DjangoCon Europe 2021
Dynamic Static Sites with Django and Sphinx

Hello

Carlton Gibson
@carltongibson

Django Fellowship Program

Django Fellowship Program

djangoproject.com/fundraising/

Django Chat

djangochat.com

btn.dev

Coming Summer 2021

Dynamic Static Sites

With Django and Sphinx

Sphinx?

The documentation builder?

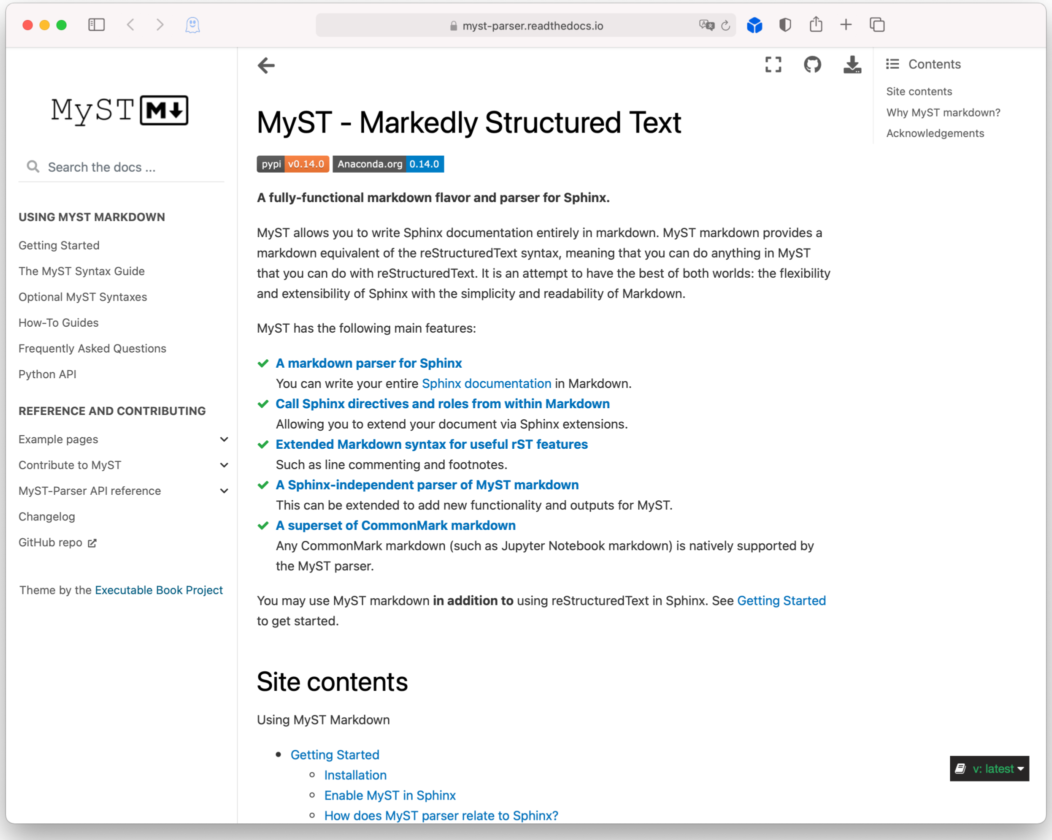
Sphinx builds great static sites too

“Static Sites with Sphinx and Markdown” by Paul Everitt (JetBrains)

- Talks
  - PyCon APAC 2020.
  - PyCon US 2021
- Tutorial
  - https://www.jetbrains.com/pycharm/guide/tutorials/sphinx_sites/

Markdown

MyST: a Markdown parser for Sphinx



MyST - Markedly Structured Text

pypi v0.14.0 Anaconda.org 0.14.0

A fully-functional markdown flavor and parser for Sphinx.

MyST allows you to write Sphinx documentation entirely in markdown. MyST markdown provides a markdown equivalent of the reStructuredText syntax, meaning that you can do anything in MyST that you can do with reStructuredText. It is an attempt to have the best of both worlds: the flexibility and extensibility of Sphinx with the simplicity and readability of Markdown.

MyST has the following main features:

✓ **A markdown parser for Sphinx**
You can write your entire Sphinx documentation in Markdown.
✓ **Call Sphinx directives and roles from within Markdown**
Allowing you to extend your document via Sphinx extensions.
✓ **Extended Markdown syntax for useful rST features**
Such as line commenting and footnotes.
✓ **A Sphinx-independent parser of MyST markdown**
This can be extended to add new functionality and outputs for MyST.
✓ **A superset of CommonMark markdown**
Any CommonMark markdown (such as Jupyter Notebook markdown) is natively supported by the MyST parser.

