
drf-querystringfilter Documentation

Release 1.0.0

Stefano Apostolico

Jul 09, 2018

Contents

1	drf-querystringfilter	3
1.1	Documentation	3
1.2	Basic Usage	3
2	Installation	5
3	Usage	7
3.1	Filtering	7
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	13
4.4	Tips	13
5	Credits	15
5.1	Development Lead	15
5.2	Contributors	15
6	1.0	17
7	0.7.0	19
8	0.6.0	21
9	0.5.0 18/06/2018	23
10	0.4.0 29/05/2017	25
11	0.3.0 10/10/16	27
12	0.2.0 19/09/16	29
13	0.1.0 11/09/16	31

Contents:

CHAPTER 1

drf-querystringfilter

pypi package **1.0.0** Filter backend for DjangoRestFramework able to parse url parameters

Supports drf 3.5.x, 3.6.x, 3.7.x, 3.8.x Django 1.10.x, 1.11.x, 2.0.x, python 2.7, 3.6

1.1 Documentation

The full documentation is at <https://drf-querystringfilter.readthedocs.org>.

1.2 Basic Usage

```
class UserSerializer(ModelSerializer):
    class Meta:
        model = User
        exclude = ()

class Users(ListAPIView):
    serializer_class = UserSerializer
    filter_fields = ['username', 'email', 'is_staff', 'date_joined']
    filter_blacklist = None
    filter_backends = (QueryStringFilterBackend,)
    queryset = User.objects.all()
```

now you can query using...

```
- /users/?username=sax
- /users/?username__startswith=sa&date_joined__year=2000
- /users/?email__contains=@gmail.com
- /users/?is_staff=true
```

1.2.1 Links

Stable			
Development			
Project home page:	https://github.com/saxix/drf-querystringfilter		
Issue tracker:	https://github.com/saxix/drf-querystringfilter/issues?sort		
Download:	http://pypi.python.org/pypi/drf-querystringfilter/		
Documentation:	https://drf-querystringfilter.readthedocs.org/en/latest/		

CHAPTER 2

Installation

At the command line:

```
$ easy_install drf-querystringfilter
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv drf-querystringfilter  
$ pip install drf-querystringfilter
```


To use drf-querystringfilter in a project:

```
import drf_querystringfilter
```

Configure your view to use it:

```
class DemoModelView(ListAPIView):
    filter_backends = (QueryStringFilterBackend,)
    filter_fields = ['username', 'email', 'is_staff', 'date_joined']
    filter_blacklist = ['.*__'] # disable any join
```

3.1 Filtering

3.1.1 exact/iexact

```
?username=admin
```

3.1.2 contains

```
?email__contains=@gmail
objects.filter(email__contains="@gmail")
```

3.1.3 gt/gte

```
?int__gt=5  
  
objects.filter(int__gt=5)
```

3.1.4 lt/lte

```
?int__lt=5  
  
objects.filter(int__lt=5)
```

3.1.5 is

```
?flag__is=1  
?flag__is=true  
  
objects.filter(flag=True)
```

or

```
?flag__is=0  
?flag__is=false  
  
objects.filter(flag=False)
```

3.1.6 isnull

```
?flag__isnull=true  
?flag__isnull=false  
  
objects.filter(flag=True)
```

3.1.7 not

```
?name__not=abc  
  
objects.exclude(name="abc")
```

3.1.8 inlist

```
?id__in=1,2,4  
  
objects.filter(id__in=[1,2,3])
```

3.1.9 not_inlist

```
?id__not_in=1,2,4  
  
objects.exclude(id__in=[1,2,3])
```

3.1.10 in

```
?id__in=1&in__in=2&id__in=3  
  
objects.filter(id__in=[1,2,3])
```

3.1.11 not_in

```
?id__not_in=1&in__not_in=2&id__not_in=3  
  
objects.exclude(id__in=[1,2,3])
```

3.1.12 inarray

```
?json__array__inarray=a  
  
objects.filter(json__array__contains=["a"])
```

3.1.13 int_inarray

```
?json__array__int_inarray=1  
  
objects.filter(json__array__contains=[1])
```


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/saxix/drf-querystringfilter/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” is open to whoever wants to implement it.

4.1.3 Implement Features

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

4.1.4 Write Documentation

drf-querystringfilter could always use more documentation, whether as part of the official drf-querystringfilter 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/saxix/drf-querystringfilter/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 *drf-querystringfilter* for local development.

1. Fork the *drf-querystringfilter* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/drf-querystringfilter.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 drf-querystringfilter
$ cd drf-querystringfilter/
$ make 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:

```
$ make qa
$ py.test tests
$ 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, and 3.5, and for PyPy. Check https://travis-ci.org/saxix/drf-querystringfilter/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests/<test_filename>::<test_name>>
```


5.1 Development Lead

- Stefano Apostolico <s.apostolico@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

1.0

- First stable release

CHAPTER 7

0.7.0

- abstract query_params handling
- handle multiple values in query string
- BACKWARD INCOMPATIBLE: `__in` now accept raw values and can appear multiple times
- new operators `__inlist` and `__not_inlist` to be used for backward compatibility with `__in` and `__not_in`

CHAPTER 8

0.6.0

- Add handling of format query param

CHAPTER 9

0.5.0 18/06/2018

- add support for django 2.0
- add *query_params* property to allow handling POST request

CHAPTER 10

0.4.0 29/05/2017

- add ‘__inarray’ and ‘__int_inarray’ lookup to handle json/arrays lookup both str/int

CHAPTER 11

0.3.0 10/10/16

- add ‘_distinct’ parameter to enable ‘.distinct()’ queries

CHAPTER 12

0.2.0 19/09/16

- add 'ignore_filter' to ignore querystring arguments

CHAPTER 13

0.1.0 11/09/16

- First release on PyPI.