About the content model

In this article

The content model describes the structure and types of content that we publish.

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

Our content model explains the purpose of each type of content we create within GitHub Docs, and what to include when you write or update an article. We use a content model to ensure that our content consistently, clearly, and comprehensively communicates the information that people need to achieve their goals with GitHub.

We use these types across all documentation sets to provide a consistent user experience––any content type applies to any product or audience. Our content types evolve over time and we add new types as needed. We only publish content that follows the model.

Consistency helps people form mental models of the documentation and understand how to find the information they need as they return to GitHub Docs over time. It is also more efficient to maintain and update consistent content, making it easier and quicker to contribute to docs whether you are an open source contributor making your first commit or a writer on the GitHub staff documenting an entire new product.

Content structure

Docs are organized into multiple levels of hierarchy on our site.

- Top-level doc set
  - Categories
    - Map topics
      - Articles

Homepage content

The GitHub Docs homepage, docs.github.com, highlights the most important topics that

people want to find. We limit the number of doc sets on the homepage so that people can find information and the homepage does not become overcrowded and difficult to search.

The homepage includes all top-level doc sets and some categories. Content on the homepage is organized around GitHub concepts and practices. For example, the "CI/CD and DevOps" group includes top-level doc sets for GitHub Actions, GitHub Packages, and GitHub Pages.

Adding a doc set to the homepage

The goal of the homepage is to help people find information about the GitHub feature or product that they want to learn about. Every item added to the homepage dilutes the discoverability of every other item, so we limit the number of doc sets included on the homepage.

If a new top-level doc set is created, it is added to the homepage.

If a category serves as the starting point for using a GitHub product or feature, it can be added to the homepage.

For example, under the "Security" grouping on the homepage, in addition to the "Code security" top-level doc set, the "Supply chain security," "Security advisories," "Dependabot," "Code scanning," and "Secret scanning" categories are included because each of those categories are the entry point to GitHub products and features. "Security overview" is not included on the homepage because it provides additional information for using code security products and is not an introduction to a product or feature.

Top-level doc set

Top-level doc sets are organized around a GitHub product, feature, or core workflow. All top-level doc sets appear on the GitHub Docs homepage. You should only create a top-level doc set when there is a large amount of content to be contained in the new doc set, multiple categories that are broken down into map topics, and the topic applies across products, features, or account types. If the content could fit in any existing top-level doc set, it probably belongs in that existing doc set.

- Top-level doc sets are of roughly equal importance to one another (each is centered on a GitHub product or major feature).
- Most top-level doc sets have a landing page layout, unless there is a significant exception. For example, the "Site policy" doc set does not have guides or procedural articles like other doc sets, so it does not use a landing page layout.

Titles for top-level doc sets

- Feature or product based.
- Describes what part of GitHub someone is using.
- Examples
  - Organizations and teams documentation
  - GitHub Issues documentation

Category

Categories are organized around a feature or a discrete set of tasks within a top-level doc set aligned with product themes. A category's subject is narrow enough that its contents are manageable and does not grow too large to use. Some categories appear on the homepage.

- Categories often start small and grow with the product.
- Large categories may contain map topics to subdivide content around more specific user journeys or tasks.
- Use long procedural articles to group related chunks of content and keep articles within the category streamlined.
- When categories have more than ten articles, consider breaking the content into map topics or additional categories.

Titles for categories

- Task-based (begins with a gerund).
- Describes the big-picture purpose or goal of using the feature or product.
- General or high-level enough to scale with future product enhancements.
- Category titles must be 67 characters or shorter and have a `shortTitle` less than 27 characters.
- Examples
  - Setting up and managing your personal account on GitHub
  - Committing changes to your project

Intros for categories

All categories have intros. Intros should be one sentence long and general or high-level enough to scale with future product changes. If you significantly change a category's structure, check its intro for needed updates.

Map topic

Map topics introduce a section of a category, grouping articles within a category around more specific workflows or subjects that are part of the category's larger task.

Map topics contain at least three articles. When map topics have more than eight articles, it may be useful to consider breaking the content into more specific map topics.

Titles for map topics

- Task-based (begins with a gerund).
- Describes a more specific task within the larger workflow of the category it's in.
- General or high-level enough to scale with future additions to the product.
- Map topic titles must be 63 characters or shorter and have a `shortTitle` less than 30 characters.
- Examples
  - Understanding your software supply chain
  - Managing users in your enterprise

Intros for map topics

All map topics have intros. Intros should be one sentence long and general or high-level enough to scale with future product changes. If you add or remove articles in a map topic, check its intro for needed updates.

Article

An article is the basic unit of content for GitHub Docs––while we use multiple content types, they are all published as articles. Each content type has its own purpose, format, and structure, yet we use standard elements in every article type, like intros, to ensure articles provide a consistent user experience.

Content order

We organize content predictably within categories, map topics, and articles. From broadest applicability to most specific, narrow, or advanced information, following this order:

- Conceptual content
- Referential content
- Procedural content for enabling a feature or setting
- Procedural content on using a feature
- Procedural content on managing a feature or setting
- Procedural content on disabling a feature or setting
- Procedural content on destructive actions (e.g. deletion)
- Troubleshooting information

Reusing content

We use reusable and variable strings to use the same chunk of content, such as a procedural step or a conceptual paragraph, in multiple places. We generally don't reuse large sections of articles without a specific reason. When an entire section of an article might be relevant in more than one article, take a look at the purpose of both. Is there an opportunity to create a single, long-form article? Refer to the content models to clarify the best permanent home for the information, and link to it from the other article.