

Guidelines for the Developer DAO Blog

This document defines a quality assurance method for articles published on the D_D Blog and rewarded with bounties. Bounties are open for everyone that wants to write an article but require a D_D member to vouch for non-members.

Guidelines

General Rules

- The goal of the D_D blog is to educate developers, give educated opinions on popular development topics, and share the experiences of the D_D members
- **Cover topics that can't be found elsewhere or give more context/details than existing articles** (i.e., write a significantly better article)
- The articles should have at least 1000 words
- Define the target audience and other prerequisites right at the beginning
- Focus on problems and solutions related to the article's title
 - Avoid history or context about companies or their founders, or the ecosystem if it's unrelated to the topic
 - If you need more words, add details instead of straying into related topics.
 - Explain specific products only if they directly relate to the topic
 - Exception: D_D partners may include a paragraph before the summary/conclusion that promotes their product
- Use solutions from our preferred partners when possible
- Add reasons for using a product/technology

Writing Style

- Use third person plural (us/we) for tutorials and guides to emphasize the feeling of community
- Use the first person singular (I/me) for opinions or experience reports to ensure the reader knows this is a personal opinion of yourself and not of D_D in general.
- Use the third person plural (they/them) to refer to the third person if the gender is unknown.
- Avoid passive voice
- Avoid jargon.
 - If you must introduce a new word or phrase, ensure you have a good reason and explanation.
- Follow the editor's spelling.
 - Exceptions are obvious spelling or grammar mistakes
 - Personal opinions don't get exceptions

Title

- Keep the title as short as possible, but not too short.
- If you know search engine keywords related to your topic, include them
- Phrase a title similar to a search phrase you would use to find it
 - Questions are good

Intro Paragraph

The first paragraph of an article should follow the lede structure:

- Start with a problem or a reason (1-2 sentences)
 - why would the reader want to read the article?
- Tease a solution or explanation (1-2 sentences)
 - what solution does exist and will be explained in the article
 - don't go into detail about how the solution might fix their problem yet
- Lead to the central part of the article (1-2 sentences)

Example:

(reason) Onboarding new users to DApps is hard because they need to understand how crypto wallets work, get their first tokens and then sign transactions. (solution) Account abstraction solves this issue by allowing people to use DApps without a wallet and sponsored transactions. (transition) This article will explain how you can add account abstraction to your DApp using Biconomy's SDK while giving details on common problems with this integration.

Headings

Start with H2 for the top-level subheadings and go deeper if required.

Focus on the problem at hand and avoid wandering off. Don't give people a history lesson if it doesn't help the topic.

After the intro paragraph, the first section/subheading should answer questions in the title or explain the proposed solutions in more detail.

Note: Search engines will include the answer as snippets if it is short enough. Provide a comprehensive definition of a term in just 2-3 sentences. The initial sentence should establish the keyword or topic, while the subsequent sentences should highlight 2-3 essential facts.

In a tutorial or guide, split the topic into steps (i.e., Target Audience, Required Prerequisites, Setup, Implementation, Deployment) and logical sub-steps (i.e., Installing the Framework, Implementing the Frontend, Deploying Contract A, Deploying Contract B)

In an explanation, split the topic into logical sections (i.e., What is Account Abstraction, How does Ethereum Achieve Account Abstraction.) The same goes for opinion pieces. If you write an experience report, a simple chronological order is okay.

General to Specific and Beginner to Professional

Begin each section promptly by responding to the heading question, using no more than two sentences. Follow up with specific details and examples. This applies to every paragraph, with the first sentence as a summary or highlight of the paragraph's essential information, followed by further elaboration.

When writing your content, consider the audience seeking beginner-level information. Start with the basics and gradually increase the complexity of the information as you progress through the article without assuming any prior knowledge.

Note: *A beginner doesn't mean "programming beginner" but someone who might just have heard about the topic you're writing about; your article should target intermediate and expert-level readers.*

Focus on Facts

Always support your claims with trustworthy data. Link directly to primary sources when possible.

Graphics, Tables, and Examples

Add screenshots or screen recordings for complex UI interactions and their results.

If the results are text, copy the text before resorting to screenshots.

If possible, add diagrams to visualize abstract concepts.

Number your images, starting with Figure 1, and try to refer to them by "Figure <Number>" in the text that comes before the image. Use captions and alt text (under 125 characters) for your images; they can be the same.

Use tables to simplify comparisons.

Add examples that illustrate concepts. This includes code examples but also prose. Keep them concise (i.e., don't add 100 lines of UI code where a UI isn't required to explain a concept)

Keywords

If D_D requests the article, we provide you with SEO keywords.

The target keyword has to be in the first paragraph and the first subheading

The text should include at least 1 keyword per 200 words of text

If there is only one keyword provided, synonym use is permitted

Linking

Use informative anchor texts.

Good: [You can read more about signatures in the docs.](#)

Bad: You can read more about this [here](#).

Open outbound links in a new tab. (i.e., use the target="_blank" HTML attribute)

Link to other articles on the D_D blog when relevant.

Try to avoid outbound links at the beginning of the article.

Avoid linking to Wikipedia.

Prefer linking to content from the last 12 months; avoid content older than 3 years.

Link to primary sources if possible. If not, ensure the article you link to is high-quality; otherwise, omit the link.

Product Advertisement

If the article uses our partner's technology, include a brief section promoting them under an H2 heading at the end of the article before the summary.