:

```
[
  '/dev-404-page',
  '/404',
  '/404.html',
  '/offline-plugin-app-shell-fallback',
]
```

► **createLinkInHead** : (boolean) This value determines whether or not a link to the sitemap is automatically placed in the `head` of your site.

► **serialize** : (function) This is the function that takes the `query` output from above, and returns an array of properly structured sitemap entries. The default function looks like this:

```
({ site, allSitePage }) =>
allSitePage.edges.map(edge => {
  return {
    url: site.siteMetadata.siteUrl + edge.node.path,
    changefreq: 'daily',
    priority: 0.7,
  }
})
```

► **sitemapSize** : (below)

Handle extremely large sites, with ease

When dealing with extremely large websites, it’s always best to split up the entries into separate smaller files. The [recommended way of handling this](#) is by creating and using a `sitemap index file`, and this plugin can automatically do that for us!

Activating this feature only requires one extra option, `sitemapSize`. This integer value sets the maximum number of pages per sitemap file. Let’s say we have a website with 30K+ pages, and we want to break that up into sitemap files with a maximum of 5K per file. Our configuration, with no other options set, would look like this:

```
// ... siteMetadata above here
plugins: [
  {
    resolve: 'gatsby-plugin-sitemap',
    options: {
      sitemapSize: 5000
    }
  },
  // ... other plugins
]
```

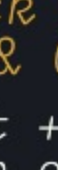
This would generate five sitemap files, named `sitemap-0.xml`, `sitemap-1.xml`, `sitemap-2.xml`, and so on... along with a main `sitemap.xml` file that behaves as a sitemap index.

Conclusion

Sitemaps are a critical part of any website for proper search engine indexing, and **especially** for newer sites with a low number of backlinks. It’s an often overlooked step, but it can dramatically speed up the indexing process. Fortunately, with Gatsby and their amazing [sitemap plugin](#), we can add one to our projects in no time!

💡 **Final tip:** I fully recommend using this plugin with all your Gatsby projects, but it’s so simple to install that it can be easily forgotten. To prevent that, be sure to add this step as a new item in your “Gatsby project checklist”! (And start making a checklist now, if you don’t have one already!) ✅

Published: July 15, 2019



Fullstack Advanced
React & GraphQL
Learn React + GraphQL by
building an online store

ⓘ About this affiliate link

Catch up with the latest in front-end web dev with our monthly newsletter:

you@unicorn.com

Subscribe

Write for Us