

Twilio Error Codes Style and Writing Guide

This guide details Twilio's rules for writing and updating error code documentation. They show you how to document an error code, and they are the criteria by which new and existing error code docs are assessed for completeness.

These criteria focus solely on how the error code should be described if the documentation is to be useful. They are not intended to help you decide how to organize your product's errors, for example whether to have a single, generic API request failed error, or multiple errors for every possible reason why the API request failed. However many errors you decide to implement, this guide will help you document each of them.

General guidance

Error code pages each focus on a specific error code. They provide at-a-glance information to help developers quickly understand the most likely reason why they have encountered the error, and how to fix the problem.

Additionally, each page provides a more detailed description of the error for those developers who require more in-depth assistance or would like a deeper understanding of why the error occurred, or for those developers who have not been assisted by the at-a-glance information.

We always aim to write clearly and inclusively to ensure that developers, wherever they are in the world and whatever their level of experience, find the solution that is right for them as quickly and as clearly described as possible. Always refer to the [Docs Style Guide](#) for usage and style information.

Be mindful that the content is aimed at developers not at Twilions, so don't unthinkingly assume they have access to the same privileged information that you may have.

Please feel free to reach out to Developer Education if you would like assistance or mentoring when you begin to write Error Code content.

Monkey

Twilio's [Monkey](#) provides a UI for creating and editing error code pages. You complete a series of fields, and Monkey converts their contents into [Markdown-formatted](#) text which is passed to the docs engine to be rendered as styled text.

Most of the fields take a single, short phrase, aimed at developers who just want an at-a-glance view. However, the **Description** field gives you scope to provide detail and explanation to developers who would like more information or need a little more help. These fields are discussed below with examples of good and bad style.

Note You must be signed in to Twilio via Okta and have a VPN active in order to access Monkey.

Error code documentation workflow

We recommend that you write your entry in a Google Doc, to allow you to spell-check your text and to seek review from colleagues and Twilio's Developer Education team. Do not format your doc — just use the formatting markup that Monkey expects. [This formatting markup is described later](#).

Monkey text fields

Message

This is short and to-the-point statement (10-15 words) of the error that the code represents: a brief, human-readable error message string. Make sure every word is relevant. Use your word count fully.

It will specify an event which has occurred, for example a call was blocked, a requested resource could not be found, an API access attempt failed to authenticate, or an API request to send a text message did not include the phone number to send it to.

Always be specific:

NO “A status message could not be sent”

YES “A status message could not be sent to the specified callback URL”

Don’t leave the reader asking a question, especially if there is room to include relevant details: in this example, what request and why couldn't it be processed?

NO “The request could not be processed”

NO “The request to send a message could not be processed”

YES “The request to send a message lacked a ‘to’ phone number”

Avoid stating that an operation “failed” — of course it did, that’s why an error was issued. Instead indicate **why** it failed, even if you can only give a generic reason (i.e., the error code covers a number of possible issues):

NO “The API request failed”

YES “The API request could not be authenticated”

Secondary

See [Description](#).

Log Type

select the most appropriate item from the list. Usually you will select **Application**.

- **Application Error** is an issue that should be fixed by the user.
- **Carrier Error** is a regular carrier error.
- **Twilio Error** is an issue that should be fixed by Twilio.

Product

Select the Twilio product to which the error relates. Only leave this field blank if the error code applies to multiple products.

Description

The **Description** combines the [Message](#), [Possible Causes](#), and [Possible Solutions](#) at-a-glance information into a single, detailed piece of content. Your goal is not only to indicate the meaning of the error, what caused it, and how the reader can stop it happening again, but also to provide the information the developer requires to understand completely why an action failed and how the solution will prevent failure.

Do think about what actually triggers the error in the backend, and what states a user must be in order to receive the error. Is there any other mistake that could also trigger this error? For example, the TwiML “invalid number” error can be caused by attempting to dial an invalid number (as you might expect) or because no number was specified. These two causes would be listed under [Possible Causes](#), but

you should also use the **Description** to suggest that, in the case of the second cause, the number was omitted because an incorrect TwiML noun was used, such as `<Client>` instead of `<Number>`.

Do explain your terminology and spell out abbreviations. Don't assume the reader has the same knowledge and understanding that you do.

Do include links to documentation which may be helpful to the reader. Include such links inline with the content, not as a "Further reading" or "See also" section at the end. Please reach out to your BU's specialist in Developer Education if you need help selecting material to link to.

Do not include sections headed "Possible Causes" and/or "Possible Solutions" sections on your Description. Monkey adds these for you. If these sections are included already, remove them.

Do not include the [Message](#) text at the top of the **Description**.

Do include examples if you can and it will clarify your content. Include examples as code panels — the [Formatting section](#) below will show you how to do this.

WARNING Monkey currently uses both the **Secondary** and **Description** fields to generate the description text that will appear on the page. Cut and paste the **first sentence** of your **Description** into the **Secondary** field. Make sure the **Secondary** field ends with a period.

