
pybtex-docutils Documentation

Release 1.0.4a2

Matthias C. M. Troffaes

Aug 23, 2023

Contents

1	Contents	3
1.1	Getting Started	3
1.2	API	5
1.3	Changes	5
1.4	License	7
2	Indices and tables	9
	Python Module Index	11
	Index	13

Release 1.0.4a2

Date Aug 23, 2023

1.1 Getting Started

1.1.1 Overview

The pybtex docutils backend allows BibTeX citations to be inserted into documentation generated by docutils.

- Download: <https://pypi.org/project/pybtex-docutils/#files>
- Documentation: <https://pybtex-docutils.readthedocs.io/en/latest/>
- Development: <http://github.com/mcmtroffaes/pybtex-docutils/>

1.1.2 Installation

For use with Sphinx, simply install `sphinxcontrib-bibtex`.

For use with pure docutils, install the module with `pip install pybtex_docutils`, or from source using `pip install -e ..`

1.1.3 Minimal Example

For use with Sphinx, refer to the `sphinxcontrib-bibtex` documentation.

For use with pure docutils, the module exposes a new `simplebibliography` directive, which will generate a citation for every entry in the specified bib files. This new directive is only intended for simple single document workflows that do not require the full power of Sphinx. You need exactly one of these directives in your document, placed at the location where you want the citations to appear (typically, at the end).

For example:

See [\[Nelson1987\]](#)_ for an introduction to non-standard analysis.

```
.. simplebibliography:: refs.bib
```

where `refs.bib` might contain:

Note that citation keys are used as labels. For this to work, it is thus necessary that all keys in your bib file are valid citation labels for docutils. In particular, they cannot contain colons. This limitation is lifted in `sphinxcontrib-bibtex`, which also provides many more citation features.

To use the directive, you have to write your own command script (there seems to be no other way currently to extend docutils). For instance:

```
#!/usr/bin/env python3
from docutils.parsers.rst import directives, Directive
from docutils.core import publish_cmdline, default_description

from pybtex_docutils import SimpleBibliography

description = ('Like rst2html5.py, but with .. simplebibliography support'
              + default_description)

if __name__ == '__main__':
    directives.register_directive("simplebibliography", SimpleBibliography)
    publish_cmdline(writer_name='html5', description=description)
```

You can then run this command as if you would run `rst2html5`.

You can also use the library to generate docutils nodes directly. For instance:

```
import io
import pybtex.database.input.bibtex
import pybtex.plugin

style = pybtex.plugin.find_plugin('pybtex.style.formatting', 'plain')()
backend = pybtex.plugin.find_plugin('pybtex.backends', 'docutils')()
parser = pybtex.database.input.bibtex.Parser()
data = parser.parse_stream(io.StringIO(u"""
@Book{1985:lindley,
  author = {D. Lindley},
  title = {Making Decisions},
  publisher = {Wiley},
  year = {1985},
  edition = {2nd},
}
"""))
for entry in style.format_entries(data.entries.values()):
    print(backend.paragraph(entry))
```

would produce:

```
<paragraph>
  D. Lindley. <emphasis>Making Decisions</emphasis>.
  Wiley, 2nd edition, 1985.
</paragraph>
```

1.2 API

The backend renders `pybtex.richtext.Text` instances into a list of `docutils.nodes.Node` instances. For typical use cases, all you need to care about are the methods `Backend.paragraph()`, `Backend.citation()`, and `Backend.citation_reference()` which are to be called on *formatted* entries.

