
drf-sideloading Documentation

Release 0.1.10

Demur Nodia

Oct 29, 2018

Contents

1	drf-sideload	3
1.1	Documentation	3
1.2	Quickstart	3
1.3	Example Project	5
1.4	Contributing	5
1.5	Credits	5
2	Installation	7
3	Usage	9
4	drf_sideload	11
4.1	drf_sideload package	11
5	Contributing	13
5.1	Types of Contributions	13
5.2	Get Started!	14
5.3	Pull Request Guidelines	15
5.4	Tips	15
6	Credits	17
6.1	Contributors	17
7	History	19
7.1	0.1.10 (2017-07-20)	19
7.2	0.1.8 (2017-07-20)	19
7.3	0.1.0 (2017-07-20)	19
	Python Module Index	21

Contents:

Extension for Django Rest Framework to enable simple sideloading

1.1 Documentation

The full documentation is at <https://drf-sideload.readthedocs.io>.

1.2 Quickstart

Install drf-sideload:

```
pip install drf-sideload
```

Import Mixin *SideloadingRelationsMixin*:

```
from drf_sideload.mixins import SideloadingRelationsMixin
```

Include mixin in view and define serializers dict `sideloading_relations` as shown in examples

It is required to define and indicate a primary relationship in `sideloading_relations` dict

Example of using mixin in `ViewSet`

```
class ProductViewSet(SideloadingRelationsMixin, viewsets.ModelViewSet):
    """
    A simple ViewSet for viewing and editing products.
    """
    queryset = Product.objects.all()
    serializer_class = ProductSerializer

    sideloading_relations = {
```

(continues on next page)

(continued from previous page)

```
'products': {'primary': True, 'serializer': ProductSerializer},
'categories': {'serializer': CategorySerializer, 'source': 'category',
→ 'prefetch': 'category'},
'suppliers': {'serializer': SupplierSerializer, 'source': 'supplier',
→ 'prefetch': 'supplier'},
'partners': {'serializer': PartnerSerializer, 'source': 'partners', 'prefetch
→ ': 'partners'}
}
```

Request:

```
GET /product/?sideload=categories,partners,suppliers
```

Response:

```
{
  "categories": [
    {
      "id": 1,
      ...
    }
  ],
  "partners": [
    {
      "id": 1,
      ...
    },
    {
      "id": 2,
      ...
    },
    {
      "id": 3,
      ...
    }
  ],
  "products": [
    {
      "id": 1,
      "name": "Product 1",
      "category": 1,
      "supplier": 1,
      "partner": [
        1,
        2,
        3
      ]
    }
  ],
  "suppliers": [
    {
      "id": 1,
      ...
    }
  ]
}
```


1.3 Example Project

directory *example* includes example project you can setup and run int locally using following commands

```
cd example
sh scripts/devsetup.sh
sh scripts/dev.sh
```

1.4 Contributing

For detailed description see [CONTRIBUTING Notes](#)

Setup for contribution

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install -r requirements_dev.txt
```

Test with specific env

```
$ tox --listenvs
py27-django18-drf34
py27-django19-drf34
# ...
$ tox -e py27-django19-drf34
```

Test coverage

```
$ make coverage
```

Use [pyenv](#) for testing using different versions locally

TODO

- fix documentation
- improve coverage

1.5 Credits

Tools used in rendering this package:

- [Cookiecutter](#)
- [cookiecutter-djangopackage](#)

CHAPTER 2

Installation

At the command line:

```
$ easy_install drf-sideloadings
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv drf-sideloadings
$ pip install drf-sideloadings
```


CHAPTER 3

Usage

Import sieldloading mixin

```
from drf_sideloadable.mixins import SideloadableRelationsMixin
```

Use in existing viewset by inheriting and defining *sideloadable_relations*

```
class ProductViewSet(SideloadableRelationsMixin, viewsets.ModelViewSet):
    """
    A simple ViewSet for viewing and editing products.
    """
    queryset = Product.objects.all()
    serializer_class = ProductSerializer
    sideloading_serializer_class = ProductSideloadableSerializer

class CategoryViewSet(SideloadableRelationsMixin, viewsets.ModelViewSet):
    """
    A more complex ViewSet with reverse relations.
```

Request:

```
GET /product/?sideload=categories,partners,suppliers
```


4.1 drf_sideloading package

4.1.1 Submodules

4.1.2 drf_sideloading.mixins module

```
class drf_sideloading.mixins.SideloadableRelationsMixin (**kwargs)
```

```
    Bases: object
```

TODO: Implement some protection for too large queries.

- **limit the number of sideloadable elements?** if over limit: - raise error - show warning - paginate, show first page and add a link to remaining paginated list of related elements? - show only the link to paginated list of related elements?

