
marshmallow-validators

Release 2.0.0

December 29, 2015

1	Get it now	3
2	WTForms support	5
3	colander support	7
4	Base converter class	9
5	Project Info	11
5.1	Changelog	11
5.2	License	11
	Python Module Index	13

Release v2.0.0 (*Changelog*)

Use 3rd-party validators (e.g. from WTForms and colander) with marshmallow.

```
from marshmallow import Schema, fields
from marshmallow_validators.wtforms import from_wtforms
from wtforms.validators import Email, Length

# Leverage WTForms i18n
locales = ['de_DE', 'de']

class UserSchema(Schema):
    email = fields.Str(
        validate=from_wtforms([Email()], locales=locales)
    )
    password = fields.Str(
        validate=from_wtforms([Length(min=8, max=300)], locales=locales)
    )

UserSchema().validate({'email': 'invalid', 'password': 'abc'})
# {'email': ['Ungültige Email-Adresse.'],
#  'password': ['Feld muss zwischen 8 und 300 Zeichen beinhalten.']}
```

Get it now

```
pip install -U marshmallow-validators
```

WTForms support

class `marshmallow_validators.wtforms.from_wtforms` (*validators*, *locales=None*)
Convert a WTForms validator from `wtforms.validators` to a marshmallow validator.

Example:

```
from marshmallow import fields
from marshmallow_validators.wtforms import from_wtforms
from wtforms.validators import Length

password = fields.Str(
    validate=from_wtforms([Length(min=8, max=100)])
)
```

Parameters

- **validators** (*list*) – WTForms validators.
- **locales** (*list*) – List of locales for error messages, in order. If `None`, will try to use locale information from the environment.

`marshmallow_validators.wtforms.make_converter` (*locales=None*)
Return a WTForms validator converter that uses the given locales.

Example:

```
from marshmallow import fields
from marshmallow_validators.wtforms import make_converter
from wtforms.validators import Length

from_wtf = make_converter(['de_DE', 'de'])

field = fields.Str(from_wtf([Length(max=3)]))
```

Parameters **locales** (*list*) – List of locales for error messages.

colander support

class `marshmallow_validators.colander.from_colander` (*validators*)
Convert a colander validator to a marshmallow validator.

Example:

```
from marshmallow import fields
from marshmallow_validators.colander import from_colander
from colander import Length

password = fields.Str(
    validate=from_colander([Length(min=8, max=100)])
)
```

Parameters `validators` (*list*) – Colander validators.

Base converter class

class `marshmallow_validators.core.BaseConverter` (*validators*)

Base converter validator that converts a third-party validators into marshmallow validators. Concrete classes must implement `make_validator`.

Parameters `validators` (*list*) – List of 3rd-party validator objects to convert.

make_validator (*validator*)

Receives a 3rd-party validator and converts it to a marshmallow validator function/callable.

Parameters `validator` – A 3rd-party validator object

Returns A callable marshmallow validator

Project Info

5.1 Changelog

5.1.1 2.0.0 (2015-12-28)

- Fix compatibility with webargs \geq 0.16.0.
- Remove redundant implementation of `ValidationError`. `marshmallow-validators` will import the `ValidationError` from `webargs` if it is available, falling back on `marshmallow.ValidationError`.

5.1.2 1.0.0 (2015-08-30)

- First release.
- Supports conversion of WTForms and colander validators.
- `ValidationError` that inherits from both `webargs' ValidationError` (if `webargs` is installed) and `marshmallow's ValidationError`.

5.2 License

Copyright 2015 Steven Loria and contributors

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.

m

`marshmallow_validators.colander`, [7](#)
`marshmallow_validators.wtforms`, [5](#)

B

`BaseConverter` (class in `marshmallow_validators.core`), 9

F

`from_colander` (class in `marshmallow_validators.colander`), 7

`from_wtforms` (class in `marshmallow_validators.wtforms`), 5

M

`make_converter()` (in module `marshmallow_validators.wtforms`), 5

`make_validator()` (`marshmallow_validators.core.BaseConverter` method), 9

`marshmallow_validators.colander` (module), 7

`marshmallow_validators.wtforms` (module), 5