
django_choices_flow Documentation

Release 0.8.0.2

Valder Gallo

Sep 27, 2017

Contents

1 Overview	1
1.1 Motivation	1
2 Usage	3
2.1 Model example	3
2.2 Set Error Message	4
2.3 Shell example	4
3 How to Contribute	5
3.1 Issues and bugs	5
3.2 Create my SandBox	5
3.3 Support	6
4 Class description	7
4.1 Base	7
4.2 Models	7
5 Indices and tables	9
Python Module Index	11

Django Workflow using Choices

Motivation

We need one easy way to validate a lot of choices flow with Django. Sort of one status machine that can validate status for different Models in different APPs with different Flow.

Some workflow programs work with XML to create flow. XML is one another easy way, but I don't like the idea to create and read XMLs. We need something with more performance, we have a lot of users.

Model example

```
1 from django.db import models
2 from django_choices_flow import Choices
3 from django_choices_flow.models import FlowIntegerField
4
5
6 class MyChoices(Choices):
7     NEW = 1, 'New content' # 'New content' is the display text
8     WAIT = 2, 'Wait'
9     CANCELED = -1, 'Canceled'
10    ERROR = -2, 'Error'
11    INVOICED = 3, 'Invoiced'
12
13    # set transaction rules
14    NEW_RULES = [NEW, INVOICED, CANCELED, ERROR]
15    WAIT_RULES = [CANCELED, ERROR, INVOICED]
16    INVOICED_RULES = [CANCELED]
17
18
19 class Invoices(models.Model):
20     """
21     To use only choices
22     """
23     number = models.IntegerField()
24     status = models.IntegerField(choices=MyChoices, default=MyChoices.NEW)
25
26     def __unicode__(self):
27         return self.number
28
29
30 class FlowInvoice(models.Model):
31     """
```

```

32     To validate flow in choices
33     """
34     number = models.IntegerField()
35     status = FlowIntegerField(choices=MyChoices, default=MyChoices.NEW)
36
37     def __unicode__(self):
38         return self.number

```

Set Error Message

CHANGE ALL MESSAGE

To change error message for all ChoiceFlow set nem messagem on **Django.settings**

CHANGE ONLY ONE MESSAGE

To change error message only in one ChoiceFlow, set **error_msg** on ChoicesFlow

```

1  class MyChoices(Choices):
2      NEW = 1, 'New content' # 'New content' is the display text
3      WAIT = 2, 'Wait'
4      CANCELED = -1, 'Canceled'
5      ERROR = -2, 'Error'
6      INVOICED = 3, 'Invoiced'
7
8      # set transaction rules
9      NEW_RULES = [NEW, INVOICED, CANCELED, ERROR]
10     WAIT_RULES = [CANCELED, ERROR, INVOICED]
11     INVOICED_RULES = [CANCELED]
12
13     error_msg = "My Custom Error Message for this ChoicesFlow"

```

Shell example

```

1  >>> flow = FlowInvoice.objects.create(number=1234)
2  >>> flow.status
3  1
4  >>> flow.status = MyChoices.INVOICED
5  >>> flow.full_clean()
6  >>> flow.save()
7  >>> flow.status
8  3
9  >>> flow.status = MyChoices.WAIT
10 >>> flow.full_clean()
11 ValidationError: {'status': [u'Invalid choice: Wait']}

```


Issues and bugs

To create issues or report bugs. Please use Github :D

<https://github.com/valdergallo/django-choices-flow/issues>

Create my SandBox

```
1 # download code
2 git clone git@github.com:valdergallo/django-choices-flow.git
3
4 # install developer packages
5 make
6
7 # check coverage
8 make coverage
9
10 # test project
11 make test
12
13 #clean extra content
14 make clean
15
16 #send package
17 make send_package
18
19 #test py2 and py3
20 tox
```

Support

License type FREEBSD

South Support migrations

Python python 2.7 and python 3.3

Version 0.9.0

Tested in Django 1.2.7; 1.3.7; 1.4.5; 1.5.1; 1.6.2

Base

class `django_choices_flow.base.MetaChoice` (**args, **kwargs*)

- Convert attributes to Tuples
- Add interator in class attributes

error_msg

get default error message

validate (*status, new_status*)

Validate workflow

Models

class `django_choices_flow.models.FlowCharField` (**args, **kwargs*)

Custom CharField with workflow validation

static get_db_value (*model_instance*)

Get database value

validate (*value, model_instance*)

Validate choice workflow

```
class django_choices_flow.models.FlowIntegerField (verbose_name=None,  
                                                name=None, primary_key=False,  
                                                max_length=None,  
                                                unique=False, blank=False,  
                                                null=False, db_index=False,  
                                                rel=None, default=<class  
                                                django.db.models.fields.NOT_PROVIDED>,  
                                                editable=True, serialize=True,  
                                                unique_for_date=None,  
                                                unique_for_month=None,  
                                                unique_for_year=None,  
                                                choices=None, help_text=u'',  
                                                db_column=None,  
                                                db_tablespace=None,  
                                                auto_created=False, validators=[],  
                                                error_messages=None)
```

Custom IntegerField with workflow validation

```
static get_db_value (model_instance)  
    Get database value
```

```
validate (value, model_instance)  
    Validate choice workflow
```

CHAPTER 5

Indices and tables

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

d

`django_choices_flow`, 7
`django_choices_flow.base`, 7
`django_choices_flow.models`, 7

D

django_choices_flow (module), 7
django_choices_flow.base (module), 7
django_choices_flow.models (module), 7

E

error_msg (django_choices_flow.base.MetaChoice attribute), 7

F

FlowCharField (class in django_choices_flow.models), 7
FlowIntegerField (class in django_choices_flow.models), 7

G

get_db_value() (django_choices_flow.models.FlowCharField static method), 7
get_db_value() (django_choices_flow.models.FlowIntegerField static method), 8

M

MetaChoice (class in django_choices_flow.base), 7

V

validate() (django_choices_flow.base.MetaChoice method), 7
validate() (django_choices_flow.models.FlowCharField method), 7
validate() (django_choices_flow.models.FlowIntegerField method), 8