

Extensions

Extensions allow you to use simple Markdown syntax to achieve complex rendering. The following are the available extensions for Besu documentation.

Important: Extensions are only available for the Markdown files under the `docs` directory.

- [Abbreviations](#)
- [Call-outs](#)
 - [Collapsible call-outs](#)
- [Checklists](#)
- [Code samples](#)
 - [Tabbed code blocks](#)
 - [Line numbers](#)
 - [Code syntax highlight](#)
- [Definitions](#)
- [Emojis](#)
- [Footnotes](#)
- [Global content](#)
- [Highlight](#)
- [Key presses](#)
- [Magic links](#)
- [Math](#)
- [Plant UML diagrams](#)
- [Strikethrough](#)
- [Symbols](#)
- [TOC](#)
- [Variables](#)

Abbreviations

We recommend avoiding the use of abbreviations, but some such as “PoW” for “proof of work” or “dapp” for “decentralized application” have become part of the Ethereum jargon. The [Abbreviations](#) extension enables defining abbreviations, so that the full form displays when you hover over the abbreviation.

Define abbreviations at the end of the Markdown file.

Example:

```
---
description: This is an example page
---

# My example This page explains PoA networks.

... more text ...

*[PoA]: proof of authority
```

Call-outs

The [Admonition](#) extension enables call-out blocks. These are used frequently in the ConsenSys documentation to include side content or highlight important content.

Begin each call-out with three exclamation points and the [call-out type](#). You can [change the call-out title](#) by adding it in quotes.

Example:

```
!!! note

This is a multi-line note in the ConsenSys documentation.
You can replace `note` with a different call-out type.
The default note title is `Note`.
Each line in the call-out content must be indented by four spaces to be included in the call-out.
```

Supported types include:

- **Note:** Used to add information about a subject that doesn't always need to be taken into account. For example, “When running on localhost, include 127.0.0.1 as a listed IP address.”
- **Important:** Used to add important information about a subject that should always be taken into account. For example, “Allow Docker up to 6G of memory.”
- **Example:** Used to display an example. This is often used with a code sample.
- **Info:** Used to add non-essential detail about a subject. For example, “The miner coinbase account is one of the accounts defined in the genesis file.”
- **Tip:** Used to add information that could help improve the use of a tool. For example, “To restart the private network in the future, start from step 4.”
- **Caution:** Used to warn the users to proceed with caution. For example, “Configuring alternative elliptic curves is an experimental feature.”

- **Warning:** Used to warn the users about something important. For example, "This will be deprecated in the next version."
- **Critical:** Used to alert the user about a potential dangerous effect such as a risk of destroying something or losing assets. For example, "Never use the development private keys for production use."

Collapsible call-outs

The [Details](#) extension enables collapsible call-outs by replacing the exclamation points with question marks. The call-out can be open or collapsed by default. This helps reduce the content length and enables a faster overview of the whole page.

Example:

```
???+ note "Collapsible note"

    This is a collapsible note.
    The plus sign makes it open by default.
    Remove the plus sign and it will be collapsed by default.
```

Checklists

The [Tasklist](#) extension enables displaying an unordered list as a checklist.

Example:

```
- [ ] This list item has an unchecked checkbox.
- [x] This list item has a checked checkbox.
```

Code samples

The [SuperFences](#) extension enables using code blocks to present code samples in the documentation. A basic code block uses triple back ticks (```) and the language name to enable [syntax highlighting](#). Always surround code blocks with blank lines.

Example:

```
```json
{
 "jsonrpc": "2.0",
 "id": 1,
 "result": true
}
```
```

Tabbed code blocks

[SuperFences](#) also enables tabbed code blocks. The following example groups usage syntax and an example in different tabs in the same block.

Example:

```
=== "Syntax"

    ```bash
 ethsigner --chain-id=<chainId>
    ```

=== "Example"

    ```bash
 ethsigner --chain-id=1337
    ```
```

Line numbers

[SuperFences](#) also enables adding [line numbers](#) to long code samples, which makes it easier when discussing the sample. The line numbers only appear in a code block that uses the `linenums="1"` parameter.

Example:

```
```javascript linenums="1"
const Web3 = require("web3");
const Web3Quorum = require("web3js-quorum");
const web3 = new Web3Quorum(new Web3("http://localhost:22000"));
web3.priv.generateAndSendRawTransaction(options);
```
```

Code syntax highlight

The [Highlight](#) extension enables automatic syntax highlighting of code blocks. Define the code language after the code block delimiter to ensure correct highlighting. If you don't provide the language name, the extension attempts to automatically discover it, but this can lead to errors.

