
Jupyter-Alabaster-Theme Documentation

Release 0.4.0

Jeff Forcier, Project Jupyter, and contributors

May 27, 2017

Contents

1	User	3
1.1	How to install	3
1.2	Important Notes	4
2	Development	5
2.1	Running the Docs Locally	5
2.2	Changing the Theme	5

A Sphinx theme for [Project Jupyter's](#) documentation, based on Jeff Forcier's [Alabaster theme](#).

Contents:

1.1 How to install

To use this theme on any Sphinx based documentation, follow these steps:

Install the `jupyter_alabaster_theme` package using pip:

```
pip install jupyter_alabaster_theme
```

Edit your `conf.py` file:

1. Set the theme to `jupyter_alabaster_theme`:

```
html_theme = 'jupyter_alabaster_theme'
```

2. Add `jupyter_alabaster_theme` to the list of extensions:

```
extensions = [  
    ...  
    'jupyter_alabaster_theme',  
]
```

3. At the bottom of `conf.py`, if the following block of code is present, remove it:

```
if not on_rtd:  
    # only import and set the theme if we're building docs locally  
    import sphinx_rtd_theme  
    html_theme = 'sphinx_rtd_theme'  
    html_theme_path = [sphinx_rtd_theme.get_html_theme_path()]
```

Update Documentation Dependencies

To get your documentation to build, you will need to update its build dependencies. Update your `environment.yml` or `requirements.txt` to depend on `jupyter_alabaster_theme` to get the build working on ReadTheDocs. Note, we have not yet released a [Conda](#) package for this theme, so you will need to list it in the `environment.yml` file under the `pip` section.

1.2 Important Notes

- Avoid using `captions` in the `toctree` since that is not accessible to mobile navigation menus and the breadcrumbs.
- Avoid adding subsections that are on the same page as the section to the `toctree`. Subsections are not accessible to the mobile navigation menus and breadcrumbs. This can be avoided by limiting `maxdepth` for a `toctree`. Another option is to set the `titlesonly` option:

```
.. toctree::
    :titlesonly:

    title1
    title2
```

- More information about the `toctree` can be found at the [Sphinx documentation site](#)
- The theme itself sets `html_sidebars` to include a custom sidebar navigation template. If you want to use different sidebar templates, simply set your own in `conf.py`, to override the theme's defaults. Otherwise you can add more by:

```
html_sidebars.update(
    # Additional sidebars can be added here
)
```


2.1 Running the Docs Locally

- Fork the repository and clone it locally.
- `cd` into the Jupyter-Alabaster-Theme directory
- Consider using a `virtualenv` or `conda environment` to manage packages.
- Install the packages (from the root of the directory) with `pip`

```
pip install -r dev-requirements.txt
```

- `cd` into the docs directory
- The docs can be built with the following:

```
make clean && make html_theme
```

The docs will be built in `build/html`. They can be viewed by starting an HTTP server and navigating to `0.0.0.0:8000` in your web browser.

```
python3 -m http.server
```

2.2 Changing the Theme

Modifying CSS:

Do not change CSS directly, instead change Post CSS files. This project is using Post CSS, specifically the `css-next` module. Post CSS allows us to have nested CSS selectors and makes it easier to read them.

To Add a Post CSS file:

1. Create a file called `foo.pcss` in `jupyter_alabaster_theme/static/pcss`

2. Add `@mimport foo.css` to `index.css`.
3. In `package.json` add a script with a name `css:foo` and action `"postcss jupyter_alabaster_theme/static/pcss/foo.pcss -o jupyter_alabaster_theme/static/css/foo.css"`,
4. Update `css` script in `package.json` to include `npm run css:foo`.

To compile Post CSS into CSS:

```
npm run css
```

Modifying Sphinx templates:

1. Make appropriate changes to templates.
2. Uninstall `jupyter_alabaster_theme`:

```
pip uninstall jupyter_alabaster_theme
```

3. Install the package:

```
pip install .
```

4. Re-build the Sphinx documentation (from the appropriate directory) and refresh your browser.

```
make clean && make html
```