



Setting up your GitHub Pages site locally with Jekyll

MAC | [WINDOWS](#) | [LINUX](#)

You can set up a local version of your Jekyll GitHub Pages site to test changes to your site locally. We highly recommend installing Jekyll to preview your site and help troubleshoot failed Jekyll builds.

In this article:

[Requirements](#)

[Step 1: Create a local repository for your Jekyll site](#)

[Step 2: Install Jekyll using Bundler](#)

[Step 3 \(optional\): Generate Jekyll site files](#)

[Step 4: Build your local Jekyll site](#)

[Keeping your site up to date with the GitHub Pages gem](#)

[Next steps: Configuring Jekyll](#)

[Further reading](#)

Article versions

[GitHub.com](#)

[GitHub Enterprise 2.8](#)

[GitHub Enterprise 2.7](#)

[GitHub Enterprise 2.6](#)

[GitHub Enterprise 2.5](#)

[GitHub Enterprise 2.4](#)

Requirements

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, you must install [Ruby](#).

- 1 Open Terminal.
- 2 Check whether you have Ruby 2.1.0 or higher installed:
- 3 If you don't have Ruby installed, [install Ruby 2.1.0 or higher](#).

```
$ ruby --version
ruby 2.X.X
```

- 4 Install Bundler:

```
$ gem install bundler
# Installs the Bundler gem
```

- 5 If you already have a local repository for your Jekyll site, skip to [Step 2](#).

Step 1: Create a local repository for your Jekyll site

- 1 If you haven't already downloaded Git, install it. For more information, see "[Set up Git](#)."
- 2 Open Terminal.
- 3 On your local computer, initialize a new Git repository for your Jekyll site:

```
$ git init Jekyll-SITE-REPOSITORY-NAME
Initialized empty Git repository in /Users/octocat/my-site/.git/
# Creates a new file directory on your local computer, initialized as a Git repository
```

- 4 Change directories to the new repository you created:

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

- 5 If your new local repository is for a [Project pages site](#), create a new `gh-pages` branch:

Note: You can skip this step if you would rather use the `master` branch for your Project Page. If you haven't checked out any branches, once you make a commit in your local repository, your change will appear on the `master` branch by default.

```
$ git checkout -b gh-pages
Switched to a new branch 'gh-pages'
# Creates a new branch called 'gh-pages', and checks it out
```

Tip: To learn more about creating a User, Organization or Project Page and which branch to use, see "[User, Organization, and Project Pages](#)." To learn more about how to build your site's source files from a `/docs` folder on the `master` branch, see "[Configuring a publishing source for GitHub Pages](#)."

Step 2: Install Jekyll using Bundler

To track your site's dependencies, Ruby will use the contents of your Gemfile to build your Jekyll site.

- 1 Check to see if you have a Gemfile in your local Jekyll site repository:

```
$ ls
Gemfile
```

If you have a Gemfile, skip to step 4. If you don't have a Gemfile, skip to step 2.

- 2 If you don't have a Gemfile, open your favorite text editor, such as [Atom](#), and add these lines to a new file:

```
source 'https://rubygems.org'
gem 'github-pages', group: :jekyll_plugins
```

- 3 Name the file `Gemfile` and save it to the [root directory](#) of your local Jekyll site repository. Skip to step 5 to install Jekyll.
- 4 If you already have a Gemfile, open your favorite text editor, such as [Atom](#), and add these lines to your Gemfile:

```
source 'https://rubygems.org'
gem 'github-pages', group: :jekyll_plugins
```

5 Install Jekyll and other [dependencies](#) from the GitHub Pages gem:

```
$ bundle install
Fetching gem metadata from https://rubygems.org/.....
Fetching version metadata from https://rubygems.org/...
Fetching dependency metadata from https://rubygems.org/..
Resolving dependencies...
```

Tip: If you see a Ruby error when you try to install Jekyll using Bundler, you may need to use a package manager, such as RVM or Homebrew, to manage your Ruby installation. For more information, see Jekyll's [troubleshooting page](#).

