

In this guide, you'll set up a site with content management using [Netlify CMS](#).

Netlify CMS is an open source, single page app written in React that lets you edit content and data files in your Git repository. Storing raw content right in the static site repository is an ideal approach, allowing both code and content to be versioned together, but that requires non-technical editors to interact with a service like GitHub. Netlify CMS was created specifically to bridge this gap, providing a solid interface that works well for technical and non-technical users alike, and interacts with your static site repository via API so that every change results in a commit.

A primary focus of Netlify CMS is to work well with modern site generators like Gatsby. Installation typically requires just an index.html file and a YAML configuration file, but you're going to leverage the Gatsby plugin for Netlify CMS to automatically install and build the CMS along with a static site.

Note: this guide uses the Gatsby Hello World starter to provide a very basic understanding of how Netlify CMS can work with your Gatsby site. If you get stuck, compare your code to the [example project](#). If you'd like to start with a full-blown template, check out [gatsby-starter-netlify-cms](#).

Setup

First, open a new terminal window and run the following to create a new site. This will create a new directory called `netlify-cms-tutorial` that contains the starter site, but you can change "netlify-cms-tutorial" in the command below to be whatever you like.

```
gatsby new netlify-cms-tutorial https://github.com/gatsbyjs/gatsby-starter-hello-world
```

Now move into the newly created directory and install the Gatsby plugin for Netlify CMS:

```
cd netlify-cms-tutorial && npm install netlify-cms-app gatsby-plugin-netlify-cms
```

Gatsby plugins are registered in a file called `gatsby-config.js` in the site root. Create that file if it's not already there, and add the following to register the Netlify CMS plugin:

```
module.exports = {
  plugins: [`gatsby-plugin-netlify-cms`],
}
```

Finally, you'll need to add a configuration file for the CMS itself. The plugin you just installed will take care of creating the Netlify CMS app and outputting it to `/admin/index.html`, so you'll want to put the configuration file in that same directory.

Still in the root directory, add a "static" folder. Gatsby will copy everything in the static folder into the output, so we'll want to place the Netlify CMS configuration file as `static/admin/config.yml`. Let's create a test configuration now - add this to your new `config.yml`:

```
backend:
  name: test-repo

media_folder: static/assets
public_folder: /assets
```

```
collections:
  - name: blog
    label: Blog
    folder: blog
    create: true
    fields:
      - { name: path, label: Path }
      - { name: date, label: Date, widget: datetime }
      - { name: title, label: Title }
      - { name: body, label: Body, widget: markdown }
```

Then in your terminal run `gatsby develop` to start the Gatsby development server. Once the server is running, it will print the address to open for viewing. It's typically `http://localhost:8000`. Open that in a browser and you should see the text "Hello World" in the top left corner. Now navigate to `/admin/` - so if your site is at `http://localhost:8000`, go to `http://localhost:8000/admin/`. **The trailing slash is required!**

You should now be viewing your Netlify CMS instance. You defined a "blog" collection in the configuration above, so you can create new blogs, but Netlify CMS will only store them in memory - if you refresh, your changes won't be there.

Saving to a Git repo

To save your content in a Git repo, the repo will need to be hosted on a service like GitHub, and you'll need a way to authenticate with that service so Netlify CMS can make changes through the service's API. For most services, including GitHub, authentication will require a server. [Netlify](#), the web platform company that started Netlify CMS, provides a very simple (and free) solution for this.

This is also a good time to consider that Netlify CMS is meant to work in production, as a part of your static site, and that the site would ideally be running on continuous deployment (every time the repo changes, the website is rebuilt and redeployed automatically). When used in production, Netlify CMS and your Gatsby site will stay synced, since your site will be rebuilt after each change, whereas running Netlify CMS locally requires you to pull changes from your remote each time to see them in the locally served site.

Pushing to GitHub

We can resolve all of the above handily by pushing our test site to GitHub and deploying it to Netlify. First, initialize your Gatsby project as a Git repo, and push it up to GitHub. If you need help on this part, check out GitHub's [guide](#).

