
Watson - Filters

Release 1.0.1

Jan 15, 2018

Contents

1	Build Status	3
2	Installation	5
3	Dependencies	7
4	Testing	9
5	Contributing	11
6	Table of Contents	13
6.1	Reference Library	13
	Python Module Index	15

Modify and convert values into something else.

CHAPTER 1

Build Status

build passing coverage 98%

CHAPTER 2

Installation

```
pip install watson-filters
```


CHAPTER 3

Dependencies

- watson-html

CHAPTER 4

Testing

Watson can be tested with `pytest`. Simply activate your virtualenv and run `python setup.py test`.

CHAPTER 5

Contributing

If you would like to contribute to Watson, please feel free to issue a pull request via Github with the associated tests for your code. Your name will be added to the AUTHORS file under contributors.

6.1 Reference Library

6.1.1 `watson.filters.abc`

class `watson.filters.abc.Filter`

All filters must extend the `BaseFilter` or be callable.

6.1.2 `watson.filters.string`

class `watson.filters.string.Date` (*format*='%Y-%m-%d')

Formats a string representation of a date to a particular format.

Example:

```
filter = Date()
filter('2013-09-12') # datetime object for that date
```

__init__ (*format*='%Y-%m-%d')

Initializes the filter.

Parameters `format` (`string`) – The date format to make.

class `watson.filters.string.HtmlEntities`

Encodes all html entities.

Example:

```
filter = HtmlEntities()
filter('<div>test</div>') # &lt;div&gt;test&lt;/div&gt;
```

class `watson.filters.string.Lower`

Converts all characters to lowercase.

Example:

```
filter = Lower()
filter('ABCD') # abcd
```

class `watson.filters.string.Numbers` (*regex='[^0-9]', replacement='', flags=0*)
Strips all characters except for numbers.

Example:

```
filter = Numbers()
filter('abcd1234') # 1234
```

`__init__` (*regex='[^0-9]', replacement='', flags=0*)

class `watson.filters.string.RegEx` (*regex, replacement='', flags=0*)
Uses regular expressions to replace values.

Example:

```
filter = RegEx('ing', replacement='ed')
filter('Stopping') # Stopped
```

`__init__` (*regex, replacement='', flags=0*)
Initializes the filter.

Parameters

- **regex** (*string/regex*) – The pattern to match.
- **replacement** (*string*) – The value to be used in the replacement.
- **flags** (*int*) – The regex flags.

class `watson.filters.string.StripTags` (*regex='</?\s(?:[^\s]*=(\s*"[""]*"|\s*'['']*|\s*\{[\s\}]*|\s*\[[\s\]]*\)|\s*>)*?>', flags=2*)

Strips all html tags.

Thanks to django for the regex used below.

Example:

```
filter = StripTags()
filter('test<div>blah</div>') # testblah
```

`__init__` (*regex='</?\s(?:[^\s]*=(\s*"[""]*"|\s*'['']*|\s*\{[\s\}]*|\s*\[[\s\]]*\)|\s*>)*?>', flags=2*)

class `watson.filters.string.Trim`
Strips whitespace from value.

class `watson.filters.string.Upper`
Converts all characters to uppercase.

Example:

```
filter = Upper()
filter('abcd') # ABCD
```

W

`watson.filters.abc`, 13

`watson.filters.string`, 13

Symbols

`__init__()` (watson.filters.string.Date method), 13
`__init__()` (watson.filters.string.Numbers method), 14
`__init__()` (watson.filters.string.RegEx method), 14
`__init__()` (watson.filters.string.StripTags method), 14

D

Date (class in watson.filters.string), 13

F

Filter (class in watson.filters.abc), 13

H

HtmlEntities (class in watson.filters.string), 13

L

Lower (class in watson.filters.string), 13

N

Numbers (class in watson.filters.string), 14

R

RegEx (class in watson.filters.string), 14

S

StripTags (class in watson.filters.string), 14

T

Trim (class in watson.filters.string), 14

U

Upper (class in watson.filters.string), 14

W

watson.filters.abc (module), 13
watson.filters.string (module), 13