

---

# **django-rest-framework-tricks**

## **Documentation**

*Release 0.1.8*

**Artur Barseghyan <[artur.barseghyan@gmail.com](mailto:artur.barseghyan@gmail.com)>**

**Jun 30, 2017**



<b>1</b>	<b>Prerequisites</b>	<b>3</b>
<b>2</b>	<b>Dependencies</b>	<b>5</b>
<b>3</b>	<b>Installation</b>	<b>7</b>
<b>4</b>	<b>Main features and highlights</b>	<b>9</b>
<b>5</b>	<b>Usage examples</b>	<b>11</b>
5.1	Nested serializers . . . . .	11
5.1.1	Sample model . . . . .	12
5.1.1.1	Required imports . . . . .	12
5.1.1.2	Model definition . . . . .	12
5.1.2	Sample serializers . . . . .	13
5.1.2.1	Required imports . . . . .	13
5.1.2.2	Defining the serializers . . . . .	13
5.1.3	Sample ViewSet . . . . .	14
5.1.3.1	Required imports . . . . .	15
5.1.3.2	ViewSet definition . . . . .	15
5.1.3.3	Sample OPTIONS call . . . . .	15
<b>6</b>	<b>Testing</b>	<b>19</b>
<b>7</b>	<b>Writing documentation</b>	<b>21</b>
<b>8</b>	<b>License</b>	<b>23</b>
<b>9</b>	<b>Support</b>	<b>25</b>
<b>10</b>	<b>Author</b>	<b>27</b>
<b>11</b>	<b>Documentation</b>	<b>29</b>
<b>12</b>	<b>Indices and tables</b>	<b>31</b>



Collection of various tricks for Django REST framework.



# CHAPTER 1

---

## Prerequisites

---

- Django 1.8, 1.9, 1.10 and 1.11.
- Python 2.7, 3.4, 3.5, 3.6





## CHAPTER 2

---

### Dependencies

---

- `djangorestframework`



1. Install latest stable version from PyPI:

```
pip install django-rest-framework-tricks
```

or latest stable version from GitHub:

```
pip install https://github.com/barseghyanartur/django-rest-framework-tricks/  
↪archive/stable.tar.gz
```

2. Add `rest_framework` and `rest_framework_tricks` to `INSTALLED_APPS`:

```
INSTALLED_APPS = (  
    # ...  
    # REST framework  
    'rest_framework',  
  
    # REST framework tricks (this package)  
    'rest_framework_tricks',  
  
    # ...  
)
```



## CHAPTER 4

---

### Main features and highlights

---

- *Nested serializers*: Nested serializers for non-relational fields.



## Nested serializers

Nested serializers for non-relational fields.

Our imaginary `Book` model consists of the following (non-relational) Django model fields:

- `title`: `CharField`
- `description`: `TextField`
- `summary`: `TextField`
- `publication_date`: `DateTimeField`
- `state`: `CharField` (with choices)
- `isbn`: `CharField`
- `price`: `DecimalField`
- `pages`: `IntegerField`
- `stock_count`: `IntegerField`

In our REST API, we want to split serializer into parts using nested serializers and have the following structure:

```
{
  "id": "",
  "title": "",
  "description": "",
  "summary": "",
  "publishing_information": {
    "publication_date": "",
    "isbn": "",
    "pages": ""
  },
  "stock_information": {
    "stock_count": "",
```

```
        "price": "",
        "state": ""
    }
}
```

## Sample model

The only customisation here is that we declare two `NestedProxyField` fields on the `Book` model level for to be used in `BookSerializer` serializer.

Note, that the change does not cause model change (no migrations or whatever).

