

# Using YAML frontmatter

## In this article

- About YAML frontmatter
- YAML frontmatter values
- Escaping single quotes
- Autogenerated mini TOCs
- Filenames
- Index pages
- Homepage
- Creating new product guides pages

You can use YAML frontmatter to define versioning, add metadata, and control the layout for articles.

Articles in the "Contributing to GitHub Docs" section refer to the documentation itself and are a resource for GitHub staff and open source contributors.

## About YAML frontmatter

YAML frontmatter is an authoring convention popularized by Jekyll that provides a way to add metadata to pages. It is a block of key-value content that lives at the top of every Markdown file within GitHub Docs. For more information, see the [YAML frontmatter documentation](#).

## YAML frontmatter values

The following frontmatter values have special meanings and requirements for GitHub Docs. There's also a schema that's used by the test suite to validate every page's frontmatter. For more information, see [lib/frontmatter.js](#).

- [versions](#)
- [redirect\\_from](#)
- [title](#)
- [shortTitle](#)
- [intro](#)
- [permissions](#)
- [product](#)
- [layout](#)
- [children](#)
- [childGroups](#)
- [featuredLinks](#)
- [showMiniToc](#)
- [allowTitleToDifferFromFilename](#)

- [changelog](#)
- [defaultPlatform](#)
- [defaultTool](#)
- [learningTracks](#)
- [includeGuides](#)
- [type](#)
- [topics](#)
- [communityRedirect](#)
- [effectiveDate](#)

## versions

- Purpose: Indicates the [versions](#) to which a page applies. For more information about the different types of versioning, see "[Versioning documentation](#)."
- Type: `Object`. Allowable keys map to product names and can be found in the `versions` object in [lib/frontmatter.js](#).
- This frontmatter value is currently **required** for all pages.
- The `*` is used to denote all releases for the version.

This frontmatter value is used by the docs site to generate "permalinks" for each version of an article. For more information, see [Permalinks](#).

Example that applies to GitHub.com and recent versions of GitHub Enterprise Server:

```
title: About your personal dashboard
versions:
  fpt: '*'
  ghes: '>=2.20'
```

Example that applies to all supported versions of GitHub Enterprise Server, but not GitHub.com:

```
title: Downloading your license
versions:
  ghes: '*'
```

You can also version a page for a range of releases. This would version the page for GitHub.com, and GitHub Enterprise Server versions 2.22 and 3.0 only:

```
versions:
  fpt: '*'
  ghes: '>=2.22 <3.1'
```

## redirect\_from

- Purpose: List URLs that should redirect to this page.
- Type: `Array`
- Optional

Example:

```
title: Getting started with GitHub Desktop
redirect_from:
  - /articles/first-launch/
  - /articles/error-github-enterprise-version-is-too-old/
  - /articles/getting-started-with-github-for-windows/
```

For more information, see "[Configuring redirects](#)."

## title

- Purpose: Set a human-friendly title for use in the rendered page's `<title>` tag and an `h1` element at the top of the page.
- Type: `String`
- Optional. If omitted, the page `<title>` will still be set, albeit with a generic value like `GitHub.com` or `GitHub Enterprise`.

## shortTitle

- Purpose: An abbreviated variant of the page title for use in breadcrumbs and navigation elements.
- Type: `String`
- Optional. If omitted, `title` will be used.

Article type	Maximum character length
articles	31
categories	27
map topics	30

Example:

```
title: Contributing to projects with GitHub Desktop
shortTitle: Contributing to projects
```

## intro

- Purpose: Sets the intro for the page. This string will render after the `title`.
- Type: `String`
- Optional.

## permissions

- Purpose: Sets the permission statement for the article. This string will render after the `intro`.
- Type: `String`
- Optional.

## product

- Purpose: Sets the product callout for the article. This string will render after the `intro` and `permissions` statement.
- Type: `String`
- Optional.

## layout

- Purpose: Render the proper page layout.
- Type: `String` that matches the name of the layout. For a layout named

`components/landing` , the value would be `product-landing` .

