Add a Plugin to Your Site

Introduction

Gatsby plugins are Node.js packages that you can add to your site. Plugins let you quickly add new features to your Gatsby site without needing to build them from scratch.

In this guide, you'll learn how to install and configure a plugin.

Prerequisites

Before you begin, you should already have:

- An existing Gatsby site. (Need help creating one? Follow the Quick Start.)

Directions

The general process for adding a plugin to your site looks like this:

1. Install the plugin.
2. Configure the plugin in your `gatsby-config.js` file.
3. (When applicable) Use features from the plugin in your site.

In this guide, you'll add the `gatsby-plugin-sitemap` plugin to your Gatsby site, but you can use a similar procedure with any other plugin.

Using a different plugin? Check the plugin's README in the Gatsby Plugin Library for more details on how to set it up.

Step 1: Install the plugin

Add the plugin as a project dependency in your `package.json` file by running the following command in the terminal:

```
npm install gatsby-plugin-sitemap
```

Check the plugin's README file to see if there are any other dependencies that you also need to install. (For example, `gatsby-plugin-mdx` also requires `@mdx-js/mdx` and `@mdx-js/react` to be installed.)

Step 2: Configure the plugin in your `gatsby-config.js` file

Your `gatsby-config.js` file contains information about your site, including configuration for plugins.

To add a plugin to your site, add its name to the `plugins` array in your `gatsby-config.js` file:

```
module.exports = {
  siteMetadata: {
    title: "My Cool Website",
  },
  plugins: ["gatsby-plugin-sitemap"], // highlight-line
}
```

Note: Generally, the order of plugins in your `plugins` array isn't important. In cases where order is important, the plugin README will have more details.

Using plugin configuration options If your plugin requires additional configuration options, add a configuration object to your `plugins` array *instead* of using a string with the plugin's name. For example, here's how you would configure `gatsby-plugin-sitemap` with a different `output` option:

```
module.exports = {
  siteMetadata: {
    title: "My Cool Website",
  },
  plugins: [
    // highlight-start
    {
      resolve: "gatsby-plugin-sitemap",
      options: {
        output: `/my-cool-sitemap.xml`,
      },
    },
    // highlight-end
  ],
}
```

Check the plugin README in the Gatsby Plugin Library for more information about the configuration options available.

Note: Plugin options will be stringified by Gatsby, so they cannot be functions.

Step 3: (When applicable) Use features from the plugin in your site

Different plugins provide different features that you can use in your Gatsby site.

For some plugins, like `gatsby-plugin-sitemap`, you can skip this step. Once you add the plugin to your `gatsby-config.js` file, the plugin handles the rest of the functionality automatically.

Other plugins might provide components or functions, which you then need to import and use in other files throughout your site. For example, the `gatsby-plugin-image` plugin exports a `<GatsbyImage>` component that you can use to add images to your site. Check the plugin's README to see whether there are any additional steps you need to do to use the plugin in your site.

Additional Resources

- Gatsby Plugin Library - A list of all the available Gatsby plugins.
- How-To Guide: Creating a Generic Plugin - An introduction to creating your own plugin.