Deploying to Netlify

Now you can publish your Gatsby site straight from GitHub to Netlify from the [create site page](#) - the proper build command for Gatsby will be provided automatically, just select your GitHub repo and go with the default options. Once you connect your GitHub repo to Netlify, deployment will begin. Note that the first deployment could take a few minutes since a lot of things aren't cached yet. Subsequent deploys will be faster.

Once deployment is complete you'll be able to view your live site, which should look the same as it did locally.

Authenticating with GitHub

Netlify CMS will need to authenticate with GitHub to save your content changes to your repo. As mentioned above, this requires a server, and Netlify handles that aspect for you. First you'll need to add your deployed site as an OAuth application in your GitHub settings - follow the steps in the [Netlify docs](#). This will allow scripts running on your deployed site, such as Netlify CMS, to access your GitHub account via API.

Lastly, we'll need to change your Netlify CMS config file with your GitHub repo information so that changes are saved there. Replace the `backend` value in your `static/admin/config.yml` to match the example below. Note that the `repo` value must be your GitHub username followed immediately by a forward slash, and then your repository name. If your username is "test-user" and your repo name is "test-repo", you would put "test-user/test-repo".

```
backend:
  name: github
  repo: your-username/your-repo-name
  branch: your-main-branch
```

Now you can save the `config.yml` file, commit the change, and push it to your GitHub repo.

Authenticating with GitLab

See the [GitLab Backend](#) section for details on how to configure authentication with GitLab.

If you use the [Client-Side Implicit Grant](#) option, disable the Netlify Identity service in your `gatsby-config.js` :

```
{
  resolve: `gatsby-plugin-netlify-cms`,
  options: {
    enableIdentityWidget: false,
  }
}
```

Making changes

Alright - you're all set to make changes in Netlify CMS and see them as commits in your GitHub repo! Open Netlify CMS on your deployed site at `/admin/`, allow access to GitHub when the permissions window pops up (check for blocked pop-ups if you don't see it), and try creating and publishing a new blog post. Once you've done that, you'll find a new "blog" directory in your GitHub repo containing a Markdown file with your blog post content!

This is the basic function of Netlify CMS - providing a comfortable editing experience and outputting raw content files to a Git repository. You've probably noticed that, even though the file was created in your repo, it's not anywhere on your site. That's because Netlify CMS doesn't go beyond creating the raw content - it's able to work with almost any static site generator because it allows the generator to determine how to build the raw content into something useful, whether for a website, mobile app, or something else entirely.

Right now, Gatsby doesn't know the new blog post is there, and it isn't set up to process Markdown. Let's fix that.

Processing Netlify CMS output with Gatsby

Gatsby can be configured to process Markdown by following the [Adding Markdown Pages](#) guide in the docs. Our `config.yml` file for Netlify CMS is set up to use the same fields used in the guide, so you can follow the instructions to the letter and should work fine. **Note:** When configuring the `gatsby-source-filesystem` plugin in the Adding Markdown Pages Guide, the path to your markdown files should be `${__dirname}/blog`.

Once this is complete, get your changes committed and pushed up to your GitHub repo and check your site! The new blog post will be at whatever you entered in the path field when creating your blog entry in Netlify CMS. If you

followed the example in Gatsby's Adding Markdown Pages guide and used `"/blog/my-first-blog"`, then your blog post would be at `"your-site-name.netlify.com/blog/my-first-blog"`.

Wrapping up

This was a very basic example meant to help you understand how Netlify CMS works with Gatsby. As mentioned in the beginning of this guide, if you got stuck, you can compare your code to the [example repo](#), which is a working example created by following this guide. You can also reach out to the Netlify CMS community on [Gitter](#). Lastly, if you'd like to move into a more complete boilerplate to get going with Gatsby and Netlify CMS, you can clone and



Deploy to Netlify

deploy the [official Gatsby Netlify CMS starter](#) to Netlify with

You can also see [blog posts about using Gatsby + Netlify CMS](#)