Best practices for technical documentation
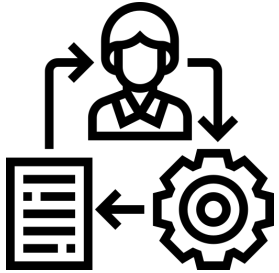
Tricia Burmeister (TBurmeister_WMF)
Created for WMF Data Documentathon
October 2022

WIKIMEDIA
FOUNDATION

Three major steps

Consider your audience → **Write or update content** → **Polish and improve**

Consider your audience.

What are they trying to achieve?

Critical user journeys

To help users complete their task quickly and easily, you must understand their **roles** and their **goals**:

- Who are your users? What are the different roles they may have when they need to use your docs?
- What would they be trying to do when occupying different roles?

Organizational silos often manifest as findability barriers, adding more steps for users and complicating their journey.

The challenge is to make your docs reflect your users' needs and goals, rather than your internal team structure or technical architecture.

User journey exercise!

I am a student interested in researching how people access wikis during important events. For example, I want to see if there's an increase in mobile views and edits on wikis during mass protests or strikes.

What might this user already know or not know?
What misconceptions might they have?

- I know some Python and R, but am still learning.
- I think the data I need is something like "mobile page views/edits" by date.
- I don't yet know that some wikis are multilingual while others are not.
- I have never edited a wiki myself, nor have I been involved in any other wiki projects.

Focus on readers' tasks

Goal: **Find and get access to the right data for my research question.**

Steps:
Understand the data models and types of data available
  - ...
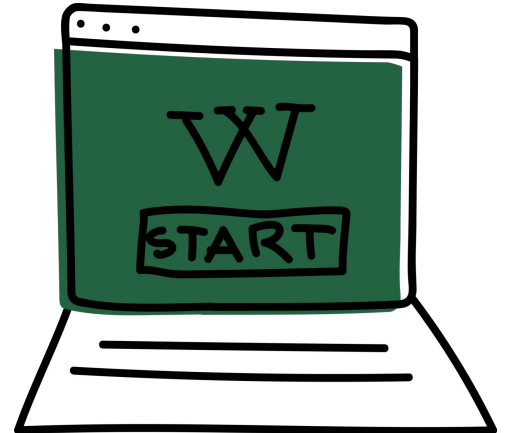
Verify that data is available in a format I can work with
  - ...

Identify the specific dataset(s) that may answer my question
  - ...

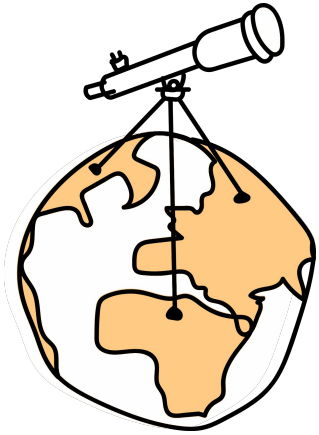Set up an account / obtain access to the data
  - ....

Write and update.

Focusing on your audience helps you know what to include, and what to remove.

Updating existing content

Ensure everything is accurate and correct. Test code examples to verify they actually work.

Fill in content gaps.

Add connections to prerequisites and next steps.

Simplify: Remove anything that isn't essential for your audience or their tasks!

Creating new content

Pretend you're explaining it to someone who's a member of one of your key audiences.

Start with prerequisites, end with next steps.

Create an outline and then fill it in.

Write, then edit.

Include real examples, like working API calls with explanations of the request and sample responses.

Polish and improve.

Style, consistency, and accessibility make docs more user-friendly and maintainable.

Style helps comprehension

Plain language is the most understandable by diverse audiences, and is easiest to translate.

Inclusive and neutral voice and tone helps readers feel welcome and included, regardless of their experience or background.

Active voice and imperative mood improve clarity and make docs easier to understand.
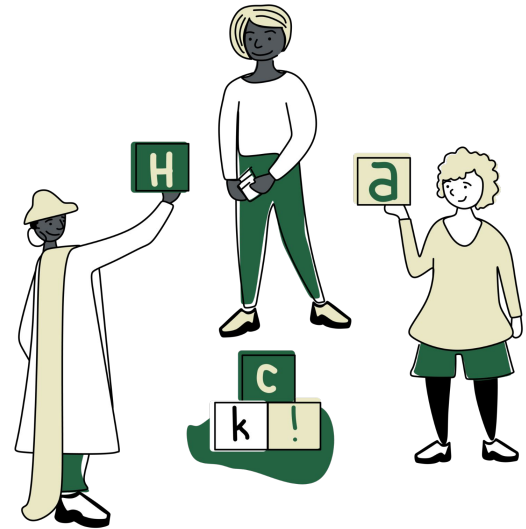
Style exercise!

Rewrite this sentence to use plain language:

"There is no hard and fast limit on API requests, but be considerate and try not to take a site down. Most system administrators reserve the right to unceremoniously block you if you do endanger the stability of their site."

Possible solutions:

"There is no official limit on API requests, but administrators may block your client if it impacts the stability of a site."

→ See more examples in the Documentation Style Guide
→ Use a Style review checklist

Style exercise!

Rewrite this sentence to use active voice:

"This code can be shared with other users."

Possible solutions:

"You can share this code with other users"

OR

"Other users can find and use this code".

