django-ulogin Documentation

Release 0.3.0

Mikhail Porokhovnichenko

Dec 07, 2021

Contents

1 Synopsis **1**
1.1 The ct helper . . . 1

2 Installation **3**
2.1 With our lovely pip . . . 3
2.2 With pipenv . . . 3
2.3 Or poetry . . . 3

3 Usage **5**
3.1 An old way . . . 5
3.2 A new way . . . 5
3.3 Managers and querysets . . . 6

4 1.2 **7**

5 1.1.1 **9**

6 1.1 **11**

7 1.0.5 **13**

8 1.0.4 **15**

9 1.0.3 **17**

10 1.0.2 **19**

11 1.0.1 **21**

12 1.0.0 **23**

13 0.3.7 **25**

14 0.3.6 **27**

15 0.3.5 **29**

16 0.3.4 **31**

17 0.3.3 33

18 0.3.2 35

19 Authors 37

CHAPTER 1

Synopsis

The django application aimed to simplify work with generic relation and brings some syntax sugar to it.

The first purpose the library was written is ability to make queryset filtering through generic relation fields like it was normal ones.

```
Comment.objects.filter(content_type=blog_post)
```

1.1 The ct helper

One of the most useful features of the package is a small function called `ct()`. It just return a

CHAPTER 2

Installation

2.1 With our lovely pip

A stable version is available on Python Package Index, thus you can install it within `pip` or `easy_install` (is somebody uses this one?)

```
$ pip install django-generic-helpers
```

A development version can be grabbed directly from the github repository:

```
$ pip install -e git+https://github.com/marazmiki/django-generic-helpers#egg=django-
↪generic-helpers
```

2.2 With pipenv

Let's not discuss the pros and cons of this one, it is just used by lots of p eople. So, to add the dependency into your `Pipfile` and then lock it out, type in your console:

```
$ pipenv install django-generic-helpers
```

2.3 Or poetry

Yet another package management tool:

```
$ poetry add django-generic-helpers
```

CHAPTER 3

Usage

3.1 An old way

The old way we've added generic fields to an orbitrary model was an inheritance from the base abstract `GenericRelationModel` model.

```
from django.db import models
from generic_helpers.models import GenericRelationModel

class Vote(GenericRelationModel):
    pass
```

or from `BlankGenericRelationModel` to get an optional field:

```
from generic_helpers.models import BlankGenericRelationModel

class Vote(BlankGenericRelationModel):
    pass
```

Well, it wasn't quite usable, because a model could have the only generic relation field with a fixed name.

```
from generic_helpers.decorators import generic_relation

@generic_relation('content_object', ct_field='content_type', pk_field='o_id')
@generic_relation('content_object', ct_field='content_type', fk_field='o_id')
class Vote(models.Model):
    pass
```

3.2 A new way

With a new one, we just declare a generic relation field like as usual model one.

```
from django.db import models
from generic_helpers.fields import GenericRelationField

class Vote(models.Model)
    content_object = GenericRelationField(
        replace_manager=False,                  # 1
        manager_name='gr',                      # 2
        ct_field='content_type',                # 3
        fk_field='object_pk',                   # 4
        fk_field_type=models.IntegerField(),    # 5
        allowed_content_types=[],                 # 6
        denied_content_types=[],                  # 7

        # ...and other keyword arguments that
        # the ForeignKey field accepts
    )
```

With this, you can create an arbitrary number of generic relation fields with different options and customized managers.

```
Vote.objects.filter(content_object=my_post)
```

2)

3.3 Managers and querysets

post = Post.objects.get(pk=1)

Vote.objects.get_for_object(content_object=post)

Filtering allowed here:

Vote.objects.get_for_object(content_object=post).filter(content_object__user=None) # Is it possible?

CHAPTER 4

1.2

- Added support for `Django 4.x`

CHAPTER 5

1.1.1

- A sad fix: forgot to declare Python 3.10 support in `pyproject.toml`
- Also, added a new test for manager, that revealed a bug with queryset on `Django 3.2+`
- Does not use Python 2.x syntax for `super()` anymore

CHAPTER 6

1.1

- Dropped all the environments excced end of support: Python 3.5 and older, Django 2.1 and lower;
- Get rid of some legacy;
- Switched on poetry instead of pipenv.
- Added support for Django 3.2

CHAPTER 7

1.0.5

- Fixed a bug with customized `save()` method on a model class. Thank @volody2006 for reporting
- Moved all the regression tests into a separated file

CHAPTER 8

1.0.4

- Fixed a bug with `allowed_content_types` when referencing to a model whose name is the beginning of another one (reported by @busy)
- Added `bump2version`

CHAPTER 9

1.0.3

- Added `Python 3.8` and `Django 3.0` support

CHAPTER 10

1.0.2

- A bugfix: prevented `makemigrations` from generating infinite migrations because of `allowed_content_types` everytime is a new object
- Added `Django==2.2` support
- Updated `README` a bit: replaced an old text to the actual one; made badges SVG.
-

CHAPTER 11

1.0.1

- A bugfix: don't change a customized manager with default django's one

CHAPTER 12

1.0.0

- A new declarative way to add generic relation fields to your model: any number or generic relation, any field name(s);
- Ability to add a generic relation for a restricted set of models. You can either enumerate all allowed models or all denied ones;
- An alternative way to add these fields: with a decorator (actually, it's not a new feature, just saved from unpublished release);
- The `ct` helper now caches own results;
- Added support for Python 3.6;
- Added support for Python 3.7;
- Added Django 2.0 and 2.1 support;
- Removed support for Python 3.3 and older;
- Removed support for Django 1.10 and older

CHAPTER 13

0.3.7

- Update django head version

CHAPTER 14

0.3.6

- Update django head version

CHAPTER 15

0.3.5

- Add CHANGELOG.rst into manifest

CHAPTER 16

0.3.4

- Remove deprecation warning

CHAPTER 17

0.3.3

- Moved the **ct** shortcut into **utils** module

CHAPTER 18

0.3.2

- Added CHANGELOG :)
- Added Python 3.4x support;
- Dropped Python 3.2x support;
- Improved code styling with pep8;
- Wheel available;
- Coverage support;
- Updated Django head version;
- Fix 0.3.1 install bug (import six from django).

CHAPTER 19

Authors

- Mikhail Porokhovnichenko <marazmiki@gmail.com>