Example:

```
```json
{
 "jsonrpc" : "2.0",
 "id" : 51,
 "result" : {
 "startingBlock" : "0x5a0",
 "currentBlock" : "0xad9",
 "highestBlock" : "0xad9"
 }
}
```
```

Refer to the [full list of support languages](#).

Definitions

The [Definition List](#) extension enables listing definitions. This may be useful for creating glossaries.

Example:

```
Consensus layer
: The layer of the Ethereum network that enforces network rules, and generates and verifies blocks.
  Previously known as "Ethereum 2.0" or "Eth2."

Execution layer
: The layer of the Ethereum network that contains and executes smart contracts.
  Previously known as "Ethereum 1.0" or "Eth1."
```

Emojis

You can use emojis in the Markdown. Emojis are fun, but they can also be useful to draw the user's attention. Use only neutral emojis such as `:warning:`, which displays [blocked URL](#).

Refer to the [full list of available emojis](#).

Footnotes

The [Footnotes](#) extension enables adding footnotes at the end of a page.

Example:

```
---
description: This is an example page
---

# My example

You can add a footnote[^1] to provide additional information about a subject.
The footnote number displays as a clickable superscript that takes the user to the footnote.

... more text ...

[^1]: The footnote goes here.
  A clickable arrow displays that takes the user back to the original text.
  When writing multi-line footnotes, indent each line with four spaces.
```

Global content

If you have content to be repeated on multiple pages, you can create it in a common page in the `/docs/global` directory and include it in all required pages.

For example, use `--8<-- "global/test_account.md"` to include the content of the `test_accounts.md` page in another page.

Info

The [exclude plugin](#) excludes pages in the `/docs/global` directory (configurable in `mkdocs.yml`) from the final rendered site, since otherwise, every Markdown file is rendered and copied. It's useful to prevent global files to be reachable as standalone pages, since they're intended to be included in other pages.

Excluded pages are still in the source repository, but they aren't copied in the final site and don't appear in the search results.

Highlight

The [Mark](#) extension enables highlighting of text. Text surrounded by double equal signs is highlighted in yellow.

Example:

```
==This is highlighted text==
```

Key presses

The [Keys](#) extension enables styling key presses by surrounding them with double plus signs.

Example:

```
++ctrl+alt+delete++
```

Magic links

The [MagicLink](#) extension automatically displays a URL as a link. You don't need to surround the URL with Markdown link syntax.

Math

The [Arithmatex](#) extension enables rendered math formulas in the documentation using [MathJax](#) syntax.

Example:

```

$$\prod_{k=1}^t \sigma_{i_k}^{L_{i_k}(0)}$$
  
Constructing the threshold signature  $\sigma$  from  $t$  individual  
signatures  $\sigma_{i_k}$ ,  $k=1, \dots, t$  and the Lagrange polynomials  
 $L_{i_1}, \dots, L_{i_t}$  associated to the set  $I=\{i_1, \dots, i_t\}$  of signers.
```

Plant UML diagrams

You can add diagrams using [Plant UML](#) syntax.

Example:

```
```\plantuml format="svg" alt="Plantum diagram example" title="My super diagram"  
Actor1 -> Actor2: calls
Actor1 <-- Actor2: responds
```\n
```

Strikethrough

The [Tilde](#) extension enables displaying text strikethrough by surrounding the text with double tildes (~).

Example:

```
~~This text is struck through~~
```

Symbols

The [SmartSymbols](#) extension enables symbols. For example, `-->` displays as \rightarrow .

TOC

The [Table of Contents](#) extension automatically displays a table of contents of the current page on the right side of the page. It displays titles to the third level (titles prefixed with `###`).

This extension also displays a permalink on the right of any title. This permalink can be used to point directly to the title from another website.

Variables

The [Extra Variables](#) plugin enables using values from the `extra` section in `mkdocs.yml`, or if using the new system, from `mkdocs.extra.yml`, as variables in the Markdown code.

For example, if you have the following in `mkdocs.yml` or `mkdocs.extra.yml`:

```
extra:
  support:
    email: quorum@consensys.net
```

You can display the email in any documentation page using:

```
{{support.email}}
```