Version: 2.4.0

On this page

Docs Introduction

The docs feature provides users with a way to organize Markdown files in a hierarchical format.

ⓘ INFO

Check the Docs Plugin API Reference documentation for an exhaustive list of options.

Your site's documentation is organized by four levels, from lowest to highest:

1. Individual pages.
2. Sidebars.
3. Versions.
4. Plugin instances.

The guide will introduce them in that order: starting from how individual pages can be configured, to how to create a sidebar or multiple ones, to how to create and manage versions, to how to use multiple docs plugin instances.

Docs-only mode

A freshly initialized Docusaurus site has the following structure:

```
example.com/                           -> generated from `src/pages/index.js`

example.com/docs/intro                 -> generated from `docs/intro.md`
example.com/docs/tutorial-basics/...   -> generated from `docs/tutorial-basics/...`
...

example.com/blog/2021/08/26/welcome    -> generated from `blog/2021-08-26-welcome/index.md`
example.com/blog/2021/08/01/mdx-blog-post  -> generated from `blog/2021-08-01-mdx-blog-post.mdx`
...
```

All docs will be served under the subroute `docs/`. But what if **your site only has docs**, or you want to prioritize your docs by putting them at the root?

Assume that you have the following in your configuration:

docusaurus.config.js
```
module.exports = {
  // ...
  presets: [
    '@docusaurus/preset-classic',
    {
      docs: {
        /* docs plugin options */
      },
      blog: {
        /* blog plugin options */
      },
      // ...
    },
  ],
};
```

To enter docs-only mode, change it to like this:

docusaurus.config.js

```js
module.exports = {
  // ...
  presets: [
    '@docusaurus/preset-classic',
    {
      docs: {
        routeBasePath: '/', // Serve the docs at the site's root
        /* other docs plugin options */
      },
      blog: false, // Optional: disable the blog plugin
      // ...
    },
  ],
};
```

Note that you **don't necessarily have to give up on using the blog** or other plugins; all that `routeBasePath: '/'` does is that instead of serving the docs through `https://example.com/docs/some-doc`, they are now at the site root: `https://example.com/some-doc`. The blog, if enabled, can still be accessed through the `blog/` subroute.

Don't forget to put some page at the root (`https://example.com/`) through adding the front matter:

docs/intro.md

```
---
slug: /
---
```

This page will be the home page when users visit https://example.com/.

⚠ CAUTION

If you added `slug: /` to a doc to make it the homepage, you should delete the existing homepage at `./src/pages/index.js`, or else there will be two files mapping to the same route!

Now, the site's structure will be like the following:

```
example.com/                    -> generated from `docs/intro.md`
example.com/tutorial-basics/... -> generated from `docs/tutorial-basics/...`
...
```

💡 TIP

There's also a "blog-only mode" for those who only want to use the blog feature of Docusaurus 2. You can use the same method detailed above. Follow the setup instructions on [Blog-only mode](#).

✏ Edit this page

*Last updated on **Mar 23, 2023** by **Sébastien Lorber***

Learn

[Introduction](#)

[Installation](#)

[Migration from v1 to v2](#)

Community

[Stack Overflow](#) ⬈

[Feature Requests](#)

[Discord](#) ⬈

[Help](#)

More

Blog

Changelog

GitHub 🔗

Twitter 🔗

Legal

Privacy 🔗

Terms 🔗

Data Policy 🔗

Cookie Policy 🔗