
Django Opt-out application Documentation

Release 0.9.3

Janusz Skonieczny

May 07, 2018

Contents

1	Django Opt-out application	3
1.1	Features	3
1.2	Demo	4
1.3	Quickstart	4
1.4	Running Tests	5
1.5	Credits	5
2	Installation	7
2.1	Stable release	7
2.2	From sources	7
3	Usage	9
4	Api docs	11
4.1	django_opt_out package	11
5	Contributing	25
5.1	Types of Contributions	25
5.2	Get Started!	26
5.3	Pull Request Guidelines	27
5.4	Tips	27
6	Credits	29
6.1	Development Lead	29
6.2	Contributors	29
7	History	31
7.1	0.1.0 (2017-11-03)	31
8	Indices and tables	33
	Python Module Index	35

Contents:

Django Opt-out application

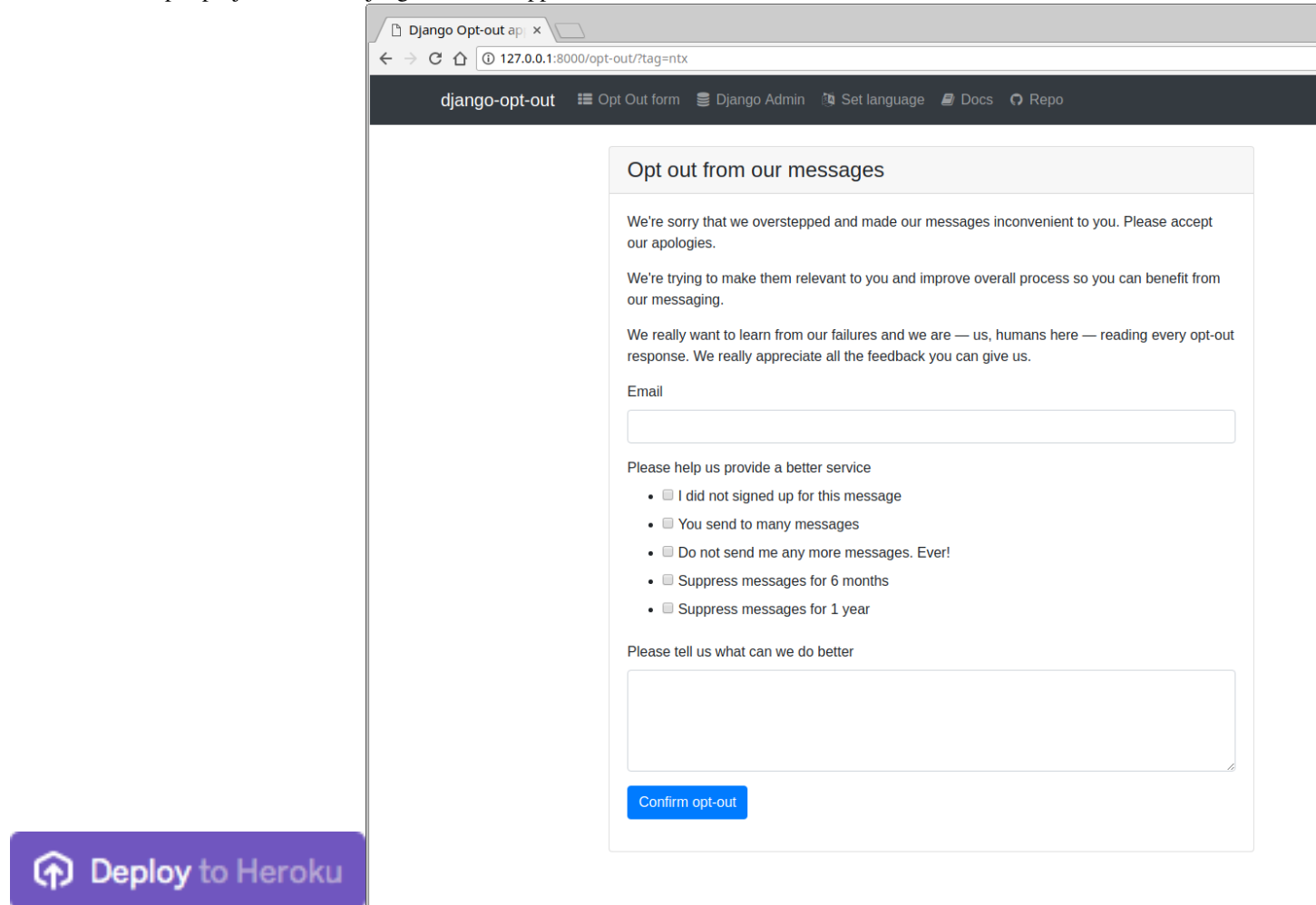
Documentation: <https://django-opt-out.readthedocs.io>.

