Guide for Writing Developer Support Articles

This document is designed to provide a guide for writing developer support articles. When writing, try to stick to the key principles outlined in the **Google developer documentation style guide**. This will help ensure that there is a consistent format and tone of voice across all of the documentation.

> ⚠ This should be considered a "living document." The content in this document is subject to change at any time.

- Voice and tone
- Style guidelines
- Templates
- Additional resources

🗣️ Voice and tone

> ℹ When writing, follow the **Voice and tone** article in Google's developer documentation style guide.

Try to keep your writing **natural**, **approachable**, and—most importantly—**respectful**. Try to avoid writing in dry tones; be human and empathetic to your audience. Avoid jargon, buzzwords, and pop culture references. The purpose of your article is to **help** and **support** the developer looking for this information.

In short, to quote Google's style guide:

> *Try to sound like a knowledgeable friend who understands what the developer wants to do.*

Always assume that your audience is a **novice developer**. It is good practice to provide more information and explain complex concepts than assume the audience understands what are talking about.

You are writing for a **global audience**, so keep your writing as simple and clear as possible. Not all audiences are fluent English speakers and may require translation via Google translate. Clear writing will help translate the article more clearly into their native language.

✍️ Style guidelines

Use the following style guidelines when writing a developer support article:

- Article titles and headings
- Writing
- Formatting

Article titles and headings

Use sentence case

Always use **sentence case** for article titles and headers. There are some exceptions:

- Proper nouns (the names of specific people, places, organizations, or things). For example, "Video SDK" or "App Marketplace."
- Trademarks, such as macOS or Microsoft Office.
- Common abbreviations and acronyms, such as HTML, SSH, and API.
- When referring to a specific interface or API, such as "Get user's ZAK" or "Check a user's PM room."

Examples

✅

- Send a chatbot message
- Troubleshooting screen sharing
- Using the Zoom API with a free basic account
- Share an app without publishing it on the App Marketplace

❌

- How to send a Chatbot Message
- Troubleshooting Screen Sharing

Avoid punctuation in titles and headers

Don’t use punctation (like periods) or characters (such as @ or &) in your titles or headings, **except** for the following cases:

- Hyphens (“single sign-on”).
- Apostrophes (“Get a user’s ZAK”).
- Quotes and question marks in certain cases (see **FAQ and troubleshooting articles** below).
- If the name of a product, feature, setting, or other noun includes it as part of its name (“The Signing & Capabilities tab in Xcode”).

FAQ and troubleshooting articles

It’s okay to use quotation and question marks, particularly in FAQ or troubleshooting-type article subheadings. Quotes in particular will make it clear that the information contained is an exact string or code. For example:

- The "d:undefined symbols for architecture x86_64" error
- The "Unsupported Architecture. Your executable contains unsupported architecture '[x86_64, i386]" error
- When will the Video SDK be available?

Do not use formatting in titles or headers

Avoid the use of HTML code tags for article titles and headers. If you can, try to avoid referring to anything that would use otherwise use some kind of formatting (such as an API parameter).

Avoid questions in article titles

Instead, try to frame the article title as a statement.

Examples

✓

- Zoom Video SDK release
- Updating a published app with current users

✗

- When will the Video SDK be available
- What happens to my current users when I am updating my published app

Avoid using How To in guide articles

Instead, begin the guide article with the verb, like “Get,” “Display,” and “Return.”

Example

✓

- Limit the users who can install my Zoom Marketplace app

✗

- How to limit who can install my Marketplace App

Writing

Use second person tense

Always use second person (“you”) tense when writing your articles. Avoid the use of “we.”

The exception to this rule is when speaking on Zoom’s behalf (an “editorial we”), such as soliciting audience feedback. For example, “we would love to hear your feedback in the developer forums.”

Examples

✓

- This short guide covers the following:
- First, confirm the location of the references to the SDK files in Xcode.
- We are always working to improve the Client Web SDK efficiency and performance.
- We are working to support higher resolutions, starting with 720p in a future release.

✗
- In this short guide, we will cover:
- First, we will need to confirm the location of the references to the SDK files in Xcode.

Use serial commas

Serial commas (or “Oxford commas) help avoid any potential confusion or uncertainty. Use them in a series of three or more items after “and” or “or.”

Example

✓ The user’s name, email, and phone number.

✗ The user’s name, email and phone number.

Spell out the first acronym or abbreviation

When referring to less commonly-known acronyms or abbreviations, always write it out the first instance of its use, followed by its abbreviation or acronym in parenthesis (“JSON Web Token (JWT)”). Never assume that your audience will know what the acronym or abbreviation means.

