A Gatsby plugin abstracts Gatsby APIs into an installable package. This means that modular chunks of Gatsby functionality aren’t directly written into your project, but rather versioned, centrally managed, and installed as a dependency. You can add external data, transform data, add third-party services (e.g. Google Analytics, Stripe), and more.

Using a plugin

Found a plugin you'd like to use in your project? Awesome! You can configure it for use by following the steps below. This recipe uses the `gatsby-source-filesystem` plugin as an example.

> *If you'd like to take a look at available plugins, check out the plugin library.*

Prerequisites

- An existing Gatsby site with a `gatsby-config.js` file

Directions

1. Install the `gatsby-source-filesystem` plugin by running the following command:

```
npm install gatsby-source-filesystem
```

2. Add the plugin to your `gatsby-config.js`, and set any options it needs, the filesystem source plugin takes a `name` and `path` as options:

```
module.exports = {
  plugins: [
    // highlight-start
    {
      resolve: `gatsby-source-filesystem`,
      options: {
        name: `images`,
        path: `${__dirname}/src/images`,
      },
    },
    // highlight-end
  ],
}
```

The instructions found in the README of the plugin you're using can help you determine specifics about what configurations (if any) it requires. For this example to have an effect in your own site, you would need to create the `src/images` folder and place files inside of it.

3. Run `gatsby develop`, your plugin should run as your site builds.

Additional resources

- Learn more about configuring options or using default options in the Using a Plugin in Your Site guide.
- See an example Gatsby site using this configuration in the repo for the default Gatsby starter.

Creating a new plugin using a plugin starter

If you want to create your own plugin you can get started with the Gatsby plugin starter.

Prerequisites

- An existing Gatsby site with a `gatsby-config.js` file

Directions

1. *Outside* of your Gatsby site, generate a new plugin based on the starter using the `gatsby new` command:

```
gatsby new my-plugin https://github.com/gatsbyjs/gatsby-starter-plugin
```

The directory structure should look something like this:

```
gatsby-site
└── gatsby-config.js
└── src
    ├── components
    └── pages
my-plugin
├── gatsby-browser.js
├── gatsby-node.js
├── gatsby-ssr.js
├── index.js
└── package.json
```

2. Add the plugin to your site's `gatsby-config.js`, linking it to your local plugin's root folder with `require.resolve`. The path (or name of the plugin) should be to the directory name you used when generating the plugin:

```
module.exports = {
  plugins: [
    require.resolve(`../my-plugin`), // highlight-line
  ],
}
```

3. Run `gatsby develop`. To verify the plugin starter loaded correctly in your site it will log a message to the console saying it "Loaded" before the `onPreInit` step finishes:

```
$ gatsby develop
success open and validate gatsby-configs - 0.033s
success load plugins - 0.074s
Loaded gatsby-starter-plugin
success onPreInit - 0.016s
...
```

4. Now you can implement browser, server-side rendering, or node APIs and your site will run them each time it loads your plugin!

Additional resources

- Read about creating and loading your own plugins in development in the Creating a Local Plugin guide