Possible Causes

This is a single sentence of 20-25 words, or a list of 1-3 items of up to 15 words each, indicating the most likely reason(s) for the error. Include real, common causes — edge cases should be included only in the [Description](#). Make every word count — include all relevant information. Do not restate the [Message](#).

NO "The page could not be found"

NO "Flex encountered a service error when connecting to an agent"

NO "Flex encountered a service error"

YES "The page is no longer available, or the URL of your HTTP request to Twilio was incorrect".

Make the sentence active, not passive, and address the reader directly:

NO "Attempting to use a premium feature with a trial account."

YES "You attempted to use a premium feature with your trial account."

Note If there are many possible causes, this may be an indication that the single error code is insufficient and so you should consider adding new error codes.

For lists, start each item on a new line and preface each one with an asterisk and a space:

- * Cause 1
- * Cause 2
- * Cause 3

You can include [Markdown-formatted links](#) in your text. You must ensure the links are correct — this is not checked for you.

Keep causes brief — just state them plainly. If you need to provide more detail, do so in the [Description](#) section. For example:

Possible Causes

Your request could not be authenticated because you did not include credentials or your credentials were wrong.

Description

Your request could not be authenticated. This will be the case if you provided incorrect credentials or no credentials at all — you must submit your API key. You may have also specified an unsupported authentication method — Twilio uses Basic authentication. Make sure your requests do include an Authentication: Basic header.

If the error clearly indicates the cause — because it is specific — you do not need to include a **Possible Causes** entry.

Possible Solutions

This is a single sentence of 20-25 words, or a list of 1-3 items of up to 15 words each, indicating the most common solutions to the error. Less commonplace solutions should be included only in the [Description](#). Make your solution(s) match your [Possible Causes](#). Make every word count — include all relevant information. Do not repeat the causes.

NO “Check your credentials.”

NO “Check your request.”

NO “No further information available.”

YES “Check that the credentials used in your HTTP request are correct”

Solutions should not be the exact opposite of the cause. For example, if the cause is “You supplied invalid authorization credentials with your request”, the solution should not be “Supply valid authorization credentials with your request” but, say, “Make sure that the variables your code is using to add authorization credentials to your requests contain the correct values”.

For lists, start each item on a new line and preface each one with an asterisk and a space:

- * Solution 1
- * Solution 2
- * Solution 3

You can include [Markdown-formatted links](#) in your text. You must ensure the links are correct — this is not checked for you.

If you need to include code examples, place them within the [Description](#) section.

Keep solutions brief — just state them plainly. If you need to provide a lot more detail, do so in the [Description](#) section. For example:

Possible Solutions

Check your request uses Basic authentication and includes an API key.

Description

Check the authentication header that you include with your request to make sure that it specifies Basic authentication and includes your API key as the authorisation token.

Updates to existing pages

You may add [Possible Causes](#) (and equivalent [Possible Solutions](#)) to an existing error code page. Make sure you expand the [Description](#) accordingly. But avoid overloading the page: this is a guide to an error and common causes/solutions. If the cause/solution you are adding is an extreme edge case, add it solely to the [Description](#).

Formatting

Monkey's text fields can be formatted using [the Markdown markup language](#). The **Markdown** field shows how all the data fields above are converted into a block of Markdown-formatted text, and a rendered version is presented below it so you can see how the content will appear on the page.

Formatting should be kept to a minimum. Use only the following formatting options, for clarity and consistency.

To emphasize words or phrases, and to indicate UI elements, e.g., in Console, used **bold formatting**: wrap the word or phrase in pairs of asterisks. For example, `**Save**` becomes **Save**.

Use code formatting to indicate URLs, paths, directories, function names, variable names, and such. Wrap the element in backticks. For example, ``myFunction()`` becomes `myFunction()`.

To include a link, wrap the link text in square brackets and add the link URL in round brackets immediately after (with no space between them). For example:

```
[contact Twilio Help for details](mailto:help@twilio.com)
```

```
[Visit the Twilio docs to learn more](https://twilio.com/docs/voice)
```

For the best SEO, use full sentences or phrases as your link text, not single words. Use sentences that express an action, and don't split them up. For example:

NO Go to [Twilio docs](https://twilio.com/docs/voice)

YES [Visit the Twilio docs to learn more](https://twilio.com/docs/voice)

NO You can [contact](mailto:help@twilio.com) Twilio Help for details

YES You can [contact Twilio Help for details](mailto:help@twilio.com)

Code blocks used to present example code and data are formatted by placing the code or data between three backticks ````` on separate lines. For example:

```
```
int myVariable = 42;
```
```

Indicate the type of code or data by stating it immediately after the first `````. For example:

```
```c
int myVariable = 42;
```
```

```
```shell
twilio api:microvisor:v1:list
```
```

```
```json
{
 "instruction": "reject",
 "activity_sid": "WA0123456789abcdef0123456789abcdef"
}
```
```

Note Twilio code style mandates two spaces for each indent level.

Full error code examples

<https://www.twilio.com/docs/api/errors/10002>