1.1 Features

- A single page form for opt-out feedback submission
- Feedback options text controlled from django admin
- Predefined feedback defaults available from django manage command
- Feedback translations done in django admin
- Feedback options selection based on tags supplied to the opt-out url
- Ability to preselect a feedback option
- Ability to change selected feedback options after submission
- Ability to set `tag:value` pair on opt-out url and store them on submission with user feedback
- Signal to modify opt-out form before rendering
- Signal on opt-out feedback submission
- Easily overridable thank you / goodbye view
- Opt-out form with a easily overridable base template

1.2 Demo

To run an example project for this django reusable app, click the button below and start a demo server on Heroku

The image shows a web browser window displaying the Django Opt-out application. The browser's address bar shows the URL '127.0.0.1:8000/opt-out/?tag=ntx'. The application's navigation bar includes links for 'Opt Out form', 'Django Admin', 'Set language', 'Docs', and 'Repo'. The main content area is titled 'Opt out from our messages' and contains an apology message, a feedback form with checkboxes for various issues, and a 'Confirm opt-out' button. To the left of the browser window is a purple button with the Heroku logo and the text 'Deploy to Heroku'.

1.3 Quickstart

Install Django Opt-out application:

```
pip install django-opt-out
```

Add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'django_opt_out.apps.DjangoOptOutConfig',  
    ...  
)
```

Add Django Opt-out application's URL patterns:

```
from django_opt_out import urls as django_opt_out_urls

urlpatterns = [
    ...
    url(r'^$', include(django_opt_out_urls)),
    ...
]
```

Add unsubscribe links to your emails:

```
from django_opt_out.utils import get_opt_out_path
email='Django Opt-out <django-opt-out@niepodam.pl>'
unsubscribe = get_opt_out_path(email, 'some', 'tags', 'controlling', 'questionnaire')

# unsubscribe link will not have a domain name and scheme
# you can build prefix from request, but I prefer to set it in settings
from django.conf import settings
unsubscribe = settings.BASE_URL + unsubscribe
body = 'Hello, Regards\n\nUnsubscribe: ' + unsubscribe

from django.core import mail
message = mail.EmailMultiAlternatives(body=body, to=[email])
message.extra_headers['List-Unsubscribe'] = "<{}>".format(unsubscribe)
message.send()
```

1.4 Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install tox
(myenv) $ tox
```

1.5 Credits

This package was created with [Cookiecutter](#) and the [wooyek/cookiecutter-django-app](#) project template.

2.1 Stable release

To install Django Opt-out application, run this command in your terminal:

```
$ pip install django-opt-out
```

This is the preferred method to install Django Opt-out application, 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

You can either clone the public repository:

```
$ git clone git://github.com/wooyek/django-opt-out
```

Or download the download source from [project website](#). Once you have a copy of the source, you can install it with:

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


To use Django Opt-out application in a project, add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'django_opt_out.apps.DjangoOptOutConfig',  
    ...  
)
```

Add Django Opt-out application's URL patterns:

```
from django_opt_out import urls as django_opt_out_urls  
  
urlpatterns = [  
    ...  
    url(r'^$', include(django_opt_out_urls)),  
    ...  
)
```


4.1 django_opt_out package

4.1.1 Subpackages

django_opt_out.management package

Subpackages

django_opt_out.management.commands package

Submodules

django_opt_out.management.commands.opt_out_feedback_defaults module

```
class django_opt_out.management.commands.opt_out_feedback_defaults.Command(stdout=None,
                                                                    stderr=None,
                                                                    no_color=False)

    Bases: django.core.management.base.BaseCommand

    add_arguments (parser)
        Entry point for subclassed commands to add custom arguments.

    data_files = (('tags.csv', <class 'django_opt_out.resources.OptOutTagResource'>), ('fe

    handle (*args, **options)
        The actual logic of the command. Subclasses must implement this method.

    help = 'Imports default opt-out feedback options to empty database'

    import_all ()

    static import_data_file (name, resource)
```

Module contents

Module contents

django_opt_out.plugins package

Subpackages

django_opt_out.plugins.sparkpost package

Submodules

django_opt_out.plugins.sparkpost.apps module

```
class django_opt_out.plugins.sparkpost.apps.DjangoOptOutSparkPostConfig(app_name,  
                                                                           app_module)  
    Bases: django.apps.config.AppConfig  
    name = 'django_opt_out.plugins.sparkpost'  
    ready()  
        Override this method in subclasses to run code when Django starts.  
    verbose_name = 'Messaging Opt-Outs SparkPost plugin'
```

django_opt_out.plugins.sparkpost.hooks module

```
django_opt_out.plugins.sparkpost.hooks.create_opt_out(sender, request, email, data,  
                                                       **kwargs)  
django_opt_out.plugins.sparkpost.hooks.get_client(setting='SPARKPOST_API_KEY')  
django_opt_out.plugins.sparkpost.hooks.remove_suppression(sender, view, request,  
                                                           opt_out, **kwargs)  
django_opt_out.plugins.sparkpost.hooks.suppress_email(sender, view, request, opt_out,  
                                                         **kwargs)
```

django_opt_out.plugins.sparkpost.signals module

django_opt_out.plugins.sparkpost.urls module

django_opt_out.plugins.sparkpost.views module

```
class django_opt_out.plugins.sparkpost.views.SparkPostUnsubscribeWebhook(**kwargs)  
    Bases: django.views.generic.base.View  
    dispatch(*args, **kwargs)  
    post(request, *args, **kwargs)  
    process_data(data)
```

Module contents

`django_opt_out.plugins.sparkpost.send_email(subject, to, ctx, template_html, template_txt=None, **kwargs)`
 Send rendered message with SparkPost unsubscribe support

Module contents

4.1.2 Submodules

4.1.3 `django_opt_out.admin` module

