Gatsby Admin

A visual interface to configure your Gatsby site. Currently alpha testing.

Getting started

For now, Gatsby Admin is marked as experimental and requires a flag to enable. Add the following flag when using `gatsby develop`:

```
GATSBY_EXPERIMENTAL_ENABLE_ADMIN=true gatsby develop
```

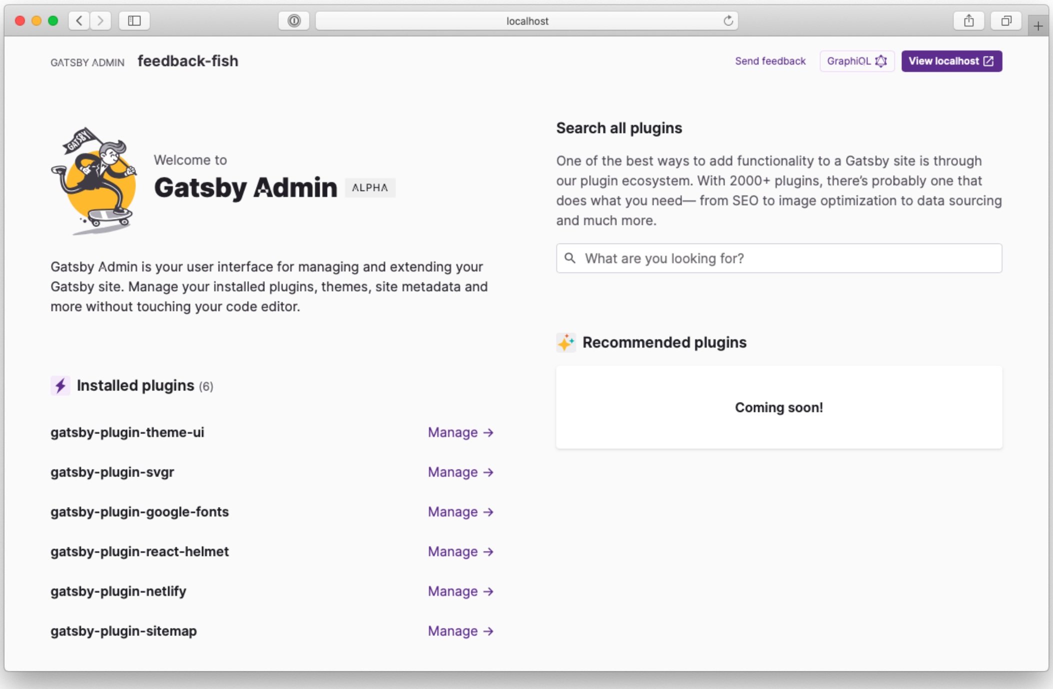
Or in the `scripts` section of your `package.json`:

```
{
  "develop": "GATSBY_EXPERIMENTAL_ENABLE_ADMIN=true gatsby develop"
}
```

Note: If you're on Windows you should install `cross-env` and prepend your script, e.g.:

```
{
  "develop": "cross-env GATSBY_EXPERIMENTAL_ENABLE_ADMIN=true gatsby develop"
}
```

Once you run `gatsby develop` with the flag enabled, you can visit `http://localhost:8000/___admin` to view Admin for your Gatsby site!



Gatsby Admin is your user interface for managing and extending your Gatsby site. Manage your installed plugins, themes, site metadata and more without touching your code editor.

Installed plugins (6)

- gatsby-plugin-theme-ui — Manage →
- gatsby-plugin-svgr — Manage →
- gatsby-plugin-google-fonts — Manage →
- gatsby-plugin-react-helmet — Manage →
- gatsby-plugin-netlify — Manage →
- gatsby-plugin-sitemap — Manage →