## Required imports

```
from django.db import models

from rest_framework_tricks.models.fields import NestedProxyField
```

## Model definition

```
BOOK_PUBLISHING_STATUS_PUBLISHED = 'published'
BOOK_PUBLISHING_STATUS_NOT_PUBLISHED = 'not_published'
BOOK_PUBLISHING_STATUS_IN_PROGRESS = 'in_progress'
BOOK_PUBLISHING_STATUS_CANCELLED = 'cancelled'
BOOK_PUBLISHING_STATUS_REJECTED = 'rejected'
BOOK_PUBLISHING_STATUS_CHOICES = (
    (BOOK_PUBLISHING_STATUS_PUBLISHED, "Published"),
    (BOOK_PUBLISHING_STATUS_NOT_PUBLISHED, "Not published"),
    (BOOK_PUBLISHING_STATUS_IN_PROGRESS, "In progress"),
    (BOOK_PUBLISHING_STATUS_CANCELLED, "Cancelled"),
    (BOOK_PUBLISHING_STATUS_REJECTED, "Rejected"),
)
BOOK_PUBLISHING_STATUS_DEFAULT = BOOK_PUBLISHING_STATUS_PUBLISHED

class Book(models.Model):
    """Book."""

    title = models.CharField(max_length=100)
    description = models.TextField(null=True, blank=True)
    summary = models.TextField(null=True, blank=True)
    publication_date = models.DateField()
    state = models.CharField(max_length=100,
                            choices=BOOK_PUBLISHING_STATUS_CHOICES,
                            default=BOOK_PUBLISHING_STATUS_DEFAULT)
    isbn = models.CharField(max_length=100, unique=True)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    pages = models.PositiveIntegerField(default=200)
    stock_count = models.PositiveIntegerField(default=30)

    # List the fields for `PublishingInformationSerializer` nested
    # serializer. This does not cause a model change.
    publishing_information = NestedProxyField(
```



```

        'publication_date',
        'isbn',
        'pages',
    )

    # List the fields for `StockInformationSerializer` nested serializer.
    # This does not cause a model change.
    stock_information = NestedProxyField(
        'stock_count',
        'price',
        'state',
    )

    class Meta(object):
        """Meta options."""

        ordering = ["isbn"]

    def __str__(self):
        return self.title

```

## Sample serializers

At first, we add `nested_proxy_field` property to the `Meta` class definitions of `PublishingInformationSerializer` and `StockInformationSerializer` nested serializers.

Then we define our (main) `BookSerializer` class, which is going to be used a `serializer_class` of the `BookViewSet`. We inherit the `BookSerializer` from `rest_framework_tricks.serializers.HyperlinkedModelSerializer` instead of the one of the Django REST framework.

### Required imports

```

from rest_framework import serializers
from rest_framework_tricks.serializers import (
    HyperlinkedModelSerializer,
)

from .models import Book

```

## Defining the serializers

### Nested serializer

```

class PublishingInformationSerializer(serializers.ModelSerializer):
    """Publishing information serializer."""

    publication_date = serializers.DateField(required=False)
    isbn = serializers.CharField(required=False)
    pages = serializers.IntegerField(required=False)

    class Meta(object):
        """Meta options."""

```

```

model = Book
fields = (
    'publication_date',
    'isbn',
    'pages',
)
# Note, that this should be set to True to identify that
# this serializer is going to be used as `NestedProxyField`.
nested_proxy_field = True

```

### Nested serializer

```

class StockInformationSerializer(serializers.ModelSerializer):
    """Stock information serializer."""

    class Meta(object):
        """Meta options."""

        model = Book
        fields = (
            'stock_count',
            'price',
            'state',
        )
        # Note, that this should be set to True to identify that
        # this serializer is going to be used as `NestedProxyField`.
        nested_proxy_field = True

```

### Main serializer to be used in the ViewSet

```

# Note, that we are importing the `HyperlinkedModelSerializer` from
# the `rest_framework_tricks.serializers`.
class BookSerializer(HyperlinkedModelSerializer):
    """Book serializer."""

    publishing_information = PublishingInformationSerializer(required=False)
    stock_information = StockInformationSerializer(required=False)

    class Meta(object):
        """Meta options."""

        model = Book
        fields = (
            'url',
            'id',
            'title',
            'description',
            'summary',
            'publishing_information',
            'stock_information',
        )

```

### Sample ViewSet

Absolutely no customisations here.

