
Sphinx Bulma Theme Documentation

Release 0.2.4

Gabriel Falcão

Apr 30, 2018

1	Installing	1
2	Configuring	3
3	Writing Python Documentation	5
3.1	Annotating Code	5
4	Theme Options	7
5	Theme Showcase	9
5.1	Admonitions	9
6	Extension Support	11
6.1	httpdomain	11
6.2	sphinxcontrib-mermaid	12
7	Extending the Theme	13
7.1	Install Dependencies	13
7.2	Customizing CSS and JS	13
7.3	Customizing Python Transformations	14
8	Python API Reference	15
8.1	get_html_theme_path()	15
8.2	add_classes_to_node()	15
8.3	process_admonition_node	15
8.4	bulmanize_documentation	16
9	Indices and tables	17
	HTTP Routing Table	19

CHAPTER 1

Installing

```
pip install sphinx-bulma-theme
```



```
import sphinx_bulma_theme

html_theme = "bulma"
html_theme_path = [sphinx_bulma_theme.get_html_theme_path()]

html_theme_options = {
    'show_topbar': True,
    'topbar_class': 'is-info',
    'topbar_logo_class': 'image is-32x32',
}
```

See also:

List with all *theme options*

Writing Python Documentation

The very idea of writing documentation sounds tedious to some, this guide aims to remove mental blocks by synthesizing some of the most common tasks for writing documentation.

By the end of the guide you will know:

- Bootstrap Sphinx documentation
- Configure a theme
- Document python modules internally with docstrings
- Reference documented parts of the code: (*modules, classes, methods, etc*)
- Make the reverse-reference: your code linking to the documentation
- Auto-generate API-reference page for python modules
- Use `sphinx.ext.intersphinx` to reference external modules documented with Sphinx (e.g.: *Flask, requests, boto3*)

3.1 Annotating Code

Sphinx ships with the `sphinx.ext.autodoc` which generates the classic API Reference section of many known open source libraries such as *Flask*, *requests* and *boto3*.

3.1.1 Functions

Example Code

```
def make_user(email, password, admin=False, **kwargs):
    """creates structured user data for storage

    :param email: string
```

(continues on next page)

(continued from previous page)

```
:param password: in plain-text
:param admin: bool - indicates user has all priviledges
:param kwargs: extra data

:returns: a :py:class:`dict` with user data
"""
data = locals()
data.update(data.pop('kwargs'))
return data
```

Reference function in a rst document

```
.. autofunction:: sphinx_bulma_theme.example.make_user
```

Rendered HTML:

sphinx_bulma_theme.example.**make_user** (*email*, *password*, *admin=False*, ***kwargs*)
creates structured user data for storage

Parameters

- **email** – string
- **password** – in plain-text
- **admin** – bool - indicates user has all priviledges
- **kwargs** – extra data

Returns a `dict` with user data

CHAPTER 4

Theme Options

default_admonition_class

Default color of admonitions

type: `str`

default: `is-light`

breadcrumbs_at_top

Whether to show breadcrumbs on the top of the page.

- **type:** `bool`
- **default:** `False`

canonical_url

Force all internal links to use this value as base url

type: `str` **default:** `None`

collapse_navigation

type: `bool` **default:** `False`

content_margin_left

type: `str` **default:** `None` **example:** `20px`

content_padding_left

type: `str` **default:** `None` **example:** `20px`

content_padding_top

If set to `None` the theme will determine the best default depending on whether the *topbar is visible*

type: `str` **default:** `None` **example:** `20px`

display_version

type: `bool` **default:** `True`

logo_only

Define a *logo_path* and set this to True if you want to omit your project title and only show the logo.

type: `bool` **default:** False

logo_path

Relative path to your project's source/_static

type: `str` **default:** None **example:** my-project-name.png

navigation_depth

The maximum depth of the toctree

type: `int` **default:** 6

show_topbar

Whether to show topbar or sidebar

type: `bool` **default:** False

sidebar_class

Sets the color of project title on sidebar

type: `str` **default:** has-text-dark **example:** my-project-name.png

sidebar_container_class

Sets the width of the sidebar

type: `str` **default:** is-3

sidebar_right

Whether to show sidebar on the right side of the page

type: `bool` **default:** False

theme_sidebar_style

Defines the inline CSS rules for the sidebar. The theme will determine the best margin for the sidebar by default but you can customize it here.

type: `str` **default:** None **example:** margin: 32px

topbar_class

type: `str` **default:** is-light

topbar_logo_class

type: `str` **default:** image is-32x32

5.1 Admonitions

Attention: Beware

Tip: Just the tip

Warning: This is a fairly descriptive message

See also:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam eu molestie dui, nec finibus odio. Nullam blandit rhoncus felis, vel vulputate neque posuere vel. Quisque eu semper ante. Sed ornare pellentesque magna, sit amet rhoncus dui imperdiet non. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nullam lacinia nisi ac placerat interdum. Phasellus sapien quam, cursus nec dolor vel, vulputate tempor lacus. Vestibulum a libero eget lacus semper condimentum vel in velit. Suspendisse volutpat augue et odio fringilla lacinia.

Note: There are many other directives available

Important: What is?

Caution: Oooops

Danger: Problems can happen

Error: Explains why

Hint: Read the source code

6.1 httpdomain

extension docs

GET `/users/(int: user_id)/posts/`
tag The posts tagged with *tag* that the user (*user_id*) wrote.

