Docusaurus

🏠 Introduction
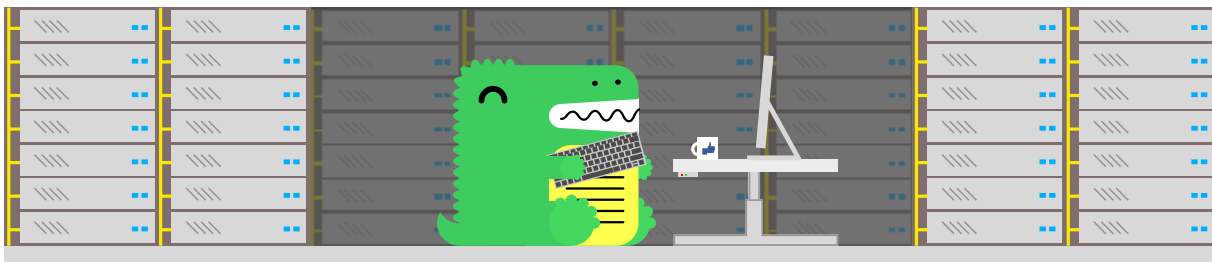
Version: 2.4.0

On this page

Introduction

Docusaurus will help you ship a **beautiful documentation site in no time**.

Building a custom tech stack is expensive. Instead, **focus on your content** and just write Markdown files.

Ready for more? Use **advanced features** like versioning, i18n, search and theme customizations.

Check the **best Docusaurus sites** for inspiration and read some **testimonials**.

Docusaurus is a **static-site generator**. It builds a **single-page application** with fast client-side navigation, leveraging the full power of **React** to make your site interactive. It provides out-of-the-box **documentation features** but can be used to create **any kind of site** (personal website, product, blog, marketing landing pages, etc).

Fast Track

Understand Docusaurus in **5 minutes** by playing!

Create a new Docusaurus site and follow the **very short** embedded tutorial.

Install Node.js and create a new Docusaurus site:

```
npx create-docusaurus@latest my-website classic
```

Start the site:

```
cd my-website
npx docusaurus start
```

Open http://localhost:3000 and follow the tutorial.

💡 TIP

Use **docusaurus.new** to test Docusaurus immediately in your browser!

Or read the **5-minute tutorial** online.

Docusaurus: Documentation Made Easy

In this presentation at Algolia Community Event, Meta Open Source team shared a brief walk-through of Docusaurus. They covered how to get started with the project, enable plugins, and set up functionalities like documentation and blogging.

Migrating from v1

Docusaurus v2 has been a total rewrite from Docusaurus v1, taking advantage of a completely modernized toolchain. After v2's official release, we highly encourage you to **use Docusaurus v2 over Docusaurus v1**, as Docusaurus v1 has been deprecated.

A lot of users are already using Docusaurus v2 (trends).

Use Docusaurus v2 if:

- You want a modern Jamstack documentation site
- You want a single-page application (SPA) with client-side routing
- You want the full power of React and MDX
- You do not need support for IE11

Use Docusaurus v1 if:

- You don't want a single-page application (SPA)
- You need support for IE11 (...do you? IE has already reached end-of-life and is no longer officially supported)

For existing v1 users that are seeking to upgrade to v2, you can follow our migration guide.

Features

Docusaurus is built with high attention to the developer and contributor experience.

- **Built with   and React**   :
    - Extend and customize with React
    - Gain full control of your site's browsing experience by providing your own React components
- **Pluggable**:
    - Bootstrap your site with a basic template, then use advanced features and plugins
    - Open source your plugins to share with the community
- **Developer experience**:
    - Start writing your docs right now
    - Universal configuration entry point to make it more maintainable by contributors
    - Hot reloading with lightning-fast incremental build on changes
    - Route-based code and data splitting
    - Publish to GitHub Pages, Netlify, Vercel, and other deployment services with ease

Our shared goal—to help your users quickly find what they need and understand your products better. We share our best practices to help you build your docs site right and well.

- **SEO friendly**:
    - HTML files are statically generated for every possible path.
    - Page-specific SEO to help your users land on your official docs directly relating their problems at hand.
- **Powered by MDX**:
    - Write interactive components via JSX and React embedded in Markdown.
    - Share your code in live editors to get your users to love your products on the spot.
- **Search**: Your full site is searchable.
- **Document Versioning**: Helps you keep documentation in sync with project releases.
- **Internationalization (i18n)**: Translate your site in multiple locales.

Docusaurus 2 is born to be compassionately accessible to all your users, and lightning-fast.

- **Lightning-fast**. Docusaurus 2 follows the PRPL Pattern that makes sure your content loads blazing fast.
- **Accessible**. Attention to accessibility, making your site equally accessible to all users.

