

Writing content to be translated

In this article

About writing content that is translation-friendly

Examples

Our documentation is translated into multiple languages. How we approach writing the English language documentation can greatly improve the quality of those translations.

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 writing content that is translation-friendly

Use the following guidelines to ensure the content you create can be successfully translated. For more information, see "[Style guide](#)."

- Use examples that are generic and can be understood by most people.
- Avoid examples that are controversial or culturally specific to a group.
- Write in active voice.
- Write simple, short, and easy-to-understand sentences.
- Avoid using too many pronouns that can make text unclear.
- Avoid using slang and jokes.
- Avoid negative sentences.
- Use industry-standard acronyms whenever possible and explain custom acronyms.
- Use indicative mood.
- Eliminate redundant and wordy expressions.
- Avoid the excessive use of stacked modifiers (noun strings). The translator can misunderstand which noun is the one being modified.
- Avoid invisible plurals in which it is not clear if the first noun is meant to be singular or plural.
- Avoid nominalization.
- Avoid using ambiguous modal auxiliary verbs.
- Avoid gender-specific words.
- Avoid prepositional phrases.
- Avoid vague nouns and pronouns (vague sentence subject).
- Keep inline links to a minimum. If they are necessary, preface them with a phrase such as "For more information, see "Link title." Alternatively, add relevant links to a "Further reading" section at the end of the topic.

Examples

This section provides examples for how to follow our guidelines for writing translation-

friendly documentation.

Avoid country specific information [↗](#)

For example, avoid 800 numbers and country specific addresses. If necessary, mention what countries the information applies to.

Avoid the excessive use of stacked modifiers (noun strings) [↗](#)

Lots of stacked modifiers can lead to incorrect translations because it's not easy to determine what modifies what. For example, use "Default source settings for the public repository" instead of "public repository default source settings."

Avoid invisible plurals [↗](#)

For example, when using the term "file retrieval," it's unclear if you are retrieving one file or all files. Provide more context to eliminate ambiguity. In the example given, you can use "retrieving all the files" or "retrieving the source.md file."

Avoid nominalization [↗](#)

For example, instead of "to reach a conclusion," use "conclude."

Avoid using ambiguous modal auxiliary verbs. [↗](#)

Avoid using words such as "may" and "might." Be more clear to avoid ambiguity.

Avoid prepositional phrases [↗](#)

Instead of writing "after trying many times" or "according to the repository log," write more directly. For example, "after trying three times."

Avoid vague nouns and pronouns [↗](#)

Vague nouns and pronouns can make it unclear who or what you are referring to, especially when that content has to be translated. For example, "Maintainers and contributors have access to files and comments. In the pull request they make changes to it." In this sentence, it is not clear if the changes are being made to the file or comments. If a pronoun seems to refer to more than one antecedent, either reword the sentence to make the antecedent clear or replace the pronoun with a noun to eliminate ambiguity.

Keep inline links to a minimum. [↗](#)

Where possible, clearly introduce links following our style guide. For example, instead of the following sentence:

Read [\[more about OAuth2.\]\(/apps/building-integrations/setting-up-and-registering-oauth-apps/\)](#) Note that OAuth2 tokens can be [\[acquired programmatically\] \(/rest/reference/oauth-authorizations/#create-a-new-authorization\)](#), for applications that are not websites.

You can use this instead:

OAuth2 tokens can be acquired programmatically for applications that are not websites. For more information, see "[[Setting up and registering OAuth Apps](#)]"

```
(apps/building-integrations/setting-up-and-registering-oauth-apps/) " and "[Create  
a new authorization](/rest/reference/oauth-authorizations/#create-a-new-  
authorization)."
```

Legal

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