

# A Complete CMS with No Server and 18 Lines of Code

CMS TUTORIAL



Shawn Erquhart posted this on August 17, 2017

A few months ago we released an open source project, Netlify CMS. We've thoroughly enjoyed opening this up to the community for contributions (and we're always [looking for more](#)).

When you think of a CMS, you typically imagine a large application running on a server. Netlify CMS is different; it's a single page app written in React and built on Git. This means you can use a free GitHub repo as a database for your content.

The CMS is almost always used with a static site generator, like Hugo or Middleman, making it difficult for folks to understand where the CMS ends and the site generator begins, causing things to appear a bit "magical". In this post, we're going to set up the CMS locally in the most minimal way possible to help you better understand what it does.

Let's get started!

## Installing Locally

First we'll install the CMS locally:

1. Create a new local directory (does not need to be a Git repo).
2. Add a new file to that directory called `index.html` :

```
<head>
  <link href="https://unpkg.com/netlify-cms/dist/cms.css" rel="stylesheet"/>
</head>
<body>
  <script src="https://unpkg.com/netlify-cms/dist/cms.js"></script>
</body>
```

1. Add another file to that directory called `config.yml` :

```
backend:
  name: test-repo

media_folder: media_folder

collections:
  - name: post
    label: Post
    folder: posts
    create: true
    fields:
      - {name: title, label: Title}
      - {name: body, label: Body, widget: markdown}
```

If you serve that html file, you'll have a working CMS running locally. Here's a quick way to run a server:

1. Get [Node](#).
2. From your command line, run: `npm install -g http-server`
3. Still from the command line, move into the directory where you just created your CMS files.
4. Run this: `http-server -o -a localhost` .

**Note:** if you have a different way of serving, make sure you open the page via "localhost" (not 127.0.0.1). This is important because Netlify provides authentication services to your GitHub account during local development, but only if you're running on localhost.

Your default browser should open, and you'll be greeted with an input for entering your email address. You can do that for a bit of personalization, or just click "Log in".

You can create, save, and edit posts at this point - but it's all being saved in your browser's memory. If you refresh the page, it's gone. Let's set up a real backend.

## Using GitHub as a Content Database

You can store your content in any GitHub repo that has at least one commit, but let's create a new one:

1. Head over to the new repo page on [GitHub](#).
2. Make sure the repo is public, not private. (This is for tutorial purposes - private repos are also supported.)
3. Check the box labeled "Initialize this repository with a README" - this way your repo will have a commit, which is required.
4. Click "Create Repository" to create your new repo.

Now let's connect Netlify CMS to your repo. You'll need the path to your repo, which consists of your GitHub username and your repo name, separated by a slash. For example, my GitHub username is erquhart, and I have a test repo called "blank", so my repo path would be `erquhart/blank` . With your repo path

in hand, let's go back to our config.yml file and make a change:

```
backend:
  name: github
  repo: erquhart/blank
```

I changed the backend name to "github", so Netlify CMS knows we're using a GitHub repo, and I added my repo path.

## Making Changes

If you refresh the CMS page, you shouldn't see anything different, but the CMS is now running from your GitHub repo. It's able to check your repo for content without any authentication because your repo is public, but if you try to create a post and save, you'll get an error. Authentication requires a server for verification, which Netlify provides for users running the CMS locally under localhost.

Let's go ahead and authenticate with GitHub:

1. In the CMS page, click the settings icon in the top right, and select "Log Out" from the dropdown.
2. Close your browser tab and re-open the CMS page in a new tab.
3. You should now see a button labeled "Login With GitHub" - click it.
4. A popup will appear asking you to allow Netlify access to your repos. Confirm everything and you'll be logged in to the CMS again.

Finally, create a post and save it. Aaaaannd...that's it! You're using a web app as a CMS, and using GitHub as a content database! Let's peek at how things are working behind the scenes, shall we?

## What's Happening?

When you saved your post, you may have expected to find it in a file on your local machine, but that isn't the case - your post exists only in your remote git repo on GitHub. To save a change, the CMS converts your post to a Markdown string, and then sends a number of requests to GitHub's API, all resulting in a markdown file being added to your repo's master branch in a new commit. The Markdown file can then be used by any static site generator that can process Markdown!

## Next Steps

Note that this walkthrough was intentionally limited. Here's a few next steps to consider for experimenting with or implementing Netlify CMS:

- Create and deploy a full-fledged editable website with a blog in [minutes](#).
- Try the [editorial workflow](#) mode for drafts and publishing control.
- Check out the available [widgets](#) for the editor (you can also [add your own](#)).
- Netlify CMS is a community project - open an issue or pull request and make it even [better](#)!

Need help? Check out our [docs](#) or hit us up on the [Gitter](#).