gatsby-plugin-netlify-cms

Gatsby v1 and Netlify CMS 1.x require `gatsby-plugin-netlify-cms@^2.0.0`.

Gatsby v2 and Netlify CMS 2.x require `gatsby-plugin-netlify-cms@^3.0.0`.

Gatsby v2 and Netlify CMS (netlify-cms-app) 2.9.x required `gatsby-plugin-netlify-cms@^4.0.0`, which is documented below.

Overview

Automatically generates an `admin/index.html` with a default Netlify CMS implementation.

Netlify CMS is a React single page app for editing git based content via API. Its built for non-technical and technical editors alike, and its super easy to install and configure. For more details, check out the docs site.

Note: `gatsby-plugin-netlify-cms@^4.0.0` changes the requirement for Netlify CMS to use a new library published `netlify-cms-app@^2.9.x` and is a **breaking change**.

Install

```
npm install netlify-cms-app gatsby-plugin-netlify-cms
```

How to use

Add the Netlify CMS plugin in your `gatsby-config.js`:

```
plugins: [`gatsby-plugin-netlify-cms`]
```

Then add your Netlify CMS configuration file in `static/admin/config.yml`.

Options

Netlify CMS can be configured via the plugin options below. You can learn about how to pass options to plugins in the Gatsby docs.

`modulePath`

(*optional*, type: `string | Array<string>`, default: `undefined`)

If you need to customize Netlify CMS, e.g. registering custom widgets or styling the preview pane, you'll need to do so in a JavaScript module and provide Gatsby with the path to your module via the `modulePath` option. Any styles imported by this module (or by the modules that it imports, all the way down the chain) are automatically applied to the editor preview pane by the plugin.

```
plugins: [
  {
    resolve: `gatsby-plugin-netlify-cms`,
    options: {
      /**
       * One convention is to place your Netlify CMS customization code in a
       * `src/cms` directory.
       */
      modulePath: `${__dirname}/src/cms/cms.js`,
    },
  },
]
```

The js module might look like this:

```
/**
 * The default export of `netlify-cms-app` is an object with all of the Netlify CMS
 * extension registration methods, such as `registerWidget` and
 * `registerPreviewTemplate`.
 */
import CMS from "netlify-cms-app"

/**
 * Any imported styles should be automatically be applied to the editor preview
 * pane thus eliminating the need to use `registerPreviewStyle` for imported
 * styles. However if you are experiencing build errors regarding importing css,
 * sass or scss into a cms module when deploying to the netlify platform, you
 * may need to follow the implementation found in netlify documentation here:
 * https://www.netlifycms.org/docs/beta-features/#raw-css-in-registerpreviewstyle
 * All of the example imports below would result in styles being applied to the
 * preview pane.
 */
import "module-that-imports-styles.js"
import "styles.scss"
import "../other-styles.css"

/**
 * Let's say you've created widget and preview components for a custom image
 * gallery widget in separate files:
 */
import ImageGalleryWidget from "./image-gallery-widget.js"
import ImageGalleryPreview from "./image-gallery-preview.js"

/**
 * Register the imported widget:
 */
CMS.registerWidget(`image-gallery`, ImageGalleryWidget, ImageGalleryPreview)
```

`manualInit`

(*optional*, type: `boolean`, default: `false`)

Set this to `true` If you need to manually initialize Netlify CMS. The plugin will take care of setting `window.CMS_MANUAL_INIT` to `true`:

```
plugins: [
  {
    resolve: `gatsby-plugin-netlify-cms`,
    options: {
      manualInit: true,
    },
  },
]
```

The js module might look like this:

```
import CMS from "netlify-cms-app"

/**
 * Optionally pass in a config object. This object will be merged into `config.yml` if it e
 */

CMS.init({
  config: {
    backend: {
      name: "git-gateway",
    },
  },
})
```

`enableIdentityWidget`

(*optional*, type: `boolean`, default: `true`)

`enableIdentityWidget` is `true` by default, allowing Netlify Identity to be used without configuration. Disable it when not using Netlify Identity to reduce bundle size.

```
plugins: [
  {
    resolve: `gatsby-plugin-netlify-cms`,
    options: {
      enableIdentityWidget: true,
    },
  },
]
```

`publicPath`

(*optional*, type: `string`, default: `"admin"`)

Customize the path to Netlify CMS on your Gatsby site.

`htmlTitle`

(*optional*, type: `string`, default: `Content Manager`)

Customize the value of the `title` tag in your CMS HTML (shows in the browser bar).

`htmlFavicon`

(*optional*, type: `string`, default: `""`)

Customize the value of the `favicon` tag in your CMS HTML (shows in the browser bar).

`includeRobots`

(*optional*, type: `boolean`, default: `false`)

By default, the CMS page is not indexed by crawlers. Use this to add a `meta` tag to invite robots to index the CMS page.

`customizeWebpackConfig`

(*optional*, type: `function`)

Function to customize webpack configuration.

Function parameters:

- config: webpack configuration for NetlifyCMS
- destructured object from onCreateWebpackConfig { store, stage, pathPrefix, getConfig, rules, loaders, plugins} as seen in https://www.gatsbyjs.com/docs/node-apis/#onCreateWebpackConfig

```
plugins: [
  {
    resolve: `gatsby-plugin-netlify-cms`,
    options: {
      customizeWebpackConfig: (config, { plugins }) => {
        const Plugin = require("...")

        config.plugins.push(
          plugins.define({
            "process.env.MY_VAR": JSON.stringify("my var value"),
          })
        )

        config.plugins.push(new Plugin())
      },
    },
  },
]
```

Example

Here is the plugin with example values for all options (note that no option is required):

```
plugins: [
  {
    resolve: `gatsby-plugin-netlify-cms`,
    options: {
      modulePath: `path/to/custom/script.js`, // default: undefined
      enableIdentityWidget: true,
      publicPath: `admin`,
      htmlTitle: `Content Manager`,
      htmlFavicon: `path/to/favicon`,
      includeRobots: false,
    },
  },
]
```

Disable widget on site

If you're not using Netlify Identity within your site you have the option to completely disable the widget (and not the CMS). To do so, add the following to `gatsby-node.js`:

```
const webpack = require(`webpack`)

exports.onCreateWebpackConfig = ({ actions }) => {
  actions.setWebpackConfig({
    plugins: [
      new webpack.IgnorePlugin({
        resourceRegExp: /^netlify-identity-widget$/,
      }),
    ],
  })
}
```

Support

For help with integrating Netlify CMS with Gatsby, check out the community Slack.