```
class django_opt_out.admin.OptOutAdmin(model, admin_site)
    Bases: import_export.admin.ImportExportMixin, django.contrib.admin.options.
    ModelAdmin

    date_hierarchy = 'ts'

    list_display = ('ts', 'email', 'ip', 'host', 'ua')

    list_filter = ('ip', 'host', 'ua', 'feedback', 'tags')

    media

    readonly_fields = ('email', 'ts', 'comment', 'ip', 'host', 'ua', 'cookies', 'data', 's')

    resource_class
        alias of django_opt_out.resources.OptOutResource

class django_opt_out.admin.OptOutFeedbackAdmin(model, admin_site)
    Bases: import_export.admin.ImportExportMixin, django.contrib.admin.options.
    ModelAdmin

    all_tag_names(obj)

    inlines = [<class 'django_opt_out.admin.OptOutFeedbackTranslationInline'>]

    list_display = ('text', 'slug', 'default', 'ordinal', 'all_tag_names')

    list_filter = ('tags', 'default')

    media

    queryset(request, queryset)

    resource_class
        alias of django_opt_out.resources.OptOutFeedbackResource

class django_opt_out.admin.OptOutFeedbackTranslationAdmin(model, admin_site)
    Bases: import_export.admin.ImportExportMixin, django.contrib.admin.options.
    ModelAdmin

    list_display = ('text', 'language', 'feedback')

    media

    resource_class
        alias of django_opt_out.resources.OptOutFeedbackTranslationResource

class django_opt_out.admin.OptOutFeedbackTranslationInline(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline
```

```
media
model
    alias of django_opt_out.models.OptOutFeedbackTranslation
resource_class
    alias of django_opt_out.resources.OptOutFeedbackResource
class django_opt_out.admin.OptOutTagAdmin(model, admin_site)
    Bases: import_export.admin.ImportExportMixin, django.contrib.admin.options.
    ModelAdmin
    list_display = ('name', 'description')
media
resource_class
    alias of django_opt_out.resources.OptOutTagResource
```

4.1.4 django_opt_out.app_settings module

4.1.5 django_opt_out.apps module

```
class django_opt_out.apps.DjangoOptOutConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig
    name = 'django_opt_out'
    ready()
        Override this method in subclasses to run code when Django starts.
    verbose_name = 'Messaging Opt-Outs'
django_opt_out.apps.setup_app_settings()
```

4.1.6 django_opt_out.cli module

Console script for django-opt-out.

4.1.7 django_opt_out.factories module

```
class django_opt_out.factories.OptOutFactory
    Bases: factory.django.DjangoModelFactory
    email = <factory.faker.Faker object>
    ts = <factory.declarations.LazyFunction object>
class django_opt_out.factories.OptOutFeedbackFactory
    Bases: factory.django.DjangoModelFactory
    tags = <factory.declarations.PostGeneration object>
    text = <factory.faker.Faker object>
class django_opt_out.factories.OptOutFeedbackTranslationFactory
    Bases: factory.django.DjangoModelFactory
    feedback = <factory.declarations.SubFactory object>
```

```

    language = 'pl'
    text = <factory.faker.Faker object>
class django_opt_out.factories.OptOutTagFactory
    Bases: factory.django.DjangoModelFactory
    name = <factory.faker.Faker object>
class django_opt_out.factories.OptOutTagValueFactory
    Bases: factory.django.DjangoModelFactory
    opt_out = <factory.declarations.SubFactory object>
    tag = <factory.declarations.SubFactory object>
class django_opt_out.factories.UserFactory
    Bases: factory.django.DjangoModelFactory
    email = <factory.faker.Faker object>
    first_name = <factory.faker.Faker object>
    is_active = True
    is_staff = False
    last_name = <factory.faker.Faker object>
    username = <factory.declarations.Sequence object>

```

4.1.8 django_opt_out.forms module

