

## Overview

In this guide, you'll walk through setting up a site with content management using [Forestry.io](https://forestry.io).

Forestry (also Forestry.io or Forestry CMS) is a Git-backed CMS that is built to work with static site generators<sup>[1]</sup>. There are no additional dependencies or plugins required to run Forestry with Gatsby apart from what is included in the [Gatsby Default Starter](https://github.com/gatsbyjs/gatsby-starter-default).

All configurations can be done through your Forestry site dashboard, but they can also be done directly in your Gatsby site repository.

To complete this tutorial, you will need a Forestry account. You can sign up at [Forestry.io](https://forestry.io).

*Note:* You can see the complete example at <https://github.com/cameron-yeo/gatsby-forestry-example>.

## Setup

First, clone the Gatsby default starter repository.

```
git clone https://github.com/gatsbyjs/gatsby-starter-default.git gatsby-forestry-example
```

Next, on GitHub create a new repository. The repository must live on a Git provider for it to be available to Forestry CMS. I'm using GitHub, but you can also use GitLab, Bitbucket, or Azure DevOps as well. Set your cloned repository's remote url to point to your newly created repository on GitHub.

```
git remote set-url origin <your-github-repo-link>
git push -u origin master
```

Now that your repository is on GitHub, follow the [Forestry.io get started guide](https://forestry.io/get-started-guide) on Forestry.io to connect the repository to Forestry. You have to allow Forestry access to your GitHub account to make this work. A pop-up window will automatically open when you select your GitHub repo in the Forestry site setup. Once the site is connected, go to the site settings on the Forestry dashboard. In the Admin Path setting, enter `"/static/admin"` and click "Deploy Admin". This will create a static HTML file for the Forestry CMS login page at the path: `/admin`.

Once Forestry is connected with your GitHub Gatsby site repository, pull Forestry's changes.

```
git pull
```

This will add a `.forestry` directory in your project root with the `settings.yml` file. This file allows you to configure your CMS settings including adding content collections. You should also see a new `/static/admin` directory that contains `index.html`. *Important:* The `/admin` directory must be located inside of `/static`. Do not rename this directory.

## Settings Configuration

Open `.forestry/settings.yml`. You will see the following default configuration values:

```
---
new_page_extension: md
auto_deploy: false
```

```
admin_path: /static/admin
webhook_url:
sections:
upload_dir: uploads
public_path: "/uploads"
front_matter_path: ""
use_front_matter_path: false
file_template: ":filename:"
build:
  preview_output_directory: public
  install_dependencies_command: npm install
  preview_docker_image: node:10
  mount_path: "/srv"
  working_dir: "/srv"
  instant_preview_command: npm run forestry:preview
```

The `new_page_extension` setting can either be `md` or `html`.

Change `upload_dir` and `public_path`:

```
upload_dir: static/uploads
public_path: "/static/uploads"
```

This sets where media is saved. Visit Gatsby's [guide on its static folder](#) to learn more. The rest of the settings can remain the same.

Open `package.json` and add the following to scripts already available:

```
"scripts": {
  "forestry:preview": "gatsby develop -p 8080 -H 0.0.0.0"
},
```

For preview to work on Forestry's dashboard, port 8080 must be used and all network interfaces have to be bound to `0.0.0.0`. For more information on Forestry's previews go to the [Forestry.io docs](#).

**Note:** The `forestry:preview` script can be named anything, but it must match the `instant_preview_command` setting in `.forestry/settings.yml`. This sets up Forestry to run a local server so that CMS content can be previewed from the Forestry dashboard using `gatsby develop`.

## CMS Content Configuration

Now you can set up a content collection. Create the file `.forestry/front_matter/blog.yml` and paste in the following:

```
---
label: Blog
hide_body: false
fields:
  - name: date
    type: datetime
    label: Date
```

```

description: ""
config:
  required: true
  date_format:
  time_format:
  display_utc: false
default: now
- name: title
  type: text
  label: Title
  description: ""
  config:
    required: true

```

Next, link the frontmatter collection in `.forestry/settings.yml`. This adds a template in the Forestry dashboard that allows you to add new Blog posts with the configured fields in the CMS.

```

---
new_page_extension: md
auto_deploy: false
admin_path: /static/admin
webhook_url:
upload_dir: static/uploads
public_path: "/static/uploads"
front_matter_path: ""
use_front_matter_path: false
file_template: ":filename:"
build:
  preview_output_directory: public
  install_dependencies_command: npm install
  preview_docker_image: node:10
  mount_path: "/srv"
  working_dir: "/srv"
  instant_preview_command: npm run forestry:preview
sections:
  - type: directory
    path: src/content/posts
    label: Posts
    create: documents
    match: "**/*"
    templates:
      - blog
---

```

Because Forestry's `admin.html` file is in the `/static` directory, this page will only be available once your Gatsby site is built. Run `gatsby build && gatsby serve`. Once the server is running, it will print the address to open for viewing. It's typically `http://localhost:8000`. Now navigate to `/admin` - so if your site is at `http://localhost:8000`, go to `http://localhost:8000/admin`

You should now be viewing your Forestry login page. Login to view your dashboard. If you don't have a user yet, create one on the Forestry Dashboard. This will allow you to login on the Gatsby site admin login page. On the left

side of your dashboard you will see "Posts". If you click on it you can add a new post. When you save your post, Forestry will push the change directly to your GitHub repo.

## Making Changes

You can now make changes in your Forestry admin dashboard and see them as commits in your GitHub repo! Open the Forestry dashboard 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 `content/posts` directory in your GitHub repo containing a Markdown file with your blog post content!

Now you can do whatever you want to with the CMS content. Here is the guide for creating pages from Markdown files in Gatsby: [Adding Markdown Pages](#). The docs also have a guide for doing this with MDX if you need to use JSX in your CMS content: [Writing Pages in MDX](#).

Both these guides explain the `gatsby-source-filesystem` plugin that Gatsby uses to locate markdown files.

## Wrapping Up

For more examples and help, visit the [Forestry.io docs](#).

*Note:* You can see the complete example at <https://github.com/cameron-yee/gatsby-forestry-example>.