You may use MyST markdown **in addition to** using reStructuredText in Sphinx. See Getting Started to get started.

Site contents

Using MyST Markdown

- Getting Started
  - Installation
  - Enable MyST in Sphinx
  - How does MyST parser relate to Sphinx?

Using the MyST Parser

```
# Install from PyPI
$ pip install myst-parser

# Enable in Sphinx’s conf.py
extensions = [
    "myst_parser",
]
```

https://myst-parser.readthedocs.io/en/latest/

Markdown

But what about dynamic?

We want to use Django

A CMS?

But what about my editor?

But what about git?

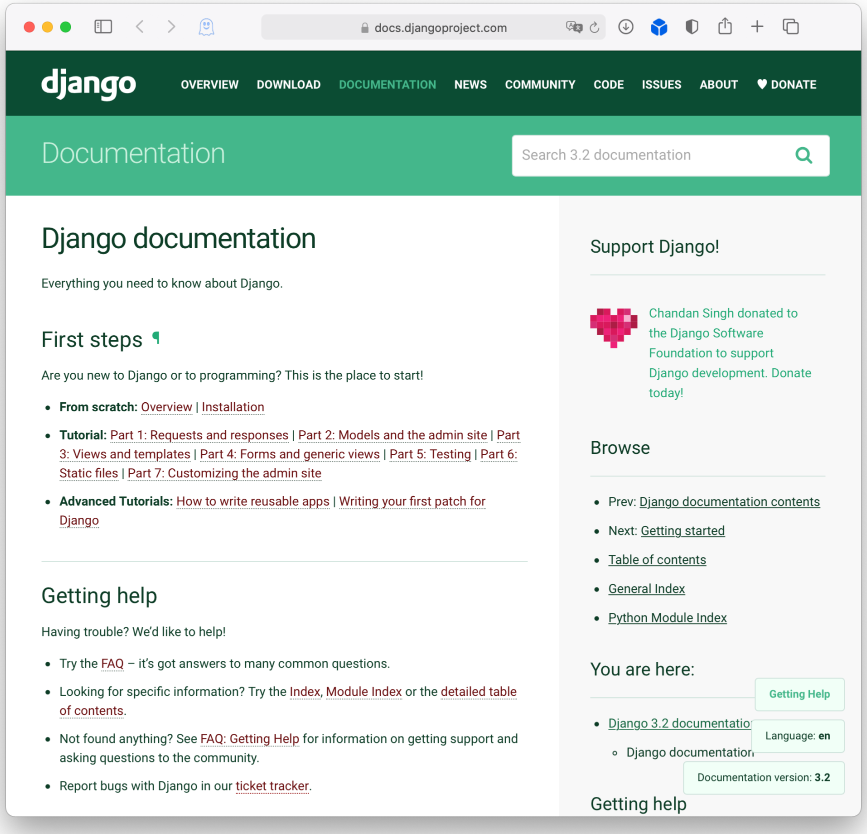
But what about Sphinx?

But can we use Sphinx?

Sphinx Builders

HTML • PDF • EPUB • Man pages • JSON

The Django Docs



Django documentation

Everything you need to know about Django.

First steps ¶

Are you new to Django or to programming? This is the place to start!

- **From scratch:** Overview | Installation
- **Tutorial:** Part 1: Requests and responses | Part 2: Models and the admin site | Part 3: Views and templates | Part 4: Forms and generic views | Part 5: Testing | Part 6: Static files | Part 7: Customizing the admin site
- **Advanced Tutorials:** How to write reusable apps | Writing your first patch for Django

Getting help

Having trouble? We'd like to help!

- Try the FAQ – it's got answers to many common questions.
- Looking for specific information? Try the Index, Module Index or the detailed table of contents.
- Not found anything? See FAQ: Getting Help for information on getting support and asking questions to the community.
- Report bugs with Django in our ticket tracker.

Support Django!

Chandan Singh donated to the Django Software Foundation to support Django development. Donate today!

Browse

- Prev: Django documentation contents
- Next: Getting started
- Table of contents
- General Index
- Python Module Index

You are here:

- Django 3.2 documentatio
  - Django documentation

Getting help

django-sphinx-view

noumenal.es/django-sphinx-view/

Setup

```
# Install from PyPI
$ pip install django-sphinx-view

INSTALLED_APPS = [
    ...
    "sphinx_view",
    ...
]
```

