
uTidylib Documentation

Release 0.4

Michal Čihař

Sep 30, 2019

Contents

| | | |
|----------|-------------------------------|-----------|
| 1 | Installing | 3 |
| 2 | Contributing | 5 |
| 3 | Running testsuite | 7 |
| 4 | Building documentation | 9 |
| 5 | License | 11 |
| 6 | Changes | 13 |
| 6.1 | 0.7 | 13 |
| 6.2 | 0.6 | 13 |
| 6.3 | 0.5 | 13 |
| 6.4 | 0.4 | 13 |
| 6.5 | 0.3 | 13 |
| 7 | Indices and tables | 15 |
| | Python Module Index | 17 |
| | Index | 19 |

The Tidy wrapper. I am the main interface to TidyLib. This package supports processing HTML with Tidy, with all the options that the tidy command line supports.

For more information on the tidy options, see the reference. These options can be given as keyword arguments to `parse` and `parseString`, by changing dashes (-) to underscores(_).

For example:

```
>>> import tidy
>>> from __future__ import print_function
>>> print(tidy.parseString(
...     '<Html>Hello Tidy!',
...     output_xhtml=1, add_xml_decl=1, indent=1, tidy_mark=0,
...     doctype='transitional'
... ))
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title></title>
  </head>
  <body>
    Hello Tidy!
  </body>
</html>
<BLANKLINE>
```

For options like `newline` and `output_encoding`, which must be set to one of a fixed number of choices, you can provide either the numeric or string version of the choice; so both `tidy.parseString('<HTML>foo</html>', newline=2)` and `tidy.parseString('<HTML>foo</html>', newline='CR')` do the same thing.

There are no plans to support other features of TidyLib, such as document-tree traversal, since Python has several quality DOM implementations. (The author uses Twisted's implementation, `twisted.web.microdom`).

`tidy.parse` (*self*, *filename*, ***kwargs*)

Parameters

- **kwargs** – named options to pass to TidyLib for processing the input file.
- **filename** – the name of a file to process

Returns a *Document* object

Open and process *filename* as an HTML file, returning a processed document object.

`tidy.parseString` (*self*, *text*, ***kwargs*)

Parameters

- **kwargs** – named options to pass to TidyLib for processing the input file.
- **text** – the string to parse

Returns a *Document* object

Use *text* as an HTML file, and process it, returning a document object.

class `tidy.Document` (*options*)

Document object as returned by `parseString()` or `parse()`.

errors

Returns list of errors as a list of *ReportItem*.

get_errors ()
Returns list of errors as a list of *ReportItem*.

gettext ()
Unicode text for output returned by tidy.

getvalue ()
Raw string as returned by tidy.

write (*stream*)
Parameters **stream** – Writable file like object.
Writes document to the stream.

class `tidy.ReportItem` (*err*)
Error report item as returned by tidy.
Attribute severity D, W, E or C indicating severity
Attribute line Line where error was fired (can be None)
Attribute col Column where error was fired (can be None)
Attribute message Error message itself
Attribute err Whole error message as returned by tidy

exception `tidy.TidyLibError`
Generic Tidy exception.

exception `tidy.InvalidOptionError`
Exception for invalid option.

exception `tidy.OptionArgError`
Exception for invalid parameter.

CHAPTER 1

Installing

To use uTidylib, you need to have HTML tidy library installed. Check <http://www.html-tidy.org/> for instructions how to obtain it.

CHAPTER 2

Contributing

You are welcome to contribute on GitHub, we use it for source code management, issue tracking and patches submission, see <<https://github.com/nijel/utidylib>>.

CHAPTER 3

Running testsuite

The testsuite can be executed using both `py.test` or `setuptools`, choose whatever approach you prefer:

```
./setup.py test  
py.test tidy
```


CHAPTER 4

Building documentation

To build the doc, just run:

```
make -C docs html
```

This requires that you have Sphinx installed.

The API documentation will be built in the `docs/_build/html/` directory.

The MIT License

Copyright (c) 2003 Cory Dodt <corydodt@twistedmatrix.com>

Copyright (c) 2014-2018 Michal Čihař <michal@cihar.com>

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.

6.1 0.7

- Not yet released.

6.2 0.6

- First official release PyPI.

6.3 0.5

- Fixed compatibility with Debian patched libtidy5deb1.

6.4 0.4

- Compatibility with html-tidy 5.6.0.
- Added support for Python 3.

6.5 0.3

- Initial release under new maintainer.
- Incorporated Debian patches.
- Various compatibility fixes (eg. with 64-bit machines).
- Various code cleanups.

- New test suite.
- New documentation.
- Support for new HTML 5 tidy library.

CHAPTER 7

Indices and tables

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

t

tidy, ??

D

Document (*class in tidy*), 1

E

errors (*tidy.Document attribute*), 1

G

get_errors() (*tidy.Document method*), 1

gettext() (*tidy.Document method*), 2

getvalue() (*tidy.Document method*), 2

I

InvalidOptionError, 2

O

OptionArgError, 2

P

parse() (*in module tidy*), 1

parseString() (*in module tidy*), 1

R

ReportItem (*class in tidy*), 2

T

tidy (*module*), 1

TidyLibError, 2

W

write() (*tidy.Document method*), 2