Design principles

- **Little to learn**. Docusaurus should be easy to learn and use as the API is quite small. Most things will still be achievable by users, even if it takes them more code and more time to write. Not having abstractions is better than having the wrong abstractions, and we don't want users to have to hack around the wrong abstractions. Mandatory talk—Minimal API Surface Area.
- **Intuitive**. Users will not feel overwhelmed when looking at the project directory of a Docusaurus project or adding new features. It should look intuitive and easy to build on top of, using approaches they are familiar with.
- **Layered architecture**. The separations of concerns between each layer of our stack (content/theming/styling) should be clear —well-abstracted and modular.
- **Sensible defaults**. Common and popular performance optimizations and configurations will be done for users but they are given the option to override them.
- **No vendor lock-in**. Users are not required to use the default plugins or CSS, although they are highly encouraged to. Certain

core infrastructures like React Loadable and React Router cannot be swapped because we do default performance optimization on them, but not higher-level ones. Choice of Markdown engines, CSS frameworks, CSS methodology, and other architectures will be entirely up to users.

We believe that, as developers, knowing how a library works helps us become better at using it. Hence we're dedicating effort to explaining the architecture and various components of Docusaurus with the hope that users reading it will gain a deeper understanding of the tool and be even more proficient in using it.

Comparison with other tools

Across all static site generators, Docusaurus has a unique focus on documentation sites and has many out-of-the-box features.

We've also studied other main static site generators and would like to share our insights on the comparison, hopefully helping you navigate through the prismatic choices out there.

Gatsby

Gatsby is packed with a lot of features, has a rich ecosystem of plugins, and is capable of doing everything that Docusaurus does. Naturally, that comes at a cost of a higher learning curve. Gatsby does many things well and is suitable for building many types of websites. On the other hand, Docusaurus tries to do one thing super well - be the best tool for writing and publishing content.

GraphQL is also pretty core to Gatsby, although you don't necessarily need GraphQL to build a Gatsby site. In most cases when building static websites, you won't need the flexibility that GraphQL provides.

Many aspects of Docusaurus 2 were inspired by the best things about Gatsby and it's a great alternative.

Docz is a Gatsby theme to build documentation websites. It is currently less featured than Docusaurus.

Next.js

Next.js is another very popular hybrid React framework. It can help you build a good documentation website, but it is not opinionated toward the documentation use-case, and it will require a lot more work to implement what Docusaurus provides out-of-the-box.

Nextra is an opinionated static site generator built on top of Next.js. It is currently less featured than Docusaurus.

VuePress

VuePress has many similarities with Docusaurus - both focus heavily on content-centric website and provides tailored documentation features out of the box. However, VuePress is powered by Vue, while Docusaurus is powered by React. If you want a Vue-based solution, VuePress would be a decent choice.

MkDocs

MkDocs is a popular Python static site generator with value propositions similar to Docusaurus.

It is a good option if you don't need a single-page application and don't plan to leverage React.

Material for MkDocs is a beautiful theme.

Docsify

Docsify makes it easy to create a documentation website, but is not a static-site generator and is not SEO friendly.

GitBook

GitBook has a very clean design and has been used by many open source projects. With its focus shifting towards a commercial product rather than an open-source tool, many of its requirements no longer fit the needs of open source projects' documentation sites. As a result, many have turned to other products. You may read about Redux's switch to Docusaurus here.

Currently, GitBook is only free for open-source and non-profit teams. Docusaurus is free for everyone.

Jekyll

Jekyll is one of the most mature static site generators around and has been a great tool to use — in fact, before Docusaurus, most of Facebook's Open Source websites are/were built on Jekyll! It is extremely simple to get started. We want to bring a similar developer experience as building a static site with Jekyll.

In comparison with statically generated HTML and interactivity added using `<script />` tags, Docusaurus sites are React apps. Using modern JavaScript ecosystem tooling, we hope to set new standards on doc sites' performance, asset building pipeline and optimizations, and ease to set up.

Staying informed

- GitHub
- Twitter
- Blog
- Discord

Something missing?

If you find issues with the documentation or have suggestions on how to improve the documentation or the project in general, please file an issue for us, or send a tweet mentioning the @docusaurus Twitter account.

For new feature requests, you can create a post on our feature requests board (Canny), which is a handy tool for road-mapping and allows for sorting by upvotes, which gives the core team a better indicator of what features are in high demand, as compared to GitHub issues which are harder to triage. Refrain from making a Pull Request for new features (especially large ones) as someone might already be working on it or will be part of our roadmap. Talk to us first!

✏️ 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 ↗