```

class django_opt_out.forms.OptOutForm(data=None, files=None, auto_id='id_%s', pre-
                                     fix=None, initial=None, error_class=<class
                                     'django.forms.utils.ErrorList'>, label_suffix=None,
                                     empty_permitted=False, instance=None,
                                     use_required_attribute=None)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object
        fields = ('email', 'feedback', 'comment')
        labels = {'comment': 'Please tell us what can we do better'}
        model
            alias of django_opt_out.models.OptOut
    base_fields = {'comment': <django.forms.fields.CharField object at 0x7f9ff9d916a0>, '
    declared_fields = {'feedback': <django_opt_out.forms.TranslatedMultipleChoiceField ob
    media
    required_css_class = 'required'
    save(commit=True)
        Save this form's self.instance object if commit=True. Otherwise, add a save_m2m() method to the form
        which can be called after the instance is saved manually at a later time. Return the model instance.
class django_opt_out.forms.TranslatedMultipleChoiceField(queryset, **kwargs)
    Bases: django.forms.models.ModelMultipleChoiceField

```

label_from_instance (*obj*)

Convert objects into strings and generate the labels for the choices presented by this object. Subclasses can override this method to customize the display of the choices.

4.1.9 django_opt_out.models module

class `django_opt_out.models.OptOut` (*id, email, ts, confirmed, data, comment, secret, ssl, host, ip, ua, cookies*)

Bases: `django_powerbank.db.models.base.BaseModel`

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

comment

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

confirmed

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

cookies

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

data

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

email

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

feedback

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza (Model) :  
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_next_by_ts (*, *field=<django.db.models.fields.DateTimeField: ts>, is_next=True, **kwargs*)

get_previous_by_ts (*, *field=<django.db.models.fields.DateTimeField: ts>, is_next=False, **kwargs*)

host

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ip

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

save (*force_insert=False, force_update=False, using=None, update_fields=None*)

Save the current instance. Override this in a subclass if you want to control the saving process.

The ‘force_insert’ and ‘force_update’ parameters can be used to insist that the “save” must be an SQL insert or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

secret

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ssl

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

tags

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

ts

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

ua

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class django_opt_out.models.OptOutFeedback (*id, text, slug, default, ordinal*)

Bases: django_powerbank.db.models.base.BaseModel

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

default

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

ordinal

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

out_outs

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

slug

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

tags

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

text

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

trans (*language=None*)**translations**

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

class `django_opt_out.models.OptOutFeedbackTranslation` (*id, feedback, language, text*)

Bases: `django_powerbank.db.models.base.BaseModel`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

feedback

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

feedback_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_language_display (*, field=<django.db.models.fields.CharField: language>)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

language

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

text

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class django_opt_out.models.OptOutTag(id, name, description)
```

Bases: `django_powerbank.db.models.base.BaseModel`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

description

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

feedback

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

tags

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
class django_opt_out.models.OptOutTagValue(id, opt_out, tag, value)
```

Bases: `django_powerbank.db.models.base.BaseModel`

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = `<django.db.models.manager.Manager object>`

opt_out

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

opt_out_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

tag

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

tag_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

value

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

4.1.10 django_opt_out.resources module

```
class django_opt_out.resources.OptOutFeedbackResource
    Bases: import_export.resources.ModelResource

    class Meta
        Bases: object

        export_order = ('text',)

        fields = ('id', 'text', 'slug', 'default', 'ordinal', 'tags')

        model
            alias of django_opt_out.models.OptOutFeedback

        fields = {'default': <import_export.fields.Field: default>, 'id': <import_export.fi

class django_opt_out.resources.OptOutFeedbackTranslationResource
    Bases: import_export.resources.ModelResource

    class Meta
        Bases: object

        export_order = ('text',)

        fields = ('id', 'feedback', 'text', 'language')

        model
            alias of django_opt_out.models.OptOutFeedbackTranslation

        fields = {'feedback': <import_export.fields.Field: feedback>, 'id': <import_export.

class django_opt_out.resources.OptOutResource
    Bases: import_export.resources.ModelResource

    class Meta
        Bases: object

        export_order = ('ts',)

        fields = ('email', 'ts', 'confirmed', 'data', 'comment', 'feedback', 'secret', 'ss

        model
            alias of django_opt_out.models.OptOut

        fields = {'comment': <import_export.fields.Field: comment>, 'confirmed': <import_ex

class django_opt_out.resources.OptOutTagResource
    Bases: import_export.resources.ModelResource

    class Meta
        Bases: object

        export_order = ('name',)

        fields = ('id', 'name')

        model
            alias of django_opt_out.models.OptOutTag

        fields = {'id': <import_export.fields.Field: id>, 'name': <import_export.fields.Fie

class django_opt_out.resources.OptOutTagValueResource
    Bases: import_export.resources.ModelResource
```

```
class Meta
    Bases: object

    export_order = ('text',)

    fields = ('opt_out', 'tag', 'value')

    model
        alias of django_opt_out.models.OptOutTagValue

    fields = {'opt_out': <import_export.fields.Field: tags>, 'tag': <import_export.fiel
```

4.1.11 django_opt_out.signals module

`django_opt_out.signals.send_signal(signal, *args, **kwargs)`
Log any errors returned in the signal call responses

4.1.12 django_opt_out.urls module

4.1.13 django_opt_out.utils module

`django_opt_out.utils.get_opt_out_path(email, *tags)`
`django_opt_out.utils.get_opt_out_url(email, base_url=None, *tags)`
`django_opt_out.utils.get_password(email)`
`django_opt_out.utils.validate_password(email, encoded)`

4.1.14 django_opt_out.views module

```
class django_opt_out.views.OptOutBase(**kwargs)
    Bases: django_powerbank.views.auth.AbstractAccessView

    check_authorization(*args, **kwargs)

class django_opt_out.views.OptOutConfirm(**kwargs)
    Bases: pascal_templates.views.CreateView

    email_confirmed()

    form_class
        alias of django_opt_out.forms.OptOutForm

    form_valid(form)
        If the form is valid, save the associated model.

    get_context_data(**kwargs)
        Insert the form into the context dict.

    get_form(form_class=None)
        Return an instance of the form to be used in this view.

    get_initial()
        Return the initial data to use for forms on this view.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    get_tags()
```