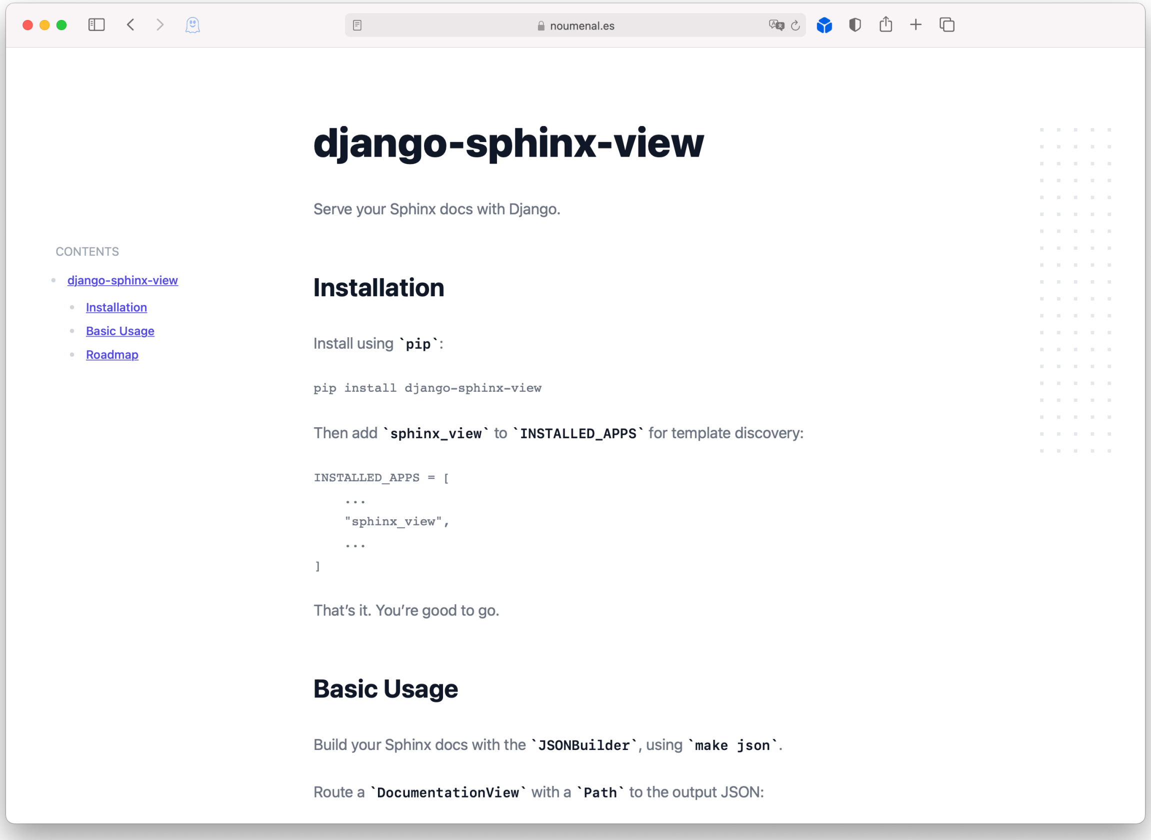
Basic Usage

```
from pathlib import Path


from django.urls import path
from sphinx_view import DocumentationView


urlpatterns = [
    # ...
    path(
        "docs<path:path>",
        DocumentationView.as_view(
            json_build_dir=Path('/path/to/output/json’),
            base_template_name="docs_base.html",
        ),
        name="documentation",
    ),
    # ...
]
```

Basic Usage

```
urlpatterns = [
    # ...
    path(
        "docs<path:path>",
        DocumentationView.as_view(
            json_build_dir=Path('/path/to/output/json’),
            base_template_name="docs_base.html",
        ),
        name="documentation",
    ),
    # ...
]
```

The django-sphinx-view's docs



noumenal.es

CONTENTS

django-sphinx-view

Serve your Sphinx docs with Django.

Installation

Install using `pip`:

```
pip install django-sphinx-view
```

Then add `sphinx_view` to `INSTALLED_APPS` for template discovery:

```
INSTALLED_APPS = [
    ...
    "sphinx_view",
    ...
]
```

That's it. You're good to go.

Basic Usage

Build your Sphinx docs with the `JSONBuilder`, using `make json`.

Route a `DocumentationView` with a `Path` to the output JSON:

Customising

Continuous Deployment

Continuous Deployment

```yaml
# From .github/workflows/main.yml
- name: Install dependencies
  run: |
    python -m pip install --upgrade pip
    pip install Sphinx myst-parser
- name: Build docs
  run: make json
  working-directory: ./docs
```

Continuous Deployment

```
# From .github/workflows/main.yml
- name: rsync files
  working-directory: ./docs
  run: rsync … ./build/json/ …
```

Roadmap

Carlton Gibson
@carltongibson
djangochat.com
btn.dev

django-sphinx-view

noumenal.es/django-sphinx-view/

fin