→ See more examples in the Documentation Style Guide
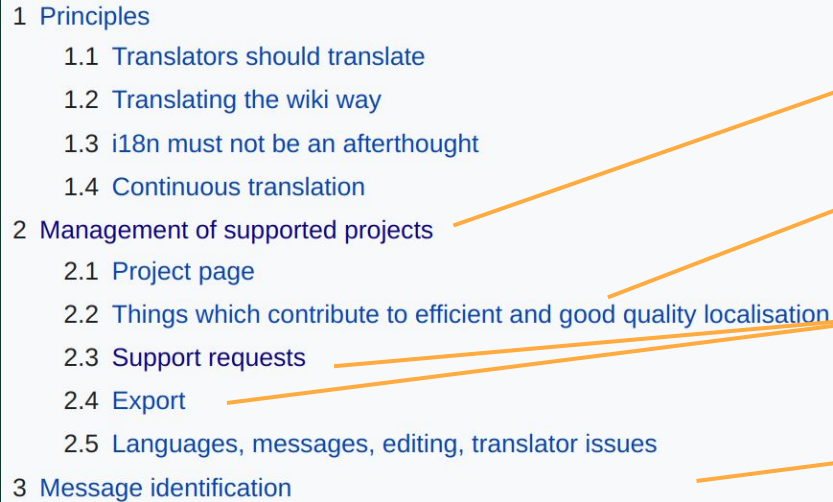
→ Use a Style review checklist

Structure saves readers' time

When readers can easily scan content of a doc, they can more quickly identify and skip over concepts they already understand or which aren't relevant for their current task.

- Headings **concisely** describe section content. In task-focused docs, headings represent the sequence of steps.
- Headings should use sentence case, and verb phrases where possible.
- Begin paragraphs and lists with the most important or representative concepts.
- Groups links to simplify navigation, but limit the number of links in a group.

Structure exercise!

Suggest improvements to this page structure and/or its headings:

Some solutions:

Use a verb and active voice:
Manage your project

Make more concise and use active voice:
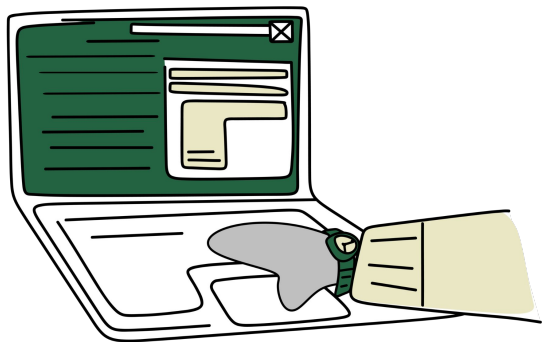Enable efficient, high-quality localisation

Use a verb and active voice:
Request support
Export translations

Split into more focused subsections?

→ See more guidance in the Documentation Style Guide

Consistency is kind

Consistent formatting and content structure improves user experience.

Reliable content can still appear untrustworthy if it has inconsistent style or formatting.

Consistent page structure facilitates skimming and enables users to build mental models that help them find info more efficiently.

Consistency exercise!

Dataset Name	Description
webrequest hive table - See also a separate list of Hive tables derived from webrequest	The webrequest stream contains data on all the hits to Wikimedia's servers. This includes requests for page HTML, images, CSS, and Javascript, as well as requests to the API.
pageview_actor hive table	The wmf.pageview_actor table is a smaller version of webrequest table with fewer columns.
pageview_hourly hive table	The wmf.pageview_hourly table contains 'pre-aggregated' webrequest data, filtered to keep only pageviews, and aggregated over a predefined set of dimensions.
projectview_hourly hive table	The `wmf.projectview_hourly` table is 'pre-aggregated' webrequest data at the project level. It is different from the `wmf.pageview_hourly` dataset in that it involves less dimensions and is therefore smaller in data size (and faster to query).
uniques devices	This dataset gives you how many distinct devices visit our projects
browser general	This dataset gives you pageview statistics broken down by user-agent related dimensions like OS family, OS major, browser family, browser major
mediawiki_api_request	The `mediawiki_api_request` table provides the log of api requests to MediaWiki
mobile apps session metrics	Contains aggregate stats about pageview sessions on the Android and iOS Wikipedia mobile apps
mobile apps uniques	Counts how many different Android and iOS Wikipedia mobile apps installs accessed Wikimedia sites during the given day or month
inter language	Traffic between different languages on the same project family
virtualpageview_hourly	Provides data about page previews on desktop Wikipedia

Accessibility is for everyone

Make link text descriptive. Use Special:MyLanguage in URLs on-wiki

example: https://www.mediawiki.org/wiki/Special:MyLanguage/Documentation/Toolkit

Check that the page is readable on mobile, with all important information visible.

Use descriptive section headings with h2, h3, and h4 styles, and no gaps in the nesting of the heading levels (no h2->h4).

Add meaningful alt text to all images.

Check if the page complies with the accessibility guide for developers. Use the WAVE tool (https://wave.webaim.org/) to check for critical issues.
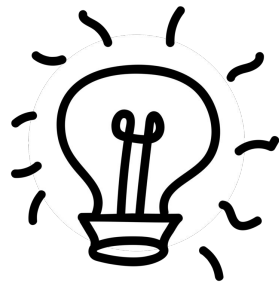
Questions?

References

- Documentation - MediaWiki
  - Visit the Toolkit for templates and checklists to help you get unblocked!

Images in this slide deck are CC BY-SA 3.0, Mam'Gobozi Design Factory (MDF), via Wikimedia Commons