# Troubleshooting Jekyll build errors for GitHub Pages sites

**In this article**

Troubleshooting build errors
Config file error
Date is not a valid datetime
File does not exist in includes directory
File is not properly UTF-8 encoded
Invalid highlighter language
Invalid post date
Invalid Sass or SCSS
Invalid submodule
Invalid YAML in data file
Markdown errors
Missing docs folder
Missing submodule
Relative permalinks configured
Syntax error in 'for' loop
Tag not properly closed
Tag not properly terminated
Unknown tag error

You can use Jekyll build error messages to troubleshoot problems with your GitHub Pages site.

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."

## Troubleshooting build errors

If Jekyll encounters an error building your GitHub Pages site locally or on GitHub Enterprise Cloud, you can use error messages to troubleshoot. For more information about error messages and how to view them, see "About Jekyll build errors for GitHub Pages sites."

If you received a generic error message, check for common issues.

- You're using unsupported plugins. For more information, see "About GitHub Pages and Jekyll."
- Your repository has exceeded our repository size limits. For more information, see "About large files on GitHub"
- You changed the `source` setting in your *_config.yml* file. If you publish your site from a branch, GitHub Pages overrides this setting during the build process.

- A filename in your published files contains a colon (`:`) which is not supported.

If you received a specific error message, review the troubleshooting information for the error message below.

After you've fixed any errors, trigger another build by pushing the changes to your site's source branch (if you are publishing from a branch) or by triggering your custom GitHub Actions workflow (if you are publishing with GitHub Actions).

## Config file error

This error means that your site failed to build because the *_config.yml* file contains syntax errors.

To troubleshoot, make sure that your *_config.yml* file follows these rules:

- Use spaces instead of tabs.
- Include a space after the `:` for each key value pair, like `timezone: Africa/Nairobi`.
- Use only UTF-8 characters.
- Quote any special characters, such as `:`, like `title: "my awesome site: an adventure in parse errors"`.
- For multi-line values, use `|` to create newlines and `>` to ignore newlines.

To identify any errors, you can copy and paste the contents of your YAML file into a YAML linter, such as YAML Validator.

**Note:** If your repository contains symbolic links, you will need to publish your site using a GitHub Actions workflow. For more information about GitHub Actions, see "GitHub Actions documentation."

## Date is not a valid datetime

This error means that one of the pages on your site includes an invalid datetime.

To troubleshoot, search the file in the error message and the file's layouts for calls to any date-related Liquid filters. Make sure that any variables passed into date-related Liquid filters have values in all cases and never pass `nil` or `""`. For more information, see "Liquid filters" in the Liquid documentation.

## File does not exist in includes directory

This error means that your code references a file that doesn't exist in your *_includes* directory.

To troubleshoot, search the file in the error message for `include` to see where you've referenced other files, such as `{% include example_header.html %}`. If any of the files you've referenced aren't in the *_includes* directory, copy or move the files into the *_includes* directory.

## File is not properly UTF-8 encoded

This error means that you used non-Latin characters, like ` `, without telling the computer to expect these symbols.

To troubleshoot, force UTF-8 encoding by adding the following line to your *_config.yml* file:

```
encoding: UTF-8
```

## Invalid highlighter language

This error means that you specified any syntax highlighter other than Rouge or Pygments in your configuration file.

To troubleshoot, update your *_config.yml* file to specify Rouge or Pygments. For more information, see "About GitHub Pages and Jekyll."

## Invalid post date

This error means that a post on your site contains an invalid date in the filename or YAML front matter.

To troubleshoot, make sure all dates are formatted as YYYY-MM-DD HH:MM:SS for UTC and are actual calendar dates. To specify a time zone with an offset from UTC, use the format YYYY-MM-DD HH:MM:SS +/-TTTT, like `2014-04-18 11:30:00 +0800`.

If you specify a date format in your *_config.yml* file, make sure the format is correct.

## Invalid Sass or SCSS

This error means your repository contains a Sass or SCSS file with invalid content.

To troubleshoot, review the line number included in the error message for invalid Sass or SCSS. To help prevent future errors, install a Sass or SCSS linter for your favorite text editor.

## Invalid submodule

This error means that your repository includes a submodule that hasn't been properly initialized.

To troubleshoot, first decide if you actually want to use a submodule, which is a Git project inside a Git project; submodules are sometimes created accidentally.

If you don't want to use a submodule, remove the submodule, replacing PATH-TO-SUBMODULE with the path to the submodule:

```
git submodule deinit PATH-TO-SUBMODULE
git rm PATH-TO-SUBMODULE
git commit -m "Remove submodule"
rm -rf .git/modules/PATH-TO-SUBMODULE
```

If do you want to use the submodule, make sure you use `https://` when referencing the submodule (not `http://`) and that the submodule is in a public repository.