```
model
    alias of django_opt_out.models.OptOut

save_tags()

template_name = 'django_opt_out/OptOut/form.html'

class django_opt_out.views.OptOutRemoved(**kwargs)
    Bases: django.views.generic.base.TemplateView

    template_name = 'django_opt_out/OptOut/removed.html'

class django_opt_out.views.OptOutSuccess(**kwargs)
    Bases: django_opt_out.views.OptOutBase, pascal_templates.views.DetailView

    model
        alias of django_opt_out.models.OptOut

    post(request, *args, **kwargs)

    template_name = 'django_opt_out/OptOut/success.html'

class django_opt_out.views.OptOutUpdate(**kwargs)
    Bases: django_opt_out.views.OptOutBase, pascal_templates.views.UpdateView

    form_class
        alias of django_opt_out.forms.OptOutForm

    get_form(form_class=None)
        Return an instance of the form to be used in this view.

    get_success_url()
        Return the URL to redirect to after processing a valid form.

    model
        alias of django_opt_out.models.OptOut

    template_name = 'django_opt_out/OptOut/form.html'
```

4.1.15 Module contents

Top-level package for Django Opt-out application.

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/wooyek/django-opt-out/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” and “help wanted” is open to whoever wants to implement it.

5.1.3 Implement Features

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

5.1.4 Write Documentation

Django Opt-out application could always use more documentation, whether as part of the official Django Opt-out application 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/wooyek/django-opt-out/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 *django-opt-out* for local development.

1. Fork the *django-opt-out* repo on github.com
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-opt-out.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-opt-out
$ cd django-opt-out/
$ 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 django-opt-out tests
$ tox -e check
$ 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.

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 versions mentioned in tox.ini file. Check https://travis-ci.org/wooyek/django-opt-out/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ pytest tests.test_models
```


CHAPTER 6

Credits

6.1 Development Lead

- @wooyek: Janusz Skonieczny <js+pypi@bravelabs.pl>

6.2 Contributors

None yet. Why not be the first?

7.1 0.1.0 (2017-11-03)

- First release on PyPI.

CHAPTER 8

Indices and tables

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

a

`django_opt_out.admin`, [13](#)
`django_opt_out.app_settings`, [14](#)
`django_opt_out.apps`, [14](#)

c

`django_opt_out.cli`, [14](#)

d

`django_opt_out`, [23](#)

f

`django_opt_out.factories`, [14](#)
`django_opt_out.forms`, [15](#)

m

`django_opt_out.management`, [12](#)
`django_opt_out.management.commands`, [12](#)
`django_opt_out.management.commands.opt_out_feedback_defaults`,
[11](#)
`django_opt_out.models`, [16](#)

p

`django_opt_out.plugins`, [13](#)
`django_opt_out.plugins.sparkpost`, [13](#)
`django_opt_out.plugins.sparkpost.apps`,
[12](#)
`django_opt_out.plugins.sparkpost.hooks`,
[12](#)
`django_opt_out.plugins.sparkpost.signals`,
[12](#)
`django_opt_out.plugins.sparkpost.urls`,
[12](#)
`django_opt_out.plugins.sparkpost.views`,
[12](#)

r

`django_opt_out.resources`, [21](#)

s

`django_opt_out.signals`, [22](#)

u

`django_opt_out.urls`, [22](#)
`django_opt_out.utils`, [22](#)

v

`django_opt_out.views`, [22](#)

A

add_arguments() (django_opt_out.management.commands.opt_out_feedback_defaults.Command method), 11

all_tag_names() (django_opt_out.admin.OptOutFeedbackAdmin method), 13

B

base_fields (django_opt_out.forms.OptOutForm attribute), 15

C

check_authorization() (django_opt_out.views.OptOutBase method), 22

Command (class in django_opt_out.management.commands), 11

comment (django_opt_out.models.OptOut attribute), 16

confirmed (django_opt_out.models.OptOut attribute), 16

cookies (django_opt_out.models.OptOut attribute), 16

create_opt_out() (in module django_opt_out.plugins.sparkpost.hooks), 12

D

data (django_opt_out.models.OptOut attribute), 16

data_files (django_opt_out.management.commands.opt_out_feedback_defaults.Command attribute), 11

date_hierarchy (django_opt_out.admin.OptOutAdmin attribute), 13

declared_fields (django_opt_out.forms.OptOutForm attribute), 15

default (django_opt_out.models.OptOutFeedback attribute), 17

description (django_opt_out.models.OptOutTag attribute), 19

dispatch() (django_opt_out.plugins.sparkpost.views.SparkPostUnsubscribeWebhook method), 12

django_opt_out (module), 23

django_opt_out.admin (module), 13

django_opt_out.app_settings (module), 14

django_opt_out.apps (module), 14

django_opt_out.cli (module), 14

django_opt_out.feedback_defaults.Command (class in django_opt_out.management.commands), 14