## Required imports

```

from rest_framework.viewsets import ModelViewSet
from rest_framework.permissions import AllowAny

from .models import Book
from .serializers import BookSerializer

```

## ViewSet definition

```

class BookViewSet(ModelViewSet):
    """Book ViewSet."""

    queryset = Book.objects.all()
    serializer_class = BookSerializer
    permission_classes = [AllowAny]

```

## Sample OPTIONS call

```

OPTIONS /books/api/books/
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```

```

{
  "name": "Book List",
  "description": "Book ViewSet.",
  "renders": [
    "application/json",
    "text/html"
  ],
  "parses": [
    "application/json",
    "application/x-www-form-urlencoded",
    "multipart/form-data"
  ],
  "actions": {
    "POST": {
      "id": {
        "type": "integer",
        "required": false,
        "read_only": true,
        "label": "ID"
      },
      "title": {
        "type": "string",
        "required": true,
        "read_only": false,
        "label": "Title",
        "max_length": 100
      },
      "description": {
        "type": "string",

```

```
        "required": false,
        "read_only": false,
        "label": "Description"
    },
    "summary": {
        "type": "string",
        "required": false,
        "read_only": false,
        "label": "Summary"
    },
    "publishing_information": {
        "type": "nested object",
        "required": false,
        "read_only": false,
        "label": "Publishing information",
        "children": {
            "publication_date": {
                "type": "date",
                "required": false,
                "read_only": false,
                "label": "Publication date"
            },
            "isbn": {
                "type": "string",
                "required": false,
                "read_only": false,
                "label": "Isbn"
            },
            "pages": {
                "type": "integer",
                "required": false,
                "read_only": false,
                "label": "Pages"
            }
        }
    },
    "stock_information": {
        "type": "nested object",
        "required": false,
        "read_only": false,
        "label": "Stock information",
        "children": {
            "stock_count": {
                "type": "integer",
                "required": false,
                "read_only": false,
                "label": "Stock count"
            },
            "price": {
                "type": "decimal",
                "required": true,
                "read_only": false,
                "label": "Price"
            },
            "state": {
                "type": "choice",
                "required": false,
                "read_only": false,
```

```
    "label": "State",
    "choices": [
        {
            "value": "published",
            "display_name": "Published"
        },
        {
            "value": "not_published",
            "display_name": "Not published"
        },
        {
            "value": "in_progress",
            "display_name": "In progress"
        },
        {
            "value": "cancelled",
            "display_name": "Cancelled"
        },
        {
            "value": "rejected",
            "display_name": "Rejected"
        }
    ]
}
}
```



## CHAPTER 6

---

### Testing

---

Project is covered with tests.

To test with all supported Python/Django versions type:

```
tox
```

To test against specific environment, type:

```
tox -e py36-django110
```

To test just your working environment type:

```
./runtests.py
```

To run a single test in your working environment type:

```
./runtests.py src/rest_framework_tricks/tests/test_nested_proxy_field.py
```

Or:

```
./manage.py test rest_framework_tricks.tests.test_nested_proxy_field
```

It's assumed that you have all the requirements installed. If not, first install the test requirements:

```
pip install -r examples/requirements/test.txt
```





Keep the following hierarchy.

```
=====  
title  
=====  
  
header  
=====  
  
sub-header  
-----  
  
sub-sub-header  
~~~~~  
  
sub-sub-sub-header  
^^^^^^  
  
sub-sub-sub-sub-header  
+++++  
  
sub-sub-sub-sub-sub-header  
*****
```



## CHAPTER 8

---

License

---

GPL 2.0/LGPL 2.1



## CHAPTER 9

---

### Support

---

For any issues contact me at the e-mail given in the *Author* section.



## CHAPTER 10

---

Author

---

Artur Barseghyan <[artur.barseghyan@gmail.com](mailto:artur.barseghyan@gmail.com)>





Contents:

### Table of Contents

- *django-rest-framework-tricks*
  - *Prerequisites*
  - *Dependencies*
  - *Installation*
  - *Main features and highlights*
  - *Usage examples*
    - \* *Nested serializers*
      - *Sample model*
      - *Required imports*
      - *Model definition*
      - *Sample serializers*
      - *Required imports*
      - *Defining the serializers*
      - *Sample ViewSet*
      - *Required imports*
      - *ViewSet definition*
      - *Sample OPTIONS call*
  - *Testing*

- *Writing documentation*
- *License*
- *Support*
- *Author*
- *Documentation*
- *Indices and tables*

## CHAPTER 12

---

### Indices and tables

---

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