Starter for a Gatsby Plugin

A minimal boilerplate for the essential files Gatsby looks for in a plugin.

🚀 Quick start

To get started creating a new plugin, you can follow these steps:

1. Initialize a new plugin from the starter with `gatsby new`

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

If you already have a Gatsby site, you can use it. Otherwise, you can create a new Gatsby site to test your plugin.

Your directory structure will look similar to this:

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

With `my-gatsby-site` being your Gatsby site, and `my-plugin` being your plugin. You could also include the plugin in your site's `plugins` folder.

2. Include the plugin in a Gatsby site

Inside of the `gatsby-config.js` file of your site (in this case, `my-gatsby-site`), include the plugin in the `plugins` array:

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

The line `require.resolve('../my-plugin')` is what accesses the plugin based on its filepath on your computer, and adds it as a plugin when Gatsby runs.

You can use this method to test and develop your plugin before you publish it to a package registry like npm. Once published, you would instead install it and add the plugin name to the array. You can read about other ways to connect your plugin to your site including using `npm link` or `yarn workspaces` in the doc on creating local plugins.

3. Verify the plugin was added correctly

The plugin added by the starter implements a single Gatsby API in the `gatsby-node` that logs a message to the console. When you run `gatsby develop` or `gatsby build` in the site that implements your plugin, you should see this message.

You can verify your plugin was added to your site correctly by running `gatsby develop` for the site.

You should now see a message logged to the console in the preinit phase of the Gatsby build process:

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

4. Rename the plugin in the `package.json`

When you clone the site, the information in the `package.json` will need to be updated. Name your plugin based off of Gatsby's conventions for naming plugins.

🧐 What's inside?

This starter generates the files Gatsby looks for in plugins.

```
/my-plugin
├── .gitignore
├── gatsby-browser.js
├── gatsby-node.js
├── gatsby-ssr.js
├── index.js
├── LICENSE
├── package.json
└── README.md
```

- **`.gitignore`**: This file tells git which files it should not track / not maintain a version history for.
- **`gatsby-browser.js`**: This file is where Gatsby expects to find any usage of the Gatsby browser APIs (if any). These allow customization/extension of default Gatsby settings affecting the browser.
- **`gatsby-node.js`**: This file is where Gatsby expects to find any usage of the Gatsby Node APIs (if any). These allow customization/extension of default Gatsby settings affecting pieces of the site build process.
- **`gatsby-ssr.js`**: This file is where Gatsby expects to find any usage of the Gatsby server-side rendering APIs (if any). These allow customization of default Gatsby settings affecting server-side rendering.

- `index.js` : A file that will be loaded by default when the plugin is required by another application. You can adjust what file is used by updating the `main` field of the `package.json` .
- `LICENSE` : This plugin starter is licensed under the 0BSD license. This means that you can see this file as a placeholder and replace it with your own license.
- `package.json` : A manifest file for Node.js projects, which includes things like metadata (the plugin's name, author, etc). This manifest is how npm knows which packages to install for your project.
- `README.md` : A text file containing useful reference information about your plugin.

🎓 Learning Gatsby

If you're looking for more guidance on plugins, how they work, or what their role is in the Gatsby ecosystem, check out some of these resources:

- The Creating Plugins section of the docs has information on authoring and maintaining plugins yourself.
- The conceptual guide on Plugins, Themes, and Starters compares and contrasts plugins with other pieces of the Gatsby ecosystem. It can also help you decide what to choose between a plugin, starter, or theme.
- The Gatsby plugin library has over 1750 official as well as community developed plugins that can get you up and running faster and borrow ideas from.