Step 3 (optional): Generate Jekyll site files

To build your Jekyll site locally, preview your site changes, and troubleshoot build errors, you must have Jekyll site files on your local computer. You may already have Jekyll site files on your local computer if you cloned a Jekyll site repository. If you don't have a Jekyll site downloaded, you can generate Jekyll site files for a basic Jekyll template site in your local repository.

If you want to use an existing Jekyll site repository on GitHub as the starting template for your Jekyll site, fork and clone the Jekyll site repository on GitHub to your local computer. For more information, see "[Fork a repo](#)."

Note: As of Jekyll 3.2, the default Jekyll site contains a Gemfile that locks Jekyll to the Gem version you build it with. To instead lock it to the version used by GitHub Pages, you'll uncomment the `gem "github-pages", group :jekyll_plugins` line in the steps below.

1 If you don't already have a Jekyll site on your local computer, create a Jekyll template site in a new directory:

```
$ jekyll _3.3.0_ new NEW-JEKYLL-SITE-REPOSITORY-NAME
New jekyll site installed in /Users/octocat/NEW-JEKYLL-SITE-REPOSITORY-NAME.
```

2 Navigate into your new site directory:

```
$ cd NEW-JEKYLL-SITE-REPOSITORY-NAME
```

3 Edit your Gemfile and remove the following line:

```
$ "jekyll", "3.3.0"
```

4 In the Gemfile, delete the `#` at the beginning of this line:

```
$ gem "github-pages", group :jekyll_plugins
```

5 Initialize your site directory as a Git repository.

```
$ git init
Initialized empty Git repository in /Users/octocat/name-of-your-directory
```

- 6** Connect your remote repository on GitHub to your local repository for your GitHub Pages site.

```
$ git remote add origin https://github.com/username-or-organization-name/your-remote-repository-name
```

- 7** To edit the Jekyll template site, open your new Jekyll site files in a text editor. Make your changes and save them in the text editor. You can preview these changes locally on your computer without committing your changes using Git by [running a Jekyll command to build your site](#).

- 8** Add or stage your changes.

```
$ git add .
```

- 9** Commit your changes with a comment.

```
$ git commit -m "updated site"
```

- 10** To publish your changes on your GitHub Pages site, [push your changes](#) to your remote repository on GitHub.

```
$ git push -u origin master
```

For more information on using this Git workflow, see "Good Resources for Learning Git and GitHub" or see this [Git cheat sheet](#).

Step 4: Build your local Jekyll site

- 1** Navigate into the [root directory](#) of your local Jekyll site repository.

- 2** Run your Jekyll site locally:

```
$ bundle exec jekyll serve
Configuration file: /Users/octocat/my-site/_config.yml
   Source: /Users/octocat/my-site
  Destination: /Users/octocat/my-site/_site
Incremental build: disabled. Enable with --incremental
Generating...
done in 0.309 seconds.
Auto-regeneration: enabled for '/Users/octocat/my-site'
Configuration file: /Users/octocat/my-site/_config.yml
  Server address: http://127.0.0.1:4000/
Server running... press ctrl-c to stop.
```

- 3** Preview your local Jekyll site in your web browser at `http://localhost:4000`.

Keeping your site up to date with the GitHub Pages gem

Jekyll is an [active open source project](#) and is updated frequently. As the GitHub Pages server is updated, the software on your computer may become out of date, resulting in your site appearing different locally from how it looks when published on GitHub.

- 1 Open Terminal.
- 2 Run this update command:

If you followed our setup recommendations and installed [Bundler](#), run `bundle update github-pages` or simply `bundle update` and all your gems will update to the latest versions.

If you don't have Bundler installed, run `gem update github-pages`

Next steps: Configuring Jekyll

To configure your pages site further, see "[Configuring Jekyll](#)." To set up a project pages site, see [Jekyll's official documentation on project pages URLs](#).

Further Reading

["Troubleshooting GitHub Pages Builds"](#)

["Using Jekyll as a static site generator with GitHub Pages"](#)

[Jekyll's official GitHub Pages documentation](#)

[Jekyll commands](#)

 [Contact a human](#)

