

How to Create a Static Website for Your Documentation Based on mkdocs and mkdocs-material

MkDocs is a tool built to create static websites from raw markdown files.

Other alternatives include [Sphinx](#), and [Jekyll](#).

We used MkDocs to create [ISE Engineering Fundamentals Playbook](#) static website from the contents in [the GitHub repository](#). Then we deployed it to [GitHub Pages](#).

We found MkDocs to be a good choice since:

1. It's easy to set up and looks great even with the vanilla version.
2. It works well with markdown, which is what we already have in the Playbook.
3. It uses a Python stack which is friendly to many contributors of this Playbook.

For comparison, Sphinx mainly generates docs from restructured-text (rst) format, and Jekyll is written in Ruby.

To setup an MkDocs website, the main assets needed are:

1. An `mkdocs.yaml` file, similar to the one we have [in the Playbook](#). This is the configuration file that defines the appearance of the website, the navigation, the plugins used and more.

2. A folder named `docs` (the default value for the directory) that contains the documentation source files.
3. A GitHub Action for automatically generating the website (e.g. on every commit to main), similar to [this one from the Playbook](#).
4. A list of plugins used during the build phase of the website. We specified ours [here](#). And these are the plugins we've used:
 - [Material for MkDocs](#): Material design appearance and user experience.
 - [pymdown-extensions](#): Improves the appearance of markdown based content.
 - [mdx_truly_sane_lists](#): For defining the indent level for lists without having to refactor the entire documentation we already had in the Playbook.

Setting up locally is very easy. See [Getting Started with MkDocs](#) for details.

For publishing the website, there's a [good integration with GitHub for storing the website as a GitHub Page](#).

Resources

- [MkDocs Plugins](#)
 - [The best MkDocs plugins and customizations](#)
-