

Twilio Documentation Writers' Guide

Updated January 11, 2021

Writing technical content which is not only informative and authoritative but also informal and engaging is not easy. Engineering staff shouldn't expect to get it right first time, especially if English is not their first language. The [Developer Education](#) team is always ready to help you out, whether that's with mentoring, preparing draft copy for publication, or writing content from scratch — please reach out to us on Slack [#Help-Docs](#). But if you prefer to prepare a draft yourself, these guidelines to help you produce better material.

Be concise

We want readers to enjoy using Twilio documentation, but we know they're very busy. They come to the documentation with a question. Our goal is to provide an answer as quickly and with as little interruption to their workflow as possible.

Keep your sentences short, to the point, and easy to read. Don't use ten words when three will do:

YES "Now validate the configuration".

NO "Now initiate the validation phase of the configuration".

Assemble sentences into short paragraphs, each of which focuses on a single, specific point. Don't repeat yourself. Avoid multiple sub-clauses.

Be code wise

Developers love code. They will zero in on it and may not even look at your text. If that solves their problem, great — that's the primary role of Twilio's documentation. Well presented, clearly understandable code can be worth a thousand words of explanation. Your examples should be thoroughly commented and use identifiers that make their meaning explicit. Clarity of function is more important than elegance.

Remember: authors express, readers refactor.

Be clear

Establish your subject right at the start with an expressive page title.

Read your work out loud. Imagine you're talking to a customer. Would they understand what you're saying, or would they be unsure whether you mean one thing or another? Where did you pause for breath? Did you say the same thing twice at different points? Did the idea presented in a paragraph follow logically on from the previous paragraph? Asking yourself questions like these can help you prepare copy that will be clear to all readers.

Be active

Always use the "active voice", never the "passive voice". This will make your text feel more vigorous and convey that it is about taking action now, not at some vague point in the future. For example, for a headline:

YES "Make a video call".

NO "Making video calls".

Or this description of a Twilio service:

YES "Twilio Chat uses a REST API"

NO "Twilio Chat is using a REST API"

Be organized

Plan the document's structure. Organize it with headings to help readers quickly find the section or sections that solve their problem. Many visitors will jump straight to the article's sample code — that's fine, but don't ignore those developers who do want to read more to gain a better understanding of how a product works. Establish your subject right at the start. This is why a clear page title is crucial.

Place additional guidance that only a few developers may need at the end so that it doesn't get in the way of other readers, but make sure you indicate early on that it is there — use a link for this. If it's a very special case, maybe it needs to be in a separate, linked document. Two or three mid-sized and well-linked documents are better than one long one that covers all the edge cases but is hard to navigate.

Be consistent

Keep the [Style Guide](#) handy. No one expects you to read it through, but refer to it while you write. It covers the common questions that writers of Twilio documentation have about how documents should be formatted, and how key words, phrases and terms are used, all with examples. It embodies the rules that the Developer Education team apply when writing and editing documentation.

Be informal

Two keys to accessibility are informality and friendliness, but take care not to be too enthusiastic. No one wants to read dry, lifeless content, but not all developers are keen to be addressed as if they were your best friend. They're here for information, not a chat.

Write conversationally — address the reader directly, as an individual, not as a distant spectator. But don't become verbose — keep to the point.

Be audience aware

Your audience is developers — people who write code or build hardware to solve problems. Many will be skilled in certain areas, very few in all areas. Even the most experienced developer will be new to some task. Don't assume specialist knowledge that the reader may lack. Don't assume that something is "simple" or "easy" for customers just because it is simple or easy for you.

Equally, though, don't assume they are a beginner — unless that's the audience at which your article is aimed. It is reasonable to assume some familiarity with the command line: what a prompt is and how directory structures are navigated. They will know how to use a GUI, though they may not know where a little-used feature is to be found. Help them out by showing them.

Readers can arrive from anywhere — often from search engines — and move on to any place. Don't rely on readers having read, or even seen, other Twilio docs. They may not go to the next page in a sequence. Keep pages self-contained, and focus on the topic being covered. Do link to related content you believe that the reader may find useful. Linking is a great way to expose visitors who have come for one piece of information to other content that they may find useful.

Be global

Not all cultural icons are shared. You may love a certain TV show, movie series, band or musical genre, but your readers may not. They may not even have heard of them! Don't alienate readers by anticipating an enthusiasm that they may not possess.

Developers come from all cultures of the world possible with views and lifestyles very different from your own. But these qualities are not relevant to their abilities as a developer so remember that they are irrelevant to the content too.