django_choices_flow Documentation

Release 0.8.0.2

Valder Gallo

Sep 27, 2017

Contents

CHAPTER 1

Overview

Django Workflow using Choices

Motivation

We need one easy way to validate alot choices flow with Django. Sort of one status machine that can validate status for diferentes Models in diferentes APPs with diferents Flow.

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

CHAPTER 2

Usage

Model example

```python
from django.db import models
from django_choices_flow import Choices
from django_choices_flow.models import FlowIntegerField


class MyChoices(Choices):
    NEW = 1, 'New content' # 'New content' is the display text
    WAIT = 2, 'Wait'
    CANCELED = -1, 'Canceled'
    ERROR = -2, 'Error'
    INVOICED = 3, 'Invoiced'

    # set transaction rules
    NEW_RULES = [NEW, INVOICED, CANCELED, ERROR]
    WAIT_RULES = [CANCELED, ERROR, INVOICED]
    INVOICED_RULES = [CANCELED]


class Invoces(models.Model):
    """
    To use only choices
    """
    number = models.IntegerField()
    status = models.IntegerField(choices=MyChoices, default=MyChoices.NEW)

    def __unicode__(self):
        return self.number


class FlowInvoice(models.Model):
    """
    To validate flow in choices
    """
    number = models.IntegerField()
    status = FlowIntegerField(choices=MyChoices, default=MyChoices.NEW)

    def __unicode__(self):
        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

```
class MyChoices(Choices):
    NEW = 1, 'New content' # 'New content' is the display text
    WAIT = 2, 'Wait'
    CANCELED = -1, 'Canceled'
    ERROR = -2, 'Error'
    INVOICED = 3, 'Invoiced'

    # set transaction rules
    NEW_RULES = [NEW, INVOICED, CANCELED, ERROR]
    WAIT_RULES = [CANCELED, ERROR, INVOICED]
    INVOICED_RULES = [CANCELED]

    error_msg = "My Custom Error Message for this ChoicesFlow"
```

Shell example

```
>>> flow = FlowInvoice.objects.create(number=1234)
>>> flow.status
1
>>> flow.status = MyChoices.INVOICED
>>> flow.full_clean()
>>> flow.save()
>>> flow.status
3
>>> flow.status = MyChoices.WAIT
>>> flow.full_clean()
ValidationError: {'status': [u'Invalid choice: Wait']}
```

CHAPTER 3

How to Contribute

Issues and bugs

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

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

Create my SandBox

```
# download code
git clone git@github.com:valdergallo/django-choices-flow.git

# install developer packages
make

# check coverage
make coverage

# test project
make test

#clean extra content
make clean

#send package
make send_package

#test py2 and py3
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

CHAPTER 4

Class description

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

Python Module Index

Index