Unless you are subclassing `Backend` to create a new backend, you should normally not import the `pybtex_docutils` module directly. Instead, use pybtex's plugin system to get the `Backend` class.

```
class pybtex_docutils.Backend (encoding=None)
    Bases: pybtex.backends.BaseBackend

    RenderType
        alias of builtins.list

    citation (entry: FormattedEntry, document: docutils.nodes.document, use_key_as_label=True) →
        docutils.nodes.citation
        Return citation node, with key as name, label as first child, and paragraph with entry text as second child.
        The citation is expected to be inserted into document prior to any docutils transforms.

    citation_reference (entry: FormattedEntry, document: docutils.nodes.document,
        use_key_as_label=True) → docutils.nodes.citation_reference
        Return citation_reference node to the given citation. The citation_reference is expected to be inserted into
        document prior to any docutils transforms.

    footnote (entry: FormattedEntry, document: docutils.nodes.document) → docutils.nodes.footnote
        Return footnote node, with key as name, and paragraph with entry text as child. The footnote is expected
        to be inserted into document prior to any docutils transforms.

        New in version 0.2.2.

    footnote_reference (entry: FormattedEntry, document: docutils.nodes.document) → docu-
        tils.nodes.footnote_reference
        Return footnote_reference node to the given citation. The footnote_reference is expected to be inserted
        into document prior to any docutils transforms.

        New in version 0.2.2.

    paragraph (entry: FormattedEntry) → docutils.nodes.paragraph
        Return a docutils.nodes.paragraph containing the rendered text for entry (without label).

        New in version 0.2.0.

class pybtex_docutils.SimpleBibliography (name, arguments, options, content, lineno, con-
        tent_offset, block_text, state, state_machine)
    Bases: docutils.parsers.rst.Directive
```

1.3 Changes

1.3.1 1.0.4 (in development)

1.3.2 1.0.3 (22 August 2023)

- Bump required docutils version to 0.14.
- Add tests for docutils 0.14 and 0.18 in github workflow.
- Fix tests for docutils 0.17 and lower (fixes issue #22, reported by kloczek).
- Drop Python 3.6 support.

1.3.3 1.0.2 (25 May 2022)

- New `simplebibliography` directive, which will generate a citation for every entry in the specified bib files, intended for simple single document workflows that do not require the full power of Sphinx.
- Fix deprecation warning with docutils 0.18 (see issue #21, reported by drammock).

1.3.4 1.0.1 (29 July 2021)

- Add `py.typed` so mypy finds the type annotations.
- Switch to github actions for regression testing.

1.3.5 1.0.0 (15 January 2021)

- Drop Python 2.7, 3.4, and 3.5 support.
- Add type annotations.
- Add support for sub and sup tags.

1.3.6 0.2.2 (9 October 2019)

- Drop Python 3.3 support.
- New `footnote` and `footnote_reference` methods for docutils footnote support.

1.3.7 0.2.1 (8 December 2014)

- Add Python 3.4 support, drop Python 3.2 support.
- Support more tags, also fail gracefully on unknown tags (see issue #6, reported by Jellyby).
- Use universal wheel for distribution.

1.3.8 0.2.0 (8 August 2013)

- **BACKWARD INCOMPATIBLE** The backend now renders into a list of docutils nodes instead of a single docutils node.
- New `paragraph()` method to render an entry into a single docutils paragraph.
- The `<inline>` wrapper nodes are no more, leading to much simpler generated code.
- Full test coverage.
- Generated citation nodes now contain text inside a paragraph.
- Minimal example.

1.3.9 0.1.0 (7 August 2013)

- Copied the backend from pybtex.
- Initial documentation.
- Initial tests and travis.ci integration.

1.4 License

pybtex-docutils is a docutils backend for pybtex
Copyright (c) 2013-2023 by Matthias C. M. Troffaes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 2

Indices and tables

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

p

`pybtex_docutils`, 4

B

Backend (*class in pybtex_docutils*), 5

C

citation() (*pybtex_docutils.Backend method*), 5

citation_reference() (*pybtex_docutils.Backend method*), 5

F

footnote() (*pybtex_docutils.Backend method*), 5

footnote_reference() (*pybtex_docutils.Backend method*), 5

P

paragraph() (*pybtex_docutils.Backend method*), 5

pybtex_docutils (*module*), 4

R

RenderType (*pybtex_docutils.Backend attribute*), 5

S

SimpleBibliography (*class in pybtex_docutils*), 5