

Creating a GitHub Pages site with Jekyll

You can use Jekyll to create a GitHub Pages site in a new or existing repository.

Who can use this feature?

- 🔑 People with admin permissions for a repository can create a GitHub Pages site with Jekyll.
- 📁 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. For more information, see "[GitHub's plans](#)."

GitHub Pages now uses GitHub Actions to execute the Jekyll build. When using a branch as the source of your build, GitHub Actions must be enabled in your repository if you want to use the built-in Jekyll workflow. Alternatively, if GitHub Actions is unavailable or disabled, adding a `.nojekyll` file to the root of your source branch will bypass the Jekyll build process and deploy the content directly. For more information on enabling GitHub Actions, see "[Managing GitHub Actions settings for a repository](#)."

Mac **Windows** Linux

Note: Organization owners can restrict the publication of GitHub Pages sites from repositories owned by the organization. For more information, see "[Managing the publication of GitHub Pages sites for your organization](#)."

Prerequisites

Before you can use Jekyll to create a GitHub Pages site, you must install Jekyll and Git. For more information, see [Installation](#) in the Jekyll documentation and "[Set up Git](#)."

We recommend using [Bundler](#) to install and run Jekyll. Bundler manages Ruby gem dependencies, reduces Jekyll build errors, and prevents environment-related bugs. To install Bundler:

- 1 Install Ruby. For more information, see "[Installing Ruby](#)" in the Ruby documentation.
- 2 Install Bundler. For more information, see "[Bundler](#)."

Creating a repository for your site

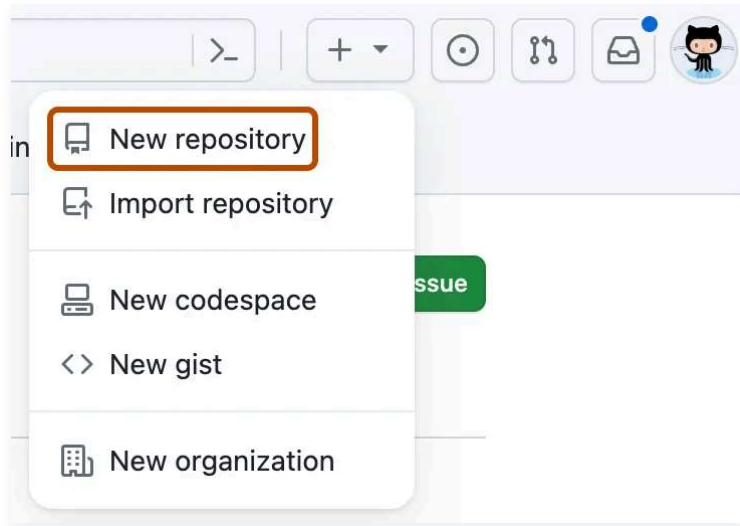
You can either create a repository or choose an existing repository for your site.

If you want to create a GitHub Pages site for a repository where not all of the files in the repository are related to the site, you will be able to configure a publishing source for your site. For example, you can have a dedicated branch and folder to hold your site source files, or you can use a custom GitHub Actions workflow to build and deploy your site source files.

If the account that owns the repository uses GitHub Free or GitHub Free for organizations, the repository must be public.

If you want to create a site in an existing repository, skip to the "[Creating your site](#)" section.

- 1 In the upper-right corner of any page, select **+**, then click **New repository**.

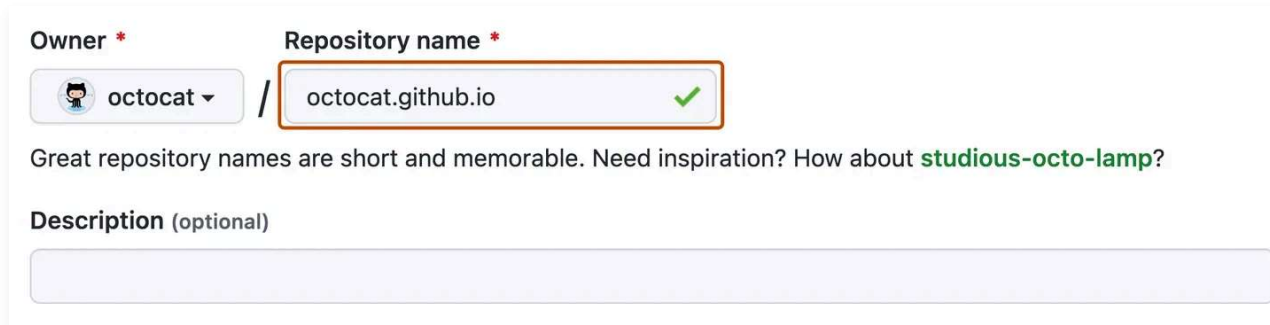


- 2 Use the **Owner** dropdown menu to select the account you want to own the repository.



- 3 Type a name for your repository and an optional description. If you're creating a user or organization site, your repository must be named `<user>.github.io` or

`<organization>.github.io` . If your user or organization name contains uppercase letters, you must lowercase the letters. For more information, see "[About GitHub Pages](#)."



Owner * Repository name *

octocat / octocat.github.io ✓

Great repository names are short and memorable. Need inspiration? How about **studious-octo-lamp**?

Description (optional)

- 4 Choose a repository visibility. For more information, see "[About repositories](#)."

Creating your site

Before you can create your site, you must have a repository for your site on GitHub. If you're not creating your site in an existing repository, see "[Creating a repository for your site](#)."

Warning: GitHub Pages sites are publicly available on the internet, even if the repository for the site is private (if your plan or organization allows it). If you have sensitive data in your site's repository, you may want to remove the data before publishing. For more information, see "[About repositories](#)."