django_opt_out.factories (module), 14

django_opt_out.forms (module), 15

django_opt_out.management (module), 12

django_opt_out.management.commands (module), 12

django_opt_out.management.commands.opt_out_feedback_defaults (module), 11

django_opt_out.models (module), 16

django_opt_out.plugins (module), 13

django_opt_out.plugins.sparkpost (module), 13

django_opt_out.plugins.sparkpost.apps (module), 12

django_opt_out.plugins.sparkpost.hooks (module), 12

django_opt_out.plugins.sparkpost.signals (module), 12

django_opt_out.plugins.sparkpost.urls (module), 12

django_opt_out.plugins.sparkpost.views (module), 12

django_opt_out.resources (module), 21

django_opt_out.signals (module), 22

django_opt_out.urls (module), 22

django_opt_out.utils (module), 22

django_opt_out.views (module), 22

DjangoOptOutConfig (class in django_opt_out.apps), 14

DjangoOptOutSparkPostConfig (class in django_opt_out.plugins.sparkpost.apps), 12

E

email (django_opt_out.factories.OptOutFactory attribute), 14

email (django_opt_out.factories.UserFactory attribute), 15

email (django_opt_out.models.OptOut attribute), 16

email_confirmed() (django_opt_out.views.OptOutConfirm method), 22

export_order (django_opt_out.resources.OptOutFeedbackResource.Meta attribute), 21

export_order (django_opt_out.resources.OptOutFeedbackTranslationResource attribute), 21

export_order (django_opt_out.resources.OptOutResource.Meta attribute), 21

export_order (django_opt_out.resources.OptOutTagResource.Meta attribute), 21

export_order (django_opt_out.resources.OptOutTagValueResource.Meta attribute), 22

F

feedback (django_opt_out.factories.OptOutFeedbackTranslationFactory attribute), 14

feedback (django_opt_out.models.OptOut attribute), 16

feedback (django_opt_out.models.OptOutFeedbackTranslation attribute), 18

feedback (django_opt_out.models.OptOutTag attribute), 19

feedback_id (django_opt_out.models.OptOutFeedbackTranslation attribute), 19

fields (django_opt_out.forms.OptOutForm.Meta attribute), 15

fields (django_opt_out.resources.OptOutFeedbackResource attribute), 21

fields (django_opt_out.resources.OptOutFeedbackResource.Meta attribute), 21

fields (django_opt_out.resources.OptOutFeedbackTranslationResource attribute), 21

fields (django_opt_out.resources.OptOutFeedbackTranslationResource.Meta attribute), 21

fields (django_opt_out.resources.OptOutResource attribute), 21

fields (django_opt_out.resources.OptOutResource.Meta attribute), 21

fields (django_opt_out.resources.OptOutTagResource attribute), 21

fields (django_opt_out.resources.OptOutTagResource.Meta attribute), 21

fields (django_opt_out.resources.OptOutTagValueResource attribute), 22

fields (django_opt_out.resources.OptOutTagValueResource.Meta attribute), 22

first_name (django_opt_out.factories.UserFactory attribute), 15

form_class (django_opt_out.views.OptOutConfirm attribute), 22

form_class (django_opt_out.views.OptOutUpdate attribute), 23

form_valid() (django_opt_out.views.OptOutConfirm method), 22

G

get_client() (in module django_opt_out.plugins.sparkpost.hooks), 12

get_context_data() (django_opt_out.views.OptOutConfirm method), 22

get_form() (django_opt_out.views.OptOutConfirm method), 22

get_language_display() (django_opt_out.models.OptOutFeedbackTranslation method), 19

get_next_by_ts() (django_opt_out.models.OptOut method), 16

get_opt_out_path() (in module django_opt_out.utils), 22

get_opt_out_url() (in module django_opt_out.utils), 22

get_password() (in module django_opt_out.utils), 22

get_previous_by_ts() (django_opt_out.models.OptOut method), 16

get_success_url() (django_opt_out.views.OptOutConfirm method), 22

get_success_url() (django_opt_out.views.OptOutUpdate method), 23

get_tags() (django_opt_out.views.OptOutConfirm method), 22

help() (django_opt_out.management.commands.opt_out_feedback_defaults method), 11

help() (django_opt_out.management.commands.opt_out_feedback_defaults method), 11

host (django_opt_out.models.OptOut attribute), 16

id (django_opt_out.models.OptOut attribute), 16

id (django_opt_out.models.OptOutFeedback attribute), 17

id (django_opt_out.models.OptOutFeedbackTranslation attribute), 19

id (django_opt_out.models.OptOutTag attribute), 19

id (django_opt_out.models.OptOutTagValue attribute), 20

import_all() (django_opt_out.management.commands.opt_out_feedback_defaults method), 11

import_data_file() (django_opt_out.management.commands.opt_out_feedback_defaults static method), 11

inlines (django_opt_out.admin.OptOutFeedbackAdmin attribute), 13

ip (django_opt_out.models.OptOut attribute), 16

is_active (django_opt_out.factories.UserFactory attribute), 15

is_staff (django_opt_out.factories.UserFactory attribute), 15

L

label_from_instance() (django_opt_out.forms.TranslatedMultipleChoiceField method), 15

labels (django_opt_out.forms.OptOutForm.Meta attribute), 15

language (django_opt_out.factories.OptOutFeedbackTranslationFactory attribute), 14
 attribute), 14
 name (django_opt_out.factories.OptOutTagFactory attribute), 15
 language (django_opt_out.models.OptOutFeedbackTranslation attribute), 15
 attribute), 19
 name (django_opt_out.models.OptOutTag attribute), 19
 last_name (django_opt_out.factories.UserFactory attribute), 15
 name (django_opt_out.plugins.sparkpost.apps.DjangoOptOutSparkPostCon attribute), 12
 list_display (django_opt_out.admin.OptOutAdmin attribute), 13
 list_display (django_opt_out.admin.OptOutFeedbackAdmin attribute), 13
 list_display (django_opt_out.admin.OptOutFeedbackTranslationAdmin attribute), 13
 list_display (django_opt_out.admin.OptOutTagAdmin attribute), 14
 list_filter (django_opt_out.admin.OptOutAdmin attribute), 13
 list_filter (django_opt_out.admin.OptOutFeedbackAdmin attribute), 13