```
check_sideloading_serializer_class ()
```

```
filter_related_objects (related_objects, lookup)
```

```
get_primary_field_name ()
```

```
get_relevant_prefetches ()
```

```
get_sideloadable_fields ()
```

```
get_sideloadable_page (page)
```

```
get_sideloadable_page_from_queryset (queryset)
```

```
get_sideloading_prefetches ()
```

```
list (request, *args, **kwargs)
```

```
parse_query_param (sideload_parameter)
```

```
    Parse query param and take validated names
```

```
    :param sideload_parameter string :return valid relation names list
```

comma separated relation names may contain invalid or unusable characters. This function finds string match between requested names and defined relation in view

```
query_param_name = u'sideload'  
relations_to_sideload = None  
sideloading_serializer_class = None
```

4.1.3 drf_sideload.serializers module

```
class drf_sideload.serializers.SideLoadableSerializer(instance=None,  
                                                    data=<class  
                                                    rest_framework.fields.empty>,  
                                                    **kwargs)
```

Bases: `rest_framework.serializers.Serializer`

to_representation (*instance*)

Object instance -> Dict of primitive datatypes.

4.1.4 Module contents

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

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/namespace-ee/drf-sideload/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.

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 Write Documentation

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

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/namespace-ee/drf-sideload/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 :)

5.2 Get Started!

Ready to contribute? Here's how to set up *drf-sideload* for local development.

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

```
$ git clone git@github.com:namespace-ee/drf-sideload.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-sideload
$ cd drf-sideload/
$ 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 drf_sideload tests
$ python manage.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.

5.3 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 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/namespace-ee/django-rest-framework-sideloading/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_drf_sideloading
```


6.1 Contributors

- Demur Nodia <demur@namespace.ee>
- Tõnis Väin <tonis@namespace.ee>

None yet. Why not be the first?

7.1 0.1.10 (2017-07-20)

- Support for django2.0

7.2 0.1.8 (2017-07-20)

- change `sideloadable_relations` dict
- always required to define ‘serializer’
- key is referenced to url and serialized in as rendered json
- add *source* which specifies original model field name

7.3 0.1.0 (2017-07-20)

- First release on PyPI.

d

`drf_sideload`, [12](#)

`drf_sideload.mixins`, [11](#)

`drf_sideload.serializers`, [12](#)

C

`check_sideloadable_serializer_class()`
(`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11

D

`drf_sideloadable` (module), 12
`drf_sideloadable.mixins` (module), 11
`drf_sideloadable.serializers` (module), 12

F

`filter_related_objects()` (`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11

G

`get_primary_field_name()`
(`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11
`get_relevant_prefetches()`
(`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11
`get_sideloadable_fields()`
(`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11
`get_sideloadable_page()` (`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11
`get_sideloadable_page_from_queryset()`
(`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11
`get_sideloadable_prefetches()`
(`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11

L

`list()` (`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11

P

`parse_query_param()` (`drf_sideloadable.mixins.SideloadableRelationsMixin`
method), 11

Q

`query_param_name` (`drf_sideloadable.mixins.SideloadableRelationsMixin`
attribute), 12

R

`relations_to_sideload` (`drf_sideloadable.mixins.SideloadableRelationsMixin`
attribute), 12

S

`SideloadableRelationsMixin` (class in
`drf_sideloadable.mixins`), 11
`SideLoadableSerializer` (class in
`drf_sideloadable.serializers`), 12
`sideloading_serializer_class`
(`drf_sideloadable.mixins.SideloadableRelationsMixin`
attribute), 12

T

`to_representation()` (`drf_sideloadable.serializers.SideLoadableSerializer`
method), 12