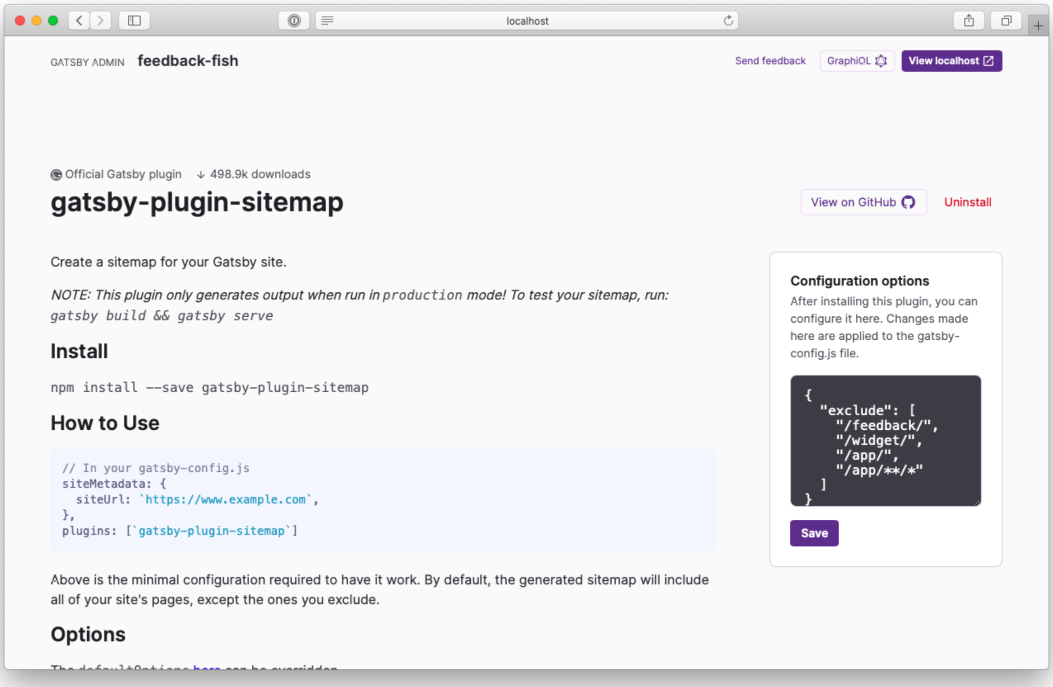
Search all plugins

One of the best ways to add functionality to a Gatsby site is through our plugin ecosystem. With 2000+ plugins, there's probably one that does what you need— from SEO to image optimization to data sourcing and much more.

Recommended plugins

Coming soon!

When you select an installed plugin or search for a new one, you will be able to install/uninstall/configure it:



Official Gatsby plugin ↓ 498.9k downloads

gatsby-plugin-sitemap

Create a sitemap for your Gatsby site.

NOTE: This plugin only generates output when run in `production` *mode! To test your sitemap, run:* `gatsby build && gatsby serve`

Install

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

How to Use

```
// In your gatsby-config.js
siteMetadata: {
  siteUrl: `https://www.example.com`,
},
plugins: [`gatsby-plugin-sitemap`]
```

Above is the minimal configuration required to have it work. By default, the generated sitemap will include all of your site's pages, except the ones you exclude.

Options

The `defaultOptions` here can be overridden

Configuration options

After installing this plugin, you can configure it here. Changes made here are applied to the gatsby-config.js file.

```
{
  "exclude": [
    "/feedback/",
    "/widget/",
    "/app/",
    "/app/**/*"
  ]
}
```

Let us know what you think by hitting the "Send feedback" button in Admin or commenting in the Admin umbrella issue!

Troubleshooting

Admin does not support all `gatsby-config.js` formats. If yours is not supported, we'd love if you could share what it looks like in the umbrella issue!

Technical documentation

Architecture

The Gatsby Admin interface (this package) is a standard Gatsby site.

It uses theme-ui (with the strict-ui experimental extension) and gatsby-interface for styling.

It fetches its data from the gatsby-recipes GraphQL server, which exposes all the information Admin needs about the locally running Gatsby site, using urql.

It also listens to the `gatsby develop` status server, which exposes information about whether you changed the config files and need to restart the develop process.

Service Discovery

`gatsby develop` automatically starts both the GraphQL and status server. However, both of these use random ports.

To discover where they are (and whether they are already running) there is a service discovery mechanism in `gatsby-core-utils`. It stores the ports of the running Gatsby site(s) at `~/.config/gatsby/sites/<pathhash>/<servername>.json`.

Admin can then fetch `http://localhost:8000/___services` (where `:8000` is the well-known port of the running site), which returns a list of all the random ports used by the site:

```
$ curl http://localhost:8000/___services | jq
{
  "developproxy": {
    "port": 8000
  },
  "developstatusserver": {
    "port": 60731
  },
  "recipesgraphqlserver": {
    "port": 50400
  }
}
```

That's how the Admin frontend knows to connect to `http://localhost:50400/graphql` to connect to the GraphQL server, and `http://localhost:60731` to connect to the develop status server.

Production Deployment

To avoid clashing with the local site and potential issues with shadowing, `gatsby develop` statically serves the built files from the develop parent proxy.

To avoid issues with yarn, lerna, and circular dependencies, `gatsby-admin` copies its built files to `gatsby/gatsby-admin-public` which is then published to npm. While not an ideal solution, it fixes the issue and works relatively reliably.

Development

Running it locally

The easiest way to work on Admin locally is to develop Admin itself.

1. Make sure to have the dependencies installed by running `yarn` in the root folder
2. Start the develop process for the Admin site by running `yarn workspace gatsby-admin run develop`

> *If you see eslint errors you'll need to temporarily replace all references to `___loader` with `window.___loader` in `packages/gatsby-link/index.js`.*

Running it on a local site

To try Admin with one of your sites locally, use the `gatsby-dev-cli` to copy the local versions of `gatsby`, `gatsby-cli`, `gatsby-recipes`, `gatsby-core-utils` and `gatsby-admin` into your project:

```
# Make sure to build the latest local versions of all packages
~/gatsby
yarn run watch
```

```
~/my-gatsby-site
$ gatsby-dev --packages gatsby gatsby-cli gatsby-recipes gatsby-core-utils gatsby-admin

# In another tab, start your site with the Admin feature flag set
~/my-gatsby-site
$ GATSBY_EXPERIMENTAL_ENABLE_ADMIN=true gatsby develop
```

Then visit `http://localhost:8000/\_\_\_admin` and you should see Gatsby Admin for your site!