- Optional. If omitted, `DefaultLayout` is used.

## children

- Purpose: Lists the relative links that belong to the product/category/map topic. For more information, see [Index pages](#).
- Type: `Array` . Default is `false` .
- Required on `index.md` pages.

## childGroups

- Purpose: Renders children into groups on the homepage. For more information, see [Homepage](#).
- Type: `Array` . Default is `false` .
- Require on the homepage `index.md` .

## featuredLinks

- Purpose: Renders the linked articles' titles and intros on product landing pages and the homepage.
- Type: `Object` .
- Optional.

The list of popular links are the links displayed on the landing page under the title "Popular." Alternately, you can customize the title "Popular" by setting the `featuredLinks.popularHeading` property to a new string.

Example:

```
featuredLinks:
  gettingStarted:
    - /path/to/page
  startHere:
    - /guides/example
  popular:
    - /path/to/popular/article1
    - /path/to/popular/article2
  popularHeading: An alternate heading to Popular
```

## showMiniToc

- Purpose: Indicates whether an article should show a mini table of contents (TOC) above the rest of the content. For more information, see [Autogenerated mini TOCs](#).
- Type: `Boolean` . Default is `true` on articles, and `false` on map topics and `index.md` pages.
- Optional.

## allowTitleToDifferFromFilename

- Purpose: Indicates whether a page is allowed to have a title that differs from its filename. For example, `content/rest/reference/orgs.md` has a title of `Organizations` instead of `Orgs` . Pages with this frontmatter set to `true` will not be flagged in tests or updated by `script/reconcile-ids-with-filenames.js` .
- Type: `Boolean` . Default is `false` .
- Optional.

## changelog

- Purpose: Render a list of items pulled from [GitHub Changelog](#) on product landing pages ( `components/landing` ). The one exception is Education, which pulls from <https://github.blog/category/community/education>.
- Type: `Object` , properties:
  - `label` -- must be present and corresponds to the labels used in the [GitHub Changelog](#)
  - `prefix` -- optional string that starts each changelog title that should be omitted in the docs feed. For example, with the prefix `GitHub Actions:` specified, changelog titles like `GitHub Actions: Some Title Here` will render as `Some Title Here` in the docs feed.
- Optional.

## defaultPlatform

- Purpose: Override the initial platform selection for a page. If this frontmatter is omitted, then the platform-specific content matching the reader's operating system is shown by default. This behavior can be changed for individual pages, for which a manual selection is more reasonable. For example, most GitHub Actions runners use Linux and their operating system is independent of the reader's operating system.
- Type: `String` , one of: `mac` , `windows` , `linux` .
- Optional.

Example:

```
defaultPlatform: linux
```

## defaultTool

- Purpose: Override the initial tool selection for a page, where the tool refers to the application the reader is using to work with GitHub (such as GitHub.com's web UI, the GitHub CLI, or GitHub Desktop) or the GitHub APIs. For more information about the tool selector, see "[Using Markdown and Liquid in GitHub Docs](#)." If this frontmatter is omitted, then the tool-specific content matching the GitHub web UI is shown by default. If a user has indicated a tool preference (by clicking on a tool tab), then the user's preference will be applied instead of the default value.
- Type: `String` , one of: `webui` , `cli` , `desktop` , `curl` , `codespaces` , `vscode` , `importer_cli` , `graphql` , `powershell` , `bash` , `javascript` .
- Optional.

```
defaultTool: cli
```

## learningTracks

- Purpose: Render a list of learning tracks on a product's sub-landing page.
- type: `String` . This should reference learning tracks' names defined in [data/learning-tracks/\\*.yaml](#) .
- Optional

**Note:** the featured track is set by a specific property in the learning tracks YAML. See that [README](#) for details.

## includeGuides

- Purpose: Render a list of articles, filterable by `type` and `topics` . Only applicable when used with `layout: product-guides` .
- Type: `Array`
- Optional.