M

media (django_opt_out.admin.OptOutAdmin attribute), 13
 media (django_opt_out.admin.OptOutFeedbackAdmin attribute), 13
 media (django_opt_out.admin.OptOutFeedbackTranslationAdmin attribute), 13
 media (django_opt_out.admin.OptOutFeedbackTranslationInline attribute), 13
 media (django_opt_out.admin.OptOutTagAdmin attribute), 14
 media (django_opt_out.forms.OptOutForm attribute), 15
 model (django_opt_out.admin.OptOutFeedbackTranslationInline attribute), 14
 model (django_opt_out.forms.OptOutForm.Meta attribute), 15
 model (django_opt_out.resources.OptOutFeedbackResource.Meta attribute), 21
 model (django_opt_out.resources.OptOutFeedbackTranslationResource.Meta attribute), 21
 model (django_opt_out.resources.OptOutResource.Meta attribute), 21
 model (django_opt_out.resources.OptOutTagResource.Meta attribute), 21
 model (django_opt_out.resources.OptOutTagValueResource.Meta attribute), 22
 model (django_opt_out.views.OptOutConfirm attribute), 23
 model (django_opt_out.views.OptOutSuccess attribute), 23
 model (django_opt_out.views.OptOutUpdate attribute), 23

N

name (django_opt_out.apps.DjangoOptOutConfig attribute), 14

O

objects (django_opt_out.models.OptOut attribute), 17
 objects (django_opt_out.models.OptOutFeedback attribute), 17
 objects (django_opt_out.models.OptOutFeedbackTranslation attribute), 19
 objects (django_opt_out.models.OptOutTag attribute), 19
 objects (django_opt_out.models.OptOutTagValue attribute), 20
 opt_out (django_opt_out.factories.OptOutTagValueFactory attribute), 15
 opt_out (django_opt_out.models.OptOutTagValue attribute), 20
 opt_out_id (django_opt_out.models.OptOutTagValue attribute), 20
 OptOut (class in django_opt_out.models), 16
 OptOut.DoesNotExist, 16
 OptOut.MultipleObjectsReturned, 16
 OptOutAdmin (class in django_opt_out.admin), 13
 OptOutBase (class in django_opt_out.views), 22
 OptOutConfirm (class in django_opt_out.views), 22
 OptOutFactory (class in django_opt_out.factories), 14
 OptOutFeedback (class in django_opt_out.models), 17
 OptOutFeedback.DoesNotExist, 17
 OptOutFeedback.MultipleObjectsReturned, 17
 OptOutFeedbackAdmin (class in django_opt_out.admin), 13
 OptOutFeedbackFactory (class in django_opt_out.factories), 14
 OptOutFeedbackResource (class in django_opt_out.resources), 21
 OptOutFeedbackResource.Meta (class in django_opt_out.resources), 21
 OptOutFeedbackTranslation (class in django_opt_out.models), 18
 OptOutFeedbackTranslation.DoesNotExist, 18
 OptOutFeedbackTranslation.MultipleObjectsReturned, 18
 OptOutFeedbackTranslationAdmin (class in django_opt_out.admin), 13
 OptOutFeedbackTranslationFactory (class in django_opt_out.factories), 14
 OptOutFeedbackTranslationInline (class in django_opt_out.admin), 13
 OptOutFeedbackTranslationResource (class in django_opt_out.resources), 21

[OptOutFeedbackTranslationResource.Meta](#) (class in [django_opt_out.resources](#)), 21
[OptOutForm](#) (class in [django_opt_out.forms](#)), 15
[OptOutForm.Meta](#) (class in [django_opt_out.forms](#)), 15
[OptOutRemoved](#) (class in [django_opt_out.views](#)), 23
[OptOutResource](#) (class in [django_opt_out.resources](#)), 21
[OptOutResource.Meta](#) (class in [django_opt_out.resources](#)), 21
[OptOutSuccess](#) (class in [django_opt_out.views](#)), 23
[OptOutTag](#) (class in [django_opt_out.models](#)), 19
[OptOutTag.DoesNotExist](#), 19
[OptOutTag.MultipleObjectsReturned](#), 19
[OptOutTagAdmin](#) (class in [django_opt_out.admin](#)), 14
[OptOutTagFactory](#) (class in [django_opt_out.factories](#)), 15
[OptOutTagResource](#) (class in [django_opt_out.resources](#)), 21
[OptOutTagResource.Meta](#) (class in [django_opt_out.resources](#)), 21
[OptOutTagValue](#) (class in [django_opt_out.models](#)), 20
[OptOutTagValue.DoesNotExist](#), 20
[OptOutTagValue.MultipleObjectsReturned](#), 20
[OptOutTagValueFactory](#) (class in [django_opt_out.factories](#)), 15
[OptOutTagValueResource](#) (class in [django_opt_out.resources](#)), 21
[OptOutTagValueResource.Meta](#) (class in [django_opt_out.resources](#)), 21
[OptOutUpdate](#) (class in [django_opt_out.views](#)), 23
[ordinal](#) ([django_opt_out.models.OptOutFeedback](#) attribute), 17
[out_outs](#) ([django_opt_out.models.OptOutFeedback](#) attribute), 17