It’s also good practice to link to articles that help further clarify any possible uncertainty. For example, linking out to a Wikipedia article about JSON Web Tokens.

Article titles

If you can, try to avoid using acronyms and abbreviations in headers or article titles. If you do use an acronym or abbreviation, be sure to write it out in full in the article body (“JSON Web Token (JWT)”) for user context.

Common acronyms or abbreviations

It’s okay to avoid spelling out common acronyms and abbreviations (for example, API, HTML, URL, HTTP, SSH). But, when in doubt, define the acronym or abbreviation. Remember: too much information is better than not enough information.

Example

✓ The `to_jid` value can be the Jabber ID (JID) of a Zoom Chat channel or user.

✗ The `to_jid` can be the JID of a Zoom Chat Channel, or Zoom Chat User.

Avoid the use of please

While it may seem polite, it is best to tell your audience what to do when being instructional. This conveys a sense of confidence in your writing.

Examples

✓
- Delete the root folder and download it again.
- If you still have issues, reach out to Zoom on the developer forum or submit a ticket to Zoom technical support.

✗
- Please delete the root folder and re-download it.
- If you are still facing issues, please reach out to Zoom on the developer forum or submit a ticket below.

Use active voice

Active voice helps make it clear to distinguish who is supposed to perform an action or what is being acted upon. Try to use active voice when possible for clarity.

Example

✓ You can use some of the Zoom API endpoints for free.

✗ Some of the Zoom API endpoints can be used for free.

For more information about active and passive voice, read Google's **Active voice** guide.

Use articles when referring to nouns

Include *a*, *an*, and *the* in your writing. When referring to a noun (such as a parameter), always include what it is for clarity (for example, “the `user Id` parameter,” “the user’s JID value,” or “a 404 HTTP status response”).

✓

- Once you have a chatbot token and **a** `to_jid` **value**…
- In order to send a chatbot message, you need to provide **the** `to_jid` **value**.

✗

- Once you have a Chatbot Token, and a `to_jid`…
- In order to Send a Chatbot Message, you need to supply the `to_jid`.

For more information, read Google’s **Articles (a, an, the)** guide.

Formatting

Avoid large blocks of text

Instead, utilize vertical space and white space. This makes your writing much easier to read and will help keep your audience engaged.

Keep articles specific and concise

Try to keep your articles specific to a concept. Writing an article about a very broad concept makes it a long read and makes it difficult for your audience to easily find what they are looking for. It also makes it more difficult to maintain.

Example

Instead of writing this general article:

> *Add webhooks to an app*

…break it into separate, specific topics related to webhooks:

> *Add webhooks to an OAuth app*
>
> *Add webhooks to a chatbot app*
>
> *Add webhooks to a JSON Web Token app*
>
> *Add webhooks to a Webhook Only app*

Images

If possible, try to avoid using a lot of images in your articles. This helps reduce the maintenance burden in the case where there is an interface update.

When you use images in your article, make certain that you:

- Explain any steps related to the image in your article. Don’t rely entirely on the image to tell the story.
- Include descriptive **alt text** for the image. The alt text replaces the image if an image breaks. It also provides information to users who rely on screen readers, users with a low-bandwidth internet connection, or users that rely on text-only web browsers.
- Try to use a high-quality image that is clear and easy to read when reduced in size (such as a thumbnail). Try to use `.svg` or `.png` format if possible.

For more information, read Google’s **Figures and other images** documentation.

Code

When referring to code, parameter and returns, command line commands, user input, and other code-related information, use monospace (`<code` HTML tags) format. This helps the audience easily identify code and command input in a sentence.

For blocks of code, such as an API response, make certain to enclose the data in a code block.

For more information, read Google's **Code in text** documentation.

Examples

✓

- Another important event to listen for is the `IAuthServiceEvent` event.
- You can check your version with the `node -v` command.

✗

- Another important event to listen for is the *IAuthServiceEvent*.
- You can check your version with *node -v.*

📄 Templates

Use the following templates to get started writing a developer support article:

- Guide Template
- Frequently Asked Questions Template
- Troubleshooting Template

💡 Additional resources

The following are useful resources to help you write your developer support articles:

- Google developer documentation style guide
- Hemingway Editor — This is a very useful Web-based tool for checking sentence difficulty, readability score (aim for grade 8 readability), passive voice, and finding simpler phrase alternatives.