

Adding content to your GitHub Pages site using Jekyll

In this article

About content in Jekyll sites

Adding a new page to your site

Adding a new post to your site

Next steps

You can add a new page or post to your Jekyll site on GitHub Pages.

GitHub Pages is available in public repositories with GitHub Free and GitHub Free for organizations, and in public and private repositories with GitHub Pro, GitHub Team, GitHub Enterprise Cloud, and GitHub Enterprise Server.

People with write permissions for a repository can add content to a GitHub Pages site using Jekyll.

About content in Jekyll sites

Before you can add content to a Jekyll site on GitHub Pages, you must create a Jekyll site. For more information, see "[Creating a GitHub Pages site with Jekyll](#)."

The main types of content for Jekyll sites are pages and posts. A page is for standalone content that isn't associated with a specific date, such as an "About" page. The default Jekyll site contains a file called `about.md`, which renders as a page on your site at `YOUR-SITE-URL/about`. You can edit the contents of that file to personalize your "About" page, and you can use the "About" page as a template to create new pages. For more information, see "[Pages](#)" in the Jekyll documentation.

A post is a blog post. The default Jekyll site contains a directory named `_posts` that contains a default post file. You can edit the contents of that post, and you can use the default post as a template to create new posts. For more information, see "[Posts](#)" in the Jekyll documentation.

Your theme includes default layouts, includes, and stylesheets that will automatically be applied to new pages and posts on your site, but you can override any of these defaults. For more information, see "[About GitHub Pages and Jekyll](#)."

To set variables and metadata, such as a title and layout, for a page or post on your site, you can add YAML front matter to the top of any Markdown or HTML file. For more information, see "[Front Matter](#)" in the Jekyll documentation.

If you are publishing from a branch, changes to your site are published automatically when the changes are merged into your site's publishing source. If you are publishing from a custom GitHub Actions workflow, changes are published whenever your workflow is triggered (typically by a push to the default branch). If you want to preview your changes first, you can make the changes locally instead of on GitHub Enterprise Server.

Then, test your site locally. For more information, see "[Testing your GitHub Pages site locally with Jekyll](#)."

Adding a new page to your site


- 1 On GitHub Enterprise Server, navigate to your site's repository.
- 2 Navigate to the publishing source for your site. For more information, see "[Configuring a publishing source for your GitHub Pages site](#)."
- 3 In the root of your publishing source, create a new file for your page called `PAGE-NAME.md`, replacing PAGE-NAME with a meaningful filename for the page.
- 4 Add the following YAML frontmatter to the top of the file, replacing PAGE-TITLE with the page's title and URL-PATH with a path you want for the page's URL. For example, if the base URL of your site is `https://octocat.github.io` and your URL-PATH is `/about/contact/`, your page will be located at `https://octocat.github.io/about/contact`.

```
layout: page
title: "PAGE-TITLE"
permalink: /URL-PATH
```

- 5 Below the frontmatter, add content for your page.
- 6 In the "Commit message" field, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message. For more information, see "[Creating a commit with multiple authors](#)."
- 7 Below the commit message fields, decide whether to add your commit to the current branch or to a new branch. If your current branch is the default branch, you should choose to create a new branch for your commit and then create a pull request. For more information, see "[Creating a pull request](#)."

☐ Commit directly to the `main` branch.

☒ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests](#).

 octocat-patch-1

Propose changes

Cancel

- 8 Click **Commit changes** or **Propose changes**.
- 9 Create a pull request for your proposed changes.
- 10 In the "Pull Requests" list, click the pull request you would like to merge.
- 11 Click **Merge pull request**. For more information, see "[Merging a pull request](#)."
- 12 If prompted, type a commit message, or accept the default message.
- 13 Click **Confirm merge**.
- 14 Optionally, delete the branch. For more information, see "[Creating and deleting branches within your repository](#)."

Adding a new post to your site


- 1 On GitHub Enterprise Server, navigate to your site's repository.
- 2 Navigate to the publishing source for your site. For more information, see "[Configuring a publishing source for your GitHub Pages site](#)."
- 3 Navigate to the `_posts` directory.
- 4 Create a new file called `YYYY-MM-DD-NAME-OF-POST.md`, replacing `YYYY-MM-DD` with the date of your post and `NAME-OF-POST` with the name of your post.
- 5 Add the following YAML frontmatter to the top of the file, including the post's title enclosed in quotation marks, the date and time for the post in `YYYY-MM-DD hh:mm:ss -0000` format, and as many categories as you want for your post.

```
layout: post
title: "POST-TITLE"
date: YYYY-MM-DD hh:mm:ss -0000
categories: CATEGORY-1 CATEGORY-2
```

- 6 Below the frontmatter, add content for your post.
- 7 In the "Commit message" field, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message. For more information, see "[Creating a commit with multiple authors](#)."
- 8 Below the commit message fields, decide whether to add your commit to the current branch or to a new branch. If your current branch is the default branch, you should choose to create a new branch for your commit and then create a pull request. For more information, see "[Creating a pull request](#)."

☐ Commit directly to the `main` branch.

☒ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests](#).

 octocat-patch-1

Propose changes

Cancel

- 9 Click **Commit changes** or **Propose changes**.
- 10 Create a pull request for your proposed changes.
- 11 In the "Pull Requests" list, click the pull request you would like to merge.
- 12 Click **Merge pull request**. For more information, see "[Merging a pull request](#)."
- 13 If prompted, type a commit message, or accept the default message.
- 14 Click **Confirm merge**.
- 15 Optionally, delete the branch. For more information, see "[Creating and deleting branches within your repository](#)."

Your post should now be up on your site! If the base URL of your site is `https://octocat.github.io`, then your new post will be located at

`https://octocat.github.io/YYYY/MM/DD/TITLE.html` .

Next steps

You can add a Jekyll theme to your GitHub Pages site to customize the look and feel of your site. For more information, see "[Adding a theme to your GitHub Pages site using Jekyll](#)."

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)