Example request:

```
GET /users/123/posts/web HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/javascript

[
  {
    "post_id": 12345,
    "author_id": 123,
    "tags": ["server", "web"],
    "subject": "I tried Nginx"
  },
  {
    "post_id": 12346,
    "author_id": 123,
    "tags": ["html5", "standards", "web"],
    "subject": "We go to HTML 5"
  }
]
```

Query Parameters

- **sort** – one of `hit`, `created-at`
- **offset** – offset number. default is 0
- **limit** – limit number. default is 30

Request Headers

- **Accept** – the response content type depends on *Accept* header
- **Authorization** – optional OAuth token to authenticate

Response Headers

- **Content-Type** – this depends on *Accept* header of request

Status Codes

- **200 OK** – no error
- **404 Not Found** – there's no user

6.2 sphinxcontrib-mermaid

extension docs

Extending the Theme

This theme makes efforts to be as extensible as possible.

In this guide we will learn how to modify docutils nodes before rendering, tweak CSS and apply javascript hacks.

7.1 Install Dependencies

We will need:

- nodejs v9.11.1 or superior
- npm 5.8.0 or superior

```
# clone the repo
git clone https://github.com/gabrielfalcao/sphinx-bulma-theme.git

cd sphinx-bulma-theme
npm install --dev
```

7.2 Customizing CSS and JS

Sphinx has its own opinion of what CSS classes should be in what elements. The theme attempts to inject the appropriate classes in certain nodes through `sphinx_bulma_theme.bulmanize_documentation()`.

But that's not enough.

To bridge the gap between bulma styles and Sphinx's optionated CSS classes this theme uses special CSS and Javascript.

Don't worry it's not as bad as it sounds.

Looking at the source tree we can spot 2 files:

- `sphinx_bulma_theme/sphinx-bulma.src.sass`

- sphinx_bulma_theme/sphinx-bulma.src.js

7.2.1 sphinx-bulma.src.sass

The good news: Bulma is build on [sass](#) so this theme makes little effort in extending it.

To build the CSS file run:

```
make sass
```

Under the hood it will run:

```
node-sass --include-path scss sphinx_bulma_theme/sphinx-bulma.src.sass sphinx_bulma_
↪theme/static/css/theme.css
```

7.2.2 sphinx-bulma.src.js

The good news: ES6 support :)

To build the JS file run:

```
make webpack
```

Under the hood it will run:

```
export NODE_ENV=production
export NODE_PATH=../../node_modules:../sphinx_bulma_theme:$NODE_PATH
webpack -p
```

7.3 Customizing Python Transformations

We will modify the file sphinx_bulma_theme/__init__.py.

The execution entrypoint we will look for is `bulmanize_documentation()`

See also:

`sphinx_bulma_theme.add_classes_to_node()` and `process_admonition_node()`

8.1 `get_html_theme_path()`

`sphinx_bulma_theme.get_html_theme_path()`

For use inside Sphinx's `conf.py` file:

```
import sphinx_bulma_theme

html_theme = "bulma"
html_theme_path = [sphinx_bulma_theme.get_html_theme_path()]

html_theme_options = {
    'show_topbar': False,
    'sidebar_right': True,
}
```

8.2 `add_classes_to_node()`

`sphinx_bulma_theme.add_classes_to_node(class_names, node)`

sets a list of class names in the node. Used for injecting [bulma modifiers](#) in the final HTML generated by Sphinx.

Parameters

- **class_names** – a list of strings
- **node** – the target node

8.3 `process_admonition_node`

`sphinx_bulma_theme.process_admonition_node(node, admonition_type, color, size)`

applies the given color and size modifiers to admonition nodes

Parameters

- **node** – the target `docutils.nodes.Admonition` (subclass) node
- **admonition_type** – see `sphinx_bulma_theme.admonition_map`
- **color** – a string, such as `is-danger`
- **size** – a string, such as `is-small` or **None**

8.4 bulmanize_documentation

`sphinx_bulma_theme.bulmanize_documentation` (*app, doctree, fromdocname*)

applies necessary internal node transformations before HTML rendering:

- Loads the `admonition_class` from *Theme Options*
- Iterate over `sphinx_bulma_theme.admonition_map`

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`
- `search`

HTTP Routing Table

/users

GET /users/(int:user_id)/posts/(tag),

11

A

`add_classes_to_node()` (in module `sphinx_bulma_theme`), 15

B

`breadcrumbs_at_top` (built-in variable), 7
`bulmanize_documentation()` (in module `sphinx_bulma_theme`), 16

C

`canonical_url` (built-in variable), 7
`collapse_navigation` (built-in variable), 7
`content_margin_left` (built-in variable), 7
`content_padding_left` (built-in variable), 7
`content_padding_top` (built-in variable), 7

D

`default_admonition_class` (built-in variable), 7
`display_version` (built-in variable), 7

G

`get_html_theme_path()` (in module `sphinx_bulma_theme`), 15

L

`logo_only` (built-in variable), 7
`logo_path` (built-in variable), 8

M

`make_user()` (in module `sphinx_bulma_theme.example`), 6

N

`navigation_depth` (built-in variable), 8

P

`process_admonition_node()` (in module `sphinx_bulma_theme`), 15

S

`show_topbar` (built-in variable), 8
`sidebar_class` (built-in variable), 8
`sidebar_container_class` (built-in variable), 8
`sidebar_right` (built-in variable), 8

T

`theme_sidebar_style` (built-in variable), 8
`topbar_class` (built-in variable), 8
`topbar_logo_class` (built-in variable), 8