- 1 Open Git Bash.
- 2 If you don't already have a local copy of your repository, navigate to the location where you want to store your site's source files, replacing PARENT-FOLDER with the folder you want to contain the folder for your repository.

```
cd PARENT-FOLDER
```

- 3 If you haven't already, initialize a local Git repository, replacing REPOSITORY-NAME with the name of your repository.

```
$ git init REPOSITORY-NAME  
> Initialized empty Git repository in /REPOSITORY-NAME/.git/  
# Creates a new folder on your computer, initialized as a Git repository
```

- 4 Change directories to the repository.

```
$ cd REPOSITORY-NAME  
# Changes the working directory
```

- 5 Decide which publishing source you want to use. For more information, see "[Configuring a publishing source for your GitHub Pages site](#)."
- 6 Navigate to the publishing source for your site. For more information, see "[Configuring a publishing source for your GitHub Pages site](#)." For example, if you chose to publish your site from the `docs` folder on the default branch, create and change directories to the `docs` folder.

```
$ mkdir docs  
# Creates a new folder called docs  
$ cd docs
```

If you chose to publish your site from the `gh-pages` branch, create and checkout the `gh-pages` branch.

```
$ git checkout --orphan gh-pages
# Creates a new branch, with no history or contents, called gh-pages, and
switches to the gh-pages branch
$ git rm -rf .
# Removes the contents from your default branch from the working directory
```

- 7 To create a new Jekyll site, use the `jekyll new` command:

```
$ jekyll new --skip-bundle .
# Creates a Jekyll site in the current directory
```

- 8 Open the Gemfile that Jekyll created.
- 9 Add `"#"` to the beginning of the line that starts with `gem "jekyll"` to comment out this line.
- 10 Add the `github-pages` gem by editing the line starting with `# gem "github-pages"`. Change this line to:

```
gem "github-pages", "~> GITHUB-PAGES-VERSION", group: :jekyll_plugins
```

Replace `GITHUB-PAGES-VERSION` with the latest supported version of the `github-pages` gem. You can find this version here: "[Dependency versions](#)."

The correct version Jekyll will be installed as a dependency of the `github-pages` gem.

- 11 Save and close the Gemfile.
- 12 From the command line, run `bundle install`.

- 13 Open the `.gitignore` file that Jekyll created and ignore the `gems.lock` file by adding this line:



GitHub Docs

Version: Free, Pro, & Team ▾

Search GitHub Docs



Sign up



GitHub Pages / Set up site with Jekyll / Create site with Jekyll

[a subfolder out into a new repository."](#)

```
domain: my-site.github.io      # if you want to force HTTPS, specify the domain
                                without the http at the start, e.g. example.com
url: https://my-site.github.io # the base hostname and protocol for your site,
                                e.g. http://example.com
baseurl: /REPOSITORY-NAME/    # place folder name if the site is served in a
                                subfolder
```

- 15 Optionally, test your site locally. For more information, see "[Testing your GitHub Pages site locally with Jekyll](#)."
- 16 Add and commit your work.

```
git add .
git commit -m 'Initial GitHub pages site with Jekyll'
```

- 17 Add your repository on GitHub.com as a remote, replacing `USER` with the account that owns the repository and `REPOSITORY` with the name of the repository.


```
$ git remote add origin https://github.com/USER/REPOSITORY.git
```

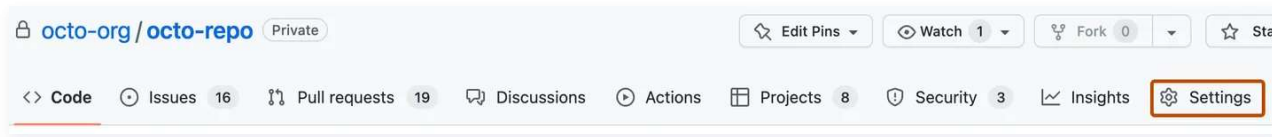
In this article



[Prerequisites](#)[Creating a repository for your site](#)[Creating your site](#)[Next steps](#)

- 18 Push the repository to GitHub, replacing BRANCH with the name of the branch you're working on.

```
git push -u origin BRANCH
```

- 19 Configure your publishing source. For more information, see "[Configuring a publishing source for your GitHub Pages site](#)."
- 20 On GitHub, navigate to your site's repository.
- 21 Under your repository name, click  **Settings**. If you cannot see the "Settings" tab, select the **...** dropdown menu, then click **Settings**.



- 22 In the "Code and automation" section of the sidebar, click  **Pages**.
- 23 To see your published site, under "GitHub Pages", click  **Visit site**.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://octocat.github.io/octo-repo/>

Last deployed by  octocat 3 hours ago

 Visit site



Note: It can take up to 10 minutes for changes to your site to publish after you push the changes to GitHub. If you don't see your GitHub Pages site changes reflected in your browser after an hour, see "[About Jekyll build errors for GitHub Pages sites](#)."

- 24 Your GitHub Pages site is built and deployed with a GitHub Actions workflow. For more information, see "[Viewing workflow run history](#)".

Note: GitHub Actions is free for public repositories. Usage charges apply for private and internal repositories that go beyond the monthly allotment of free minutes. For more information, see "[Usage limits, billing, and administration](#)".

Notes:

- If you are publishing from a branch and your site has not published automatically, make sure someone with admin permissions and a verified email address has pushed to the publishing source.
- Commits pushed by a GitHub Actions workflow that uses the `GITHUB_TOKEN` do not trigger a GitHub Pages build.

Next steps

To add a new page or post to your site, see "[Adding content to your GitHub Pages site using Jekyll](#)".

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

© 2024 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)