
Watson - Validators

Release 1.0.0

September 30, 2014

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

Validate and verify arbitrary values.

Build Status

Installation

```
pip install watson-validators
```

Testing

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

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.

Table of Contents

5.1 Reference Library

5.1.1 watson.validators.numeric

class `watson.validators.numeric.Range` (*min=None, max=None, message=""{value}" is not between {min} and {max}'*)

Validates the length of a string.

Example:

```
validator = Length(1, 10)
validator('Test') # True
validator('Testing maximum') # raises ValueError
```

`__init__` (*min=None, max=None, message=""{value}" is not between {min} and {max}'*)

5.1.2 watson.validators.string

class `watson.validators.string.Csrf` (*token=None, message='Cross-Site request forgery attempt detected, invalid token specified "{token}"*)

Validates a csrf token.

Example:

```
validator = Csrf()
validator('submitted token')
```

`__init__` (*token=None, message='Cross-Site request forgery attempt detected, invalid token specified "{token}"*)

class `watson.validators.string.Length` (*min=-1, max=-1, message=""{value}" does not meet the required length'*)

Validates the length of a string.

Example:

```
validator = Length(1, 10)
validator('Test') # True
validator('Testing maximum') # raises ValueError
```

`__init__` (*min=-1, max=-1, message=""{value}" does not meet the required length'*)

Initializes the validator.

Min, max, length are interpolated into the message.

Parameters

- **min** (*int*) – The minimum length of the string.
- **max** (*int*) – The maximum length of the string.
- **message** (*string*) – The message to be used if the validator fails.

class `watson.validators.string.RegEx` (*regex, flags=0, message="{value} does not match pattern "{pattern}"*)

Validates a value based on a regular expression.

Example:

```
validator = RegEx('Match')
validator('Match') # True
validator('Other') # raises ValueError
```

`__init__` (*regex, flags=0, message="{value} does not match pattern "{pattern}"*)

class `watson.validators.string.Required` (*message='Value is required'*)

Validates whether or not a value exists.

Example:

```
validator = Required()
validator('Test') # True
validator('') # raises ValueError
```

`__init__` (*message='Value is required'*)

W

`watson.validators.numeric`, [11](#)
`watson.validators.string`, [11](#)