Example:

```
includeGuides:  
  - /actions/guides/about-continuous-integration  
  - /actions/guides/setting-up-continuous-integration-using-workflow-templates  
  - /actions/guides/building-and-testing-nodejs  
  - /actions/guides/building-and-testing-powershell
```

## type

- Purpose: Indicate the type of article.
- Type: `String` , one of the `overview` , `quick_start` , `tutorial` , `how_to` , `reference` .
- Optional.

## topics

- Purpose: Indicate the topics covered by the article. The topics are used to filter guides on some landing pages. For example, the guides at the bottom of "[Guides for GitHub Actions](#)" can be filtered by topics, and the topics are listed under the guide intro. Refer to the content models for more details about adding topics. A full list of existing topics is located in the [allowed topics file](#). If topics in article frontmatter and the allow-topics list become out of sync, the [topics CI test](#) will fail.
- Type: Array of `String` s
- Optional: Topics are preferred for each article, but, there may be cases where existing articles don't yet have topics, or adding a topic to a new article may not add value.

## communityRedirect

- Purpose: Set a custom link and link name for `Ask the GitHub community` link in the footer.
- Type: `Object` . Properties are `name` and `href` .
- Optional.

## effectiveDate

- Purpose: Set an effective date for Terms of Service articles so that engineering teams can automatically re-prompt users to confirm the terms
- Type: `string` YEAR-MONTH-DAY e.g. 2021-10-04 is October 4th, 2021
- Optional.

**Note:** The `effectiveDate` frontmatter value is for use by GitHub staff only.

## Escaping single quotes

If you see two single quotes in a row ( `''` ) in YAML frontmatter where you might expect to see one ( `'` ), this is the YAML-preferred way to escape a single quote.

As an alternative, you can change the single quotes surrounding the frontmatter field to double quotes and leave interior single quotes unescaped.

## Autogenerated mini TOCs

---

Every article displays a mini table of contents (TOC), which is an autogenerated "In this article" section that includes links to all `H2` s in the article. Only `H2` headers are included in the mini TOCs. If an article uses `H3` or `H4` headers to divide information in a way that only certain sections are relevant to a particular task, you can help people navigate to the content most relevant to them by using a [sectional TOC](#).

You can use the `showMiniToc` frontmatter value, set to `false`, to prevent the mini TOC from showing up for an article.

Mini TOCs do not appear on product landing pages, category landing pages, or map topic pages.

Do not add hardcoded "In this article" sections in the Markdown source or else the page will display duplicate mini TOCs.

## Filenames

---

When adding a new article, make sure the filename is a [kebab-cased](#) version of the title you use in the article's `title` frontmatter. This can get tricky when a title has punctuation (such as "GitHub's Billing Plans"). A test will flag any discrepancies between title and filename. To override this requirement for a given article, you can add `allowTitleToDifferFromFilename` frontmatter.

## Index pages

---

Index pages are the Table of Contents files for the docs site. Every product, category, and map topic subdirectory has an `index.md` that serves as the landing page. Each `index.md` must contain a `children` frontmatter property with a list of relative links to the child pages of the product, category, or map topic.

**Note:** The site only knows about paths included in `children` frontmatter. If a directory or article exists but is **not** included in `children`, its path will return a 404.

## Homepage

---

The homepage is the main Table of Contents file for the docs site. The homepage must have a complete list of `children`, like every [Index page](#) but must also specify the `childGroups` frontmatter property that will be highlighted in the main content area.

`childGroups` is an array of mappings containing a `name` for the group, an optional `icon` for the group, and an array of `children`. The `children` in the array must be present in the `children` frontmatter property.

## Creating new product guides pages

---

To create a product guides page (e.g. [GitHub Actions Guide page](#)), create or modify an existing markdown file with these specific frontmatter values:

- Use the product guides page template by referencing `layout: product-guides`.
- Include the learning tracks in `learningTracks`. Optional.

- Define which articles to include with `includeGuides` . Optional.

If using learning tracks, they need to be defined in `data/learning-tracks/*.yaml` . If using `includeGuides` , make sure each of the articles in this list has `topics` and `type` in its frontmatter.

## Legal

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