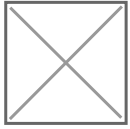


Documentation Generators

- [Read The Docs](#)

Read The Docs

Features



Static Generators

Read The Docs (RTD) is a documentation generator that utilizes either [Sphinx](#) or [MkDocs](#) to generate the documentation site and theme, depending on your choice. The [RTD GitHub](#) houses the configurations and files that generates the [Official Website](#). This provides a good example of the many features RTD offers, and allows you to step through configurations in a more feature-rich and complete setup.

Since there are different choices for the backend static site generator, the deployment procedure is different depending on which one you choose. Below, we will cover deploying Read The Docs with Sphinx and MkDocs. These two sections are each entirely different deployments of RTD, where the difference is only the [Backend Static Site Generator](#) that is used to build your documentation.

Alternatively, you can simply sign up on the RTD official website and allow them to host the documentation for you by linking a repository to your user account and following the [Quick Start Instructions](#). Since this is nearly turn-key, we won't cover deploying RTD this way.

Continuous Version Based Documentation

RTD utilizes webhooks within your VCS (GitHub, GitLab, BitBucket) to automatically build and deploy version-based documentation for your projects. If you've deployed a project to RTD, it automatically creates these webhooks for you. See the [Webhook and Automation](#) documentation for more info.

There are also two independent versions of RTD, one community and one for business related documentation that also supports private repositories and other security features. When visiting <https://readthedocs.org>, we are viewing community version. When visiting <https://readthedocs.com>, we are viewing the business version.

For the below documentation, we will be using the [Community Version](#) of Read The Docs since the business version is [Subscription Based](#).

Downloadable Documentation

RTD supports downloading any hosted documentation in PDF, ePub, and Zipped HTML formats.

Text Search

“Need a place to host your Sphinx docs? readthedocs.org hosts a lot of Sphinx docs already, and integrates well with projects' source control. It also features a powerful built-in search that exceeds the possibilities of Sphinx' JavaScript-based offline search. - [Sphinx](#)

RTD adds search functionality to your documentation that, in comparison to the sphinx builtin search capabilities, is a noteworthy improvement.

RTD with MkDocs

Deploying RTD with MkDocs is as simple as first deploying a MkDocs instance, and then importing this documentation into RTD. The RTD documentation provides instructions for [Getting Started with MkDocs](#), which outlines the process below.

Install MkDocs

“MkDocs supports Python versions 3.5, 3.6, 3.7, 3.8, and pypy3. If you have and use a package manager (such as apt-get, dnf, homebrew, yum, chocolatey, etc.) to install packages on your system, then you may want to search for a "MkDocs" package. If your package manager does not have a recent "MkDocs" package, you can still use your package manager to install "Python" and "pip". Then you can use pip to install MkDocs. - [MkDocs Official Documentation](#)

The quote above seems to suggest the preferred method for installation is `pip`, though it is possible to use your package manager. To make these instructions more portable, I'll only cover `mkdocs` installation using `pip`.

```
sudo apt update && sudo apt upgrade
sudo apt install python-pip -y
```

```
pip install --upgrade pip
sudo pip install mkdocs
```

MkDocs Man Pages

If needed, MkDocs provides man pages that can be installed with [click-man](#) by running the following commands

An explanation on why the man pages are not installed automatically with `pip install mkdocs` is described within the [click-man GitHub README](#)

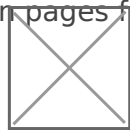
```
pip install click-man
click-man --target path/to/man/pages mkdocs
```

Following the [Man Pages Index Guidelines](#) we can install the man pages to the appropriate section using the following command

```
click-man --target /usr/share/man/man1 mkdocs
```

Now we can see that the man pages for `mkdocs` have been installed and reference them as needed

directly within our terminal -



Configuring MkDocs

MkDocs offers a range of custom configuration options which can be tailored to suit your needs aesthetically and in terms of functionality. To get started, run the following commands

```
mkdocs new my-project
cd my-project
```

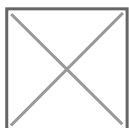
This will have created a new directory relative to your CWD `./my-project` generated two files within, `./my-project/mkdocs.yml` and `./my-project/docs/index.md`. For this section, we will be modifying `mkdocs.yml`

```
site_name: My Docs
nav:
  - home: index.md
theme: readthedocs
```

This adds a title for our documentation and says two things, we've created an `./my-project/docs/index.md` that represents the home page content, and we want to use the [ReadTheDocs](#) theme for our instance of MkDocs. Now to apply these changes, we rebuild the documentation

```
mkdocs build
```

This generates raw HTML files within `./my-project/site`, which will have the following file structure



These files do not need to be tracked by your VCS since they are generated, and if using git to track this MkDocs instance can be ignored with `echo "site/" >> .gitignore`

To update the generated file index with any removals or renamed pages, you'll need to run the following command

```
mkdocs build --clean
```

Once these files are updated, provided that you are working locally and hosting remotely, you can update your remote host with the new generated files -

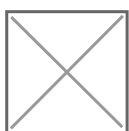
```
mkdocs build  
scp -r ./site user@host:/path/to/web/root
```

Alternatively, you can work directly on the host and serve the `./my-project/site` directory directly.

Themes

Community Themes

MkDocs is very themeable, you'll notice with the screenshot below that the [Windmil](#) theme is quite different from the [ReadTheDocs](#) theme used on the [RTD Documentation](#).



Plugins

Mkdocs supports plugins which extend the functionality in more specific use cases. Check the [GitHub Wiki](#) for a [complete list of plugins](#).

This feature came from a request for multiple projects documentation within a single MkDocs instance - [Issue #265](#). Once plugins were added to MkDocs, [Spotify/mkdocs-monorepo-plugin](#) provided a plugin to support this feature, allowing for multiple documentation folders within a single MkDocs. Spotify even hosts a [Live Demo](#) showcasing the results of using the plugin and has well-written instructions on how to use it.

Deploying MkDocs

The files generated with MkDocs can be deployed using either GitHub Pages, RTD, or a local webserver like NGINX or Apache. Below, we look at a simple NGINX configuration to serve MkDocs

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events { }

http {
    include mime.types;

    # Redirect root domains
    server {
        listen 80;
        server_name domain.com www.domain.com;
        # Pass to SSL
        return 301 https://www.domain.com$request_uri;
    }

    # SSL
    server {
        server_name domain.com www.domain.com;
        server_tokens off;
        listen 443 ssl;
        ssl_certificate /etc/letsencrypt/live/domain.com/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/domain.com/privkey.pem;

        # Configure Server-wide logging
        access_log /var/log/nginx/access.log;
```

```
error_log /var/log/nginx/error.log;
```

```
# Pass to container
```

```
location / {
```

```
    # Point NGINX to static site files
```

```
    root /path/to/my-project/site;
```

```
    include proxy_params;
```

```
}
```

```
}
```

```
}
```

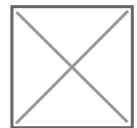
As a final step, we restart the nginx service and navigate to our domain in a web browser.

```
sudo systemctl reload nginx
```

For demonstration, I've duplicated this page within my mkdocs instance, shown in the screenshot below



This MkDocs Demo



You can even search for keywords right out of the box with the builtin search field -

RTD with Sphinx

WIP