P

[post\(\)](#) ([django_opt_out.plugins.sparkpost.views.SparkPostUnsubscribeWebhook](#) method), 12
[post\(\)](#) ([django_opt_out.views.OptOutSuccess](#) method), 23
[process_data\(\)](#) ([django_opt_out.plugins.sparkpost.views.SparkPostUnsubscribeWebhook](#) method), 12

Q

[queryset\(\)](#) ([django_opt_out.admin.OptOutFeedbackAdmin](#) method), 13

R

[readonly_fields](#) ([django_opt_out.admin.OptOutAdmin](#) attribute), 13
[ready\(\)](#) ([django_opt_out.apps.DjangoOptOutConfig](#) method), 14
[ready\(\)](#) ([django_opt_out.plugins.sparkpost.apps.DjangoOptOutSparkPostConfig](#) method), 12
[remove_suppression\(\)](#) (in module [django_opt_out.plugins.sparkpost.hooks](#)), 12
[required_css_class](#) ([django_opt_out.forms.OptOutForm](#) attribute), 15
[resource_class](#) ([django_opt_out.admin.OptOutAdmin](#) attribute), 13
[resource_class](#) ([django_opt_out.admin.OptOutFeedbackAdmin](#) attribute), 13
[resource_class](#) ([django_opt_out.admin.OptOutFeedbackTranslationAdmin](#) attribute), 13
[resource_class](#) ([django_opt_out.admin.OptOutFeedbackTranslationInline](#) attribute), 14
[resource_class](#) ([django_opt_out.admin.OptOutTagAdmin](#) attribute), 14

S

[save\(\)](#) ([django_opt_out.forms.OptOutForm](#) method), 15
[save\(\)](#) ([django_opt_out.models.OptOut](#) method), 17
[save_tags\(\)](#) ([django_opt_out.views.OptOutConfirm](#) method), 23
[secret](#) ([django_opt_out.models.OptOut](#) attribute), 17
[send_email\(\)](#) (in module [django_opt_out.plugins.sparkpost](#)), 13
[send_signal\(\)](#) (in module [django_opt_out.signals](#)), 22
[setup_app_settings\(\)](#) (in module [django_opt_out.apps](#)), 14
[slug](#) ([django_opt_out.models.OptOutFeedback](#) attribute), 18
[SparkPostUnsubscribeWebhook](#) (class in [django_opt_out.plugins.sparkpost.views](#)), 12
[ssl](#) ([django_opt_out.models.OptOut](#) attribute), 17
[suppress_email\(\)](#) (in module [django_opt_out.plugins.sparkpost.hooks](#)), 12

T

[tag](#) ([django_opt_out.models.OptOutTagValue](#) attribute), 20
[tag_id](#) ([django_opt_out.models.OptOutTagValue](#) attribute), 20
[tags](#) ([django_opt_out.factories.OptOutFeedbackFactory](#) attribute), 14
[tags](#) ([django_opt_out.models.OptOut](#) attribute), 17
[tags](#) ([django_opt_out.models.OptOutFeedback](#) attribute), 18
[tags](#) ([django_opt_out.models.OptOutTag](#) attribute), 19
[template_name](#) ([django_opt_out.views.OptOutConfirm](#) attribute), 23
[template_name](#) ([django_opt_out.views.OptOutRemoved](#) attribute), 23

`template_name` (`django_opt_out.views.OptOutSuccess` attribute), [23](#)
`template_name` (`django_opt_out.views.OptOutUpdate` attribute), [23](#)
`text` (`django_opt_out.factories.OptOutFeedbackFactory` attribute), [14](#)
`text` (`django_opt_out.factories.OptOutFeedbackTranslationFactory` attribute), [15](#)
`text` (`django_opt_out.models.OptOutFeedback` attribute), [18](#)
`text` (`django_opt_out.models.OptOutFeedbackTranslation` attribute), [19](#)
`trans()` (`django_opt_out.models.OptOutFeedback` method), [18](#)
`TranslatedMultipleChoiceField` (class in `django_opt_out.forms`), [15](#)
`translations` (`django_opt_out.models.OptOutFeedback` attribute), [18](#)
`ts` (`django_opt_out.factories.OptOutFactory` attribute), [14](#)
`ts` (`django_opt_out.models.OptOut` attribute), [17](#)

U

`ua` (`django_opt_out.models.OptOut` attribute), [17](#)
`UserFactory` (class in `django_opt_out.factories`), [15](#)
`username` (`django_opt_out.factories.UserFactory` attribute), [15](#)

V

`validate_password()` (in module `django_opt_out.utils`), [22](#)
`value` (`django_opt_out.models.OptOutTagValue` attribute), [20](#)
`verbose_name` (`django_opt_out.apps.DjangoOptOutConfig` attribute), [14](#)
`verbose_name` (`django_opt_out.plugins.sparkpost.apps.DjangoOptOutSparkPostConfig` attribute), [12](#)