

Create and edit articles

September 15, 2026

Create and edit articles

An article is the content you write or revise in Conduit. Use the editor to create a new article or to change an existing article. A slug identifies an existing article in its route.

If you need to...	Go to
Create an article	Enter and publish an article
Change an existing article	Handle create and edit variations
Find an article before opening it	Find articles
Read or manage the article after publishing	Read and manage an article
Review article terminology	Basic terminology

Before you start

Have the article information ready before you open the editor.

Prerequisites

What to have ready	Why it matters
The choice between a new article and an existing article	It determines which editor route you open.
The article slug when changing an existing article	It identifies the article to load.
The title, description, body text, and tags	They are the values you enter in the editor.

Open the editor

The editor is the form where you enter or revise an article.

Steps

1. Choose the route that matches your task:

If you need to...	Open
Create a new article	The editor route without a slug
Edit an existing article	The editor route with the article slug

2. Open the selected route.

Result: The editor opens. A new article starts with empty article fields; an existing article loads the article identified by its slug.

Understand the editor

The editor contains four article inputs, a tag area, and one publishing control.

What you enter or use	Control or area	What it does
Article title	Article Title	Accepts the article title.
Short description	What's this article about?	Accepts the article description.
Article content	Write your article (in markdown)	Accepts the article body in markdown.
Tags	Enter tags	Accepts a tag before you add it to the tag list.
Existing tags	Tag text and its close icon	Displays the tags already added and lets you remove one.
Submit the article	Publish Article	Sends a new article or an edit.

The editor does not show separate field labels or required markers. It uses the text inside the inputs to identify them. No separate save or update control is used.

Enter and publish an article

Use this procedure for both a new article and an existing article. The route you opened determines whether **Publish Article** creates an article or updates the article identified by the route slug.

Prerequisites

- The editor is open.

- You have the title, description, body, and tags for the article.

Steps

1. In **Article Title**, enter the article title.
2. In **What's this article about?**, enter the article description.
3. In **Write your article (in markdown)**, enter the article body.
4. In **Enter tags**, enter a tag.
5. Press Enter to add the tag to the tag list.
6. Repeat the tag entry and Enter action for each tag you want to add.
7. For a tag you want to remove, select its close icon.
8. Select **Publish Article**.

Result: The article submission is made. A new article uses the create path; an existing article uses the update path.

Handle create and edit variations

The editor uses the same fields and **Publish Article** control for both tasks; the route decides which article operation is submitted.

Prerequisites

- The editor is open.

Steps

1. Choose the submission path that matches the article:

If you opened...	Publish Article submits...
The editor route without a slug	A new article.
The editor route with the article slug	An update for the article identified by that slug.

2. Select **Publish Article**.

Result: The editor submits the create or update operation selected by the route.

When submission goes wrong

Returned validation or API errors appear in the editor's error list. The list shows the returned field name and value.

The submission error message is:

No fixed message is declared. The shared error list renders each returned error as `<key> <value> .`

What you see	What it means	What to do
A returned field name and value in the error list	The submission has a validation or API problem associated with that field.	Correct the information named by the returned entry, then select Publish Article again.

No fixed success-confirmation message appears after submission. Use the resulting article route as the next place to continue.

After publishing

After submission, the application redirects to the article route for the submitted article.

Open [Read and manage an article](#) to continue with that article.

Use [Find articles](#) when you need to locate an article from the article lists.

For the wider article workflow, open [Articles overview](#).

Result: The article route is the next destination after submission.