## Invalid YAML in data file

This error means that one of more files in the *_data* folder contains invalid YAML.

To troubleshoot, make sure the YAML files in your *_data* folder follow these rules:

- Use spaces instead of tabs.
- Include a space after the `:` for each key value pair, like `timezone: Africa/Nairobi`.

- Use only UTF-8 characters.
- Quote any special characters, such as `:`, like `title: "my awesome site: an adventure in parse errors"`.
- For multi-line values, use `|` to create newlines and `>` to ignore newlines.

To identify any errors, you can copy and paste the contents of your YAML file into a YAML linter, such as YAML Validator.

For more information about Jekyll data files, see "Data Files" in the Jekyll documentation.

## Markdown errors

This error means that your repository contains Markdown errors.

To troubleshoot, make sure you are using a supported Markdown processor. For more information, see "Setting a Markdown processor for your GitHub Pages site using Jekyll."

Then, make sure the file in the error message uses valid Markdown syntax. For more information, see "Markdown: Syntax" on Daring Fireball.

## Missing docs folder

This error means that you have chosen the `docs` folder on a branch as your publishing source, but there is no `docs` folder in the root of your repository on that branch.

To troubleshoot, if your `docs` folder was accidentally moved, try moving the `docs` folder back to the root of your repository on the branch you chose for your publishing source. If the `docs` folder was accidentally deleted, you can either:

- Use Git to revert or undo the deletion. For more information, see "git-revert" in the Git documentation.
- Create a new `docs` folder in the root of your repository on the branch you chose for your publishing source and add your site's source files to the folder. For more information, see "Creating new files."
- Change your publishing source. For more information, see "Configuring a publishing source for your GitHub Pages site."

## Missing submodule

This error means that your repository includes a submodule that doesn't exist or hasn't been properly initialized.

To troubleshoot, first decide if you actually want to use a submodule, which is a Git project inside a Git project; submodules are sometimes created accidentally.

If you don't want to use a submodule, remove the submodule, replacing PATH-TO-SUBMODULE with the path to the submodule:

```
git submodule deinit PATH-TO-SUBMODULE
git rm PATH-TO-SUBMODULE
git commit -m "Remove submodule"
rm -rf .git/modules/PATH-TO-SUBMODULE
```

If you do want to use a submodule, initialize the submodule. For more information, see "Git Tools - Submodules" in the *Pro Git* book.

## Relative permalinks configured

This errors means that you have relative permalinks, which are not supported by GitHub Pages, in your *_config.yml* file.

Permalinks are permanent URLs that reference a particular page on your site. Absolute permalinks begin with the root of the site, while relative permalinks begin with the folder containing the referenced page. GitHub Pages and Jekyll no longer support relative permalinks. For more information about permalinks, see "Permalinks" in the Jekyll documentation.

To troubleshoot, remove the `relative_permalinks` line from your *_config.yml* file and reformat any relative permalinks in your site with absolute permalinks. For more information, see "Editing files."

## Syntax error in 'for' loop

This error means that your code includes invalid syntax in a Liquid `for` loop declaration.

To troubleshoot, make sure all `for` loops in the file in the error message have proper syntax. For more information about proper syntax for `for` loops, see "Iteration tags" in the Liquid documentation.

## Tag not properly closed

This error message means that your code includes a logic tag that is not properly closed. For example, `{% capture example_variable %}` must be closed by `{% endcapture %}`.

To troubleshoot, make sure all logic tags in the file in the error message are properly closed. For more information, see "Liquid tags" in the Liquid documentation.

## Tag not properly terminated

This error means that your code includes an output tag that is not properly terminated. For example, `{{ page.title }` instead of `{{ page.title }}`.

To troubleshoot, make sure all output tags in the file in the error message are terminated with `}}`. For more information, see "Liquid objects" in the Liquid documentation.

## Unknown tag error

This error means that your code contains an unrecognized Liquid tag.

To troubleshoot, make sure all Liquid tags in the file in the error message match Jekyll's default variables and there are no typos in the tag names. For a list of default variables, see "Variables" in the Jekyll documentation.

Unsupported plugins are a common source of unrecognized tags. If you use an unsupported plugin in your site by generating your site locally and pushing your static files to GitHub Enterprise Cloud, make sure the plugin is not introducing tags that are not in Jekyll's default variables. For a list of supported plugins, see "About GitHub Pages and Jekyll."