
Django lookup extensions Documentation

Release 0.2.1

UNCOVER TRUTH Inc.

Jan 29, 2019

Contents:

1	Django lookup extensions	1
1.1	Installation	1
1.2	Usage	1
1.3	Supported lookup types	2
1.4	Supported vendor types	3
1.5	Credits	3
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
2.3	Prerequisites	5
3	Usage	7
3.1	Set extended manager to your models	7
3.2	Exists and Not exists	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.2.1	15
6.2	0.2.0	15
6.3	0.1.0	15
6.4	0.0.1 - 0.0.2	15
7	Indices and tables	17

Django lookup extensions

Django lookup extensions use **NOT** query.

- Free software: MIT license
- Documentation: <https://django-lookup-extensions.readthedocs.io>.

1.1 Installation

To install Django lookup extensions, run this command in your terminal:

```
$ pip install django-lookup-extensions
```

This is the preferred method to install Django lookup extensions, as it will always install the most recent stable release. If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

1.2 Usage

1.2.1 Setup

Add `'lookup_extensions'` to your `INSTALLED_APPS` setting.:

```
INSTALLED_APPS = [  
    ...  
    'lookup_extensions',  
    ...  
]
```

`AppConfig.ready` imports lookups and register them using `django.db.models.fields.Field.register_lookup`.

1.2.2 Making queries

Use lookup like Django standard lookups.

```
ModelA.objects.filter(name__neexact='test name')
```

1.3 Supported lookup types

1.3.1 Negaetes

neexact is negate exact, neexact is negate iexact, others are similar.

- neexact
- neexact
- necontains
- neicontains
- nestartswith
- neendswith
- neistartswith
- neiendswith
- neregex
- neiregex

1.3.2 Complement

- complement

1.3.3 Extra regexes

These regex lookups support metacharacters `\d`, `\D`, `w`, `\W`, `\s`, `\S`.

MySQL, PostgreSQL and Redshift also support `\<`, `\>`.

Prefix follows the previous section.

- exregex
- exiregex
- neexregex
- neexiregex

1.4 Supported vendor types

- MySQL
- PostgreSQL
- sqlite
 - lookups using *LIKE* aren't case sensitive.
 - * necontains
 - * nestartswith
 - * neendswith
- Redshift

1.5 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install Django lookup extensions, run this command in your terminal:

```
$ pip install django-lookup-extensions
```

This is the preferred method to install Django lookup extensions, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for Django lookup extensions can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/uncovertruth/django-lookup-extensions
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/uncovertruth/django-lookup-extensions/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

2.3 Prerequisites

Add `lookup_extensions` to your `INSTALLED_APPS` setting:

```
INSTALLED_APPS = [  
    'lookup_extensions',  
    # ...  
]
```

Change DATABASE 's ENGINE setting:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'lookup_extensions.backends.postgresql',  
        # ...  
    },  
}
```

- For PostgreSQL/Redshift: `lookup_extensions.backends.postgresql`
- For MySQL: `lookup_extensions.backends.mysql`
- SQLite: `lookup_extensions.backends.sqlite3`

3.1 Set extended manager to your models

```
from lookup_extensions.manager import Manager

class Article(models.Model):
    # ...

    objects = Manager()
```

3.2 Exists and Not exists

```
from django.db.models import Exists, OuterRef

tags = Tag.objects.filter(articles=OuterRef('id'), name='Tag 2')
Article.objects.filter(tag__exists=Exists(tags)).filter(author=self.aul)
```


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/uncovertruth/django-lookup-extensions/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

4.1.4 Write Documentation

Django lookup extensions could always use more documentation, whether as part of the official Django lookup extensions docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/uncovertruth/django-lookup-extensions/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *django-lookup-extensions* for local development.

1. Fork the *django-lookup-extensions* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-lookup-extensions.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv django-lookup-extensions
$ cd lookup_extensions/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 lookup_extensions tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/uncovertruth/django-lookup-extensions/pull_requests and make sure that the tests pass for all supported Python versions.

5.1 Development Lead

- UNCOVER TRUTH Inc. <develop@uncovertruth.co.jp>

5.2 Contributors

None yet. Why not be the first?

6.1 0.2.1

- Support related *complement* lookups

6.2 0.2.0

- Add support for Django 2.1
- Add support for Python 3.7
- Migrate to *complement* from *exists* and *neexists*

6.3 0.1.0

- Migrate installation to backend settings
- Support *exists* and *neexists* lookups

6.4 0.0.1 - 0.0.2

- Support extra regex lookups
- Support negaete lookups

CHAPTER 7

Indices and tables

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