Python-Deployment Documentation

Release 0.0.1

Lennart Schüler

May 29, 2019

Contents

1 Python Deployment **1**
- 1.1 Documentation with Sphinx and Read the Docs . . . 1
  - 1.1.1 Setting up Sphinx . . . 1
  - 1.1.2 Publishing on Read the Docs . . . 4
- 1.2 Requirements . . . 4
- 1.3 Indices and tables . . . 4
- 1.4 License . . . 4

2 Python Deployment API **5**
- 2.1 deployment . . . 6
  - 2.1.1 deployment.deploy . . . 7

CHAPTER 1

Python Deployment

Examples of Python Deployment Workflows

1.1 Documentation with Sphinx and Read the Docs

1.1.1 Setting up Sphinx

In order to generate a documentation from the docstrings we are going to use Sphinx.

1. Create a *docs* directory in your projects main directory.

```
mkdir docs
cd docs
```

2. Use the quickstart script provided by Sphinx.

```
sphinx-quickstart
```

And use the following settings.

```
$sphinx-quickstart
Welcome to the Sphinx 1.5.6 quickstart utility.

Please enter values for the following settings (just press Enter to
accept a default value, if one is given in brackets).

Enter the root path for documentation.
> Root path for the documentation [.]:

You have two options for placing the build directory for Sphinx output.
Either, you use a directory "_build" within the root path, or you separate
"source" and "build" directories within the root path.
> Separate source and build directories (y/n) [n]: y

Inside the root directory, two more directories will be created; "_templates"
for custom HTML templates and "_static" for custom stylesheets and other static
files. You can enter another prefix (such as ".") to replace the underscore.
> Name prefix for templates and static dir [_]:

The project name will occur in several places in the built documentation.
> Project name: Python-Deployment
> Author name(s): Lennart Schüler

Sphinx has the notion of a "version" and a "release" for the
software. Each version can have multiple releases. For example, for
Python the version is something like 2.5 or 3.0, while the release is
something like 2.5.1 or 3.0a1.  If you don't need this dual structure,
just set both to the same value.
> Project version []: 0.0.1
> Project release [0.0.1]:

If the documents are to be written in a language other than English,
you can select a language here by its language code. Sphinx will then
translate text that it generates into that language.

For a list of supported codes, see
http://sphinx-doc.org/config.html#confval-language.
> Project language [en]:

The file name suffix for source files. Commonly, this is either ".txt"
or ".rst".  Only files with this suffix are considered documents.
> Source file suffix [.rst]:

One document is special in that it is considered the top node of the
"contents tree", that is, it is the root of the hierarchical structure
of the documents. Normally, this is "index", but if your "index"
document is a custom template, you can also set this to another filename.
> Name of your master document (without suffix) [index]:

Sphinx can also add configuration for epub output:
> Do you want to use the epub builder (y/n) [n]:

Please indicate if you want to use one of the following Sphinx extensions:
> autodoc: automatically insert docstrings from modules (y/n) [n]: y
> doctest: automatically test code snippets in doctest blocks (y/n) [n]: n
> intersphinx: link between Sphinx documentation of different projects (y/n) [n]: y
> todo: write "todo" entries that can be shown or hidden on build (y/n) [n]: n
> coverage: checks for documentation coverage (y/n) [n]: y
> imgmath: include math, rendered as PNG or SVG images (y/n) [n]: y
> mathjax: include math, rendered in the browser by MathJax (y/n) [n]: n
> ifconfig: conditional inclusion of content based on config values (y/n) [n]: y
> viewcode: include links to the source code of documented Python objects (y/n) [n]: y
> githubpages: create .nojekyll file to publish the document on GitHub pages (y/n) [n]: n

A Makefile and a Windows command file can be generated for you so that you
only have to run e.g. `make html' instead of invoking sphinx-build
directly.
> Create Makefile? (y/n) [y]: y
> Create Windows command file? (y/n) [y]: n

Creating file ./source/conf.py.
Creating file ./source/index.rst.
Creating file ./Makefile.
```

3. Change `source/conf.py`. The first thing to do is to uncomment and change following lines at the top of the file.

```
import os
import sys
sys.path.insert(0, os.path.abspath('../../'))
```

A few extra extensions should also be added. The `autosummary` extension generates function/method/attribute summary lists from the docstrings, `napoleon` enables Sphinx to parse Numpy and Google style docstrings. Finally, the `numpydoc` extension loads several extensions for better support of Numpy.

```
extensions = ['sphinx.ext.autodoc',
    'sphinx.ext.intersphinx',
    'sphinx.ext.coverage',
    'sphinx.ext.imgmath',
    'sphinx.ext.ifconfig',
    'sphinx.ext.viewcode',
    'sphinx.ext.autosummery',
    'sphinx.ext.napoleon',
    'numpydoc']
```

And some more changes: - `master_doc = 'contents'` for a better overview page, which we will later add. - `html_style = 'sphinx_rtd_theme'` for a nicer theme. - For compatibility with Read the Docs:

```
html_theme_options = {
    #    'canonical_url': '',
    #    'analytics_id': '',
    'logo_only': False,
    'display_version': True,
    'prev_next_buttons_location': "top",
    #    'style_external_links': False,
    #    'vcs_pageview_mode': '',
    # Toc options
    'collapse_navigation': False,
    'sticky_navigation': True,
    'navigation_depth': 4,
    'includehidden': True,
    'titles_only': False,
}
```

- Comment out `# html_static_path = ['_static']`
- In case you use pictures hosted somewhere, add

```
suppress_warnings = [
    "image.nonlocal_uri",
    #    'app.add_directive',  # this evtl. suppresses the numpydoc induced warning
]
```

- And finally add some intersphinx mappings for links:

```
intersphinx_mapping = {
    "Python 3.7": ("https://docs.python.org/3.6", None),
    "Python": ("https://docs.python.org/", None),
    "NumPy": ("http://docs.scipy.org/doc/numpy/", None),
    "SciPy": ("http://docs.scipy.org/doc/scipy/reference", None),
    "matplotlib": ("http://matplotlib.org", None),
}
```

Now you are ready to create your docs. Have a look at the `docs/source/` folder for an example and how to use autosummary in the source files.

1.1.2 Publishing on Read the Docs

1. In the `docs/` folder create a file `requirements.txt` with a content like

```
numpy>=1.14.5
numpydoc
```

2. Log in or sign up on Read the Docs
3. Click on `Import a Project` and select your repository on GitHub and activate advanced options.
4. Select Python as the programming language.
5. Add `docs/requirements.txt` path to the `Requirements file` field.
6. Tick `Use system packages`.

Now, your documentation should be ready and hosted on Read the Docs. With every push to your repo, the documentation will automatically be built by Read the Docs.

In case you want to add a readthedocs badge, have a look at the first line after the heading of the *README.md* of this project.

1.2 Requirements

- Numpy >= 1.14.5

1.3 Indices and tables

- genindex
- modindex

1.4 License

GPL © 2019

CHAPTER 2

Python Deployment API

2.1 deployment

2.1.1 deployment.deploy