
django-relatives Documentation

Release 1

Trey Hunner

Dec 06, 2018

Contents

1	Contents	3
1.1	Usage	3
1.2	API Reference	7
2	Contributing	9
3	Examples	11
3.1	Turn read-only foreign key fields into hyperlinks	11
3.2	Link to child objects in the sidebar	11
4	Indices and tables	13
	Python Module Index	15

django-relatives provides various utilites that make it easy to link related objects in the Django admin site.

1.1 Usage

1.1.1 Installation

Install from PyPI:

```
$ pip install django-relatives
```

Since relatives uses cache, you can set settings to change the defaults:

```
RELATIVES_CACHE_KEY = 'relatives_cache'
RELATIVES_CACHE_TIME = int(60*60*24)
```

1.1.2 Edit links in inlines

To link to an inline object, include the *object_link* utility function in your admin inline's `fields` list and `readonly_fields` list.

Example

Code

```
from django.contrib import admin
from relatives.utils import object_link

from models import Company, Employee

class EmployeeInline(admin.TabularInline):
```

(continues on next page)

(continued from previous page)

```

model = Employee
fields = [object_link, 'first_name', 'last_name']
readonly_fields = fields
extra = 0
max_num = 0
can_delete = False

class CompanyAdmin(admin.ModelAdmin):
    inlines = [EmployeeInline]

admin.site.register(Company, CompanyAdmin)

admin.site.register(Employee)

```

Screenshot

Change company History

Name:

Employees		
	First name	Last name
8000	Elmer	Fudd
7000	Road	Runner

✖ Delete
Save and add another
Save and continue editing
Save

1.1.3 Customizing inline edit links

To customize the link text for your inline links, use the `object_edit_link` utility function instead, specifying the edit text and blank text (both are optional).

Example

Code

```

from django.contrib import admin
from relatives.utils import object_edit_link

from models import Company, Employee

class EmployeeInline(admin.TabularInline):

```

(continues on next page)

(continued from previous page)

```

model = Employee
edit_link = object_edit_link("Edit")
fields = [edit_link, 'employee_id', 'first_name', 'last_name']
readonly_fields = [edit_link]

class CompanyAdmin(admin.ModelAdmin):
    inlines = [EmployeeInline]

admin.site.register(Company, CompanyAdmin)

admin.site.register(Employee)

```

Screenshot

Change company History

Name:

Employees				
	Employee id	First name	Last name	Delete?
8000	<input type="text" value="8000"/>	<input type="text" value="Elmer"/>	<input type="text" value="Fudd"/>	<input type="checkbox"/>
7000	<input type="text" value="7000"/>	<input type="text" value="Road"/>	<input type="text" value="Runner"/>	<input type="checkbox"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	

[+ Add another Employee](#)

[✖ Delete](#) [Save and add another](#) [Save and continue editing](#) [Save](#)

1.1.4 Linking to foreign keys

The `contents_or_fk_link` template filter can be used to link to foreign keys for readonly admin form fields.

Django Relatives also provides a replacement for the `admin/includes/fieldset.html` template which can be used to automatically link to all readonly foreign key fields in change forms.

To use the custom fieldset template you must add `relatives` to `INSTALLED_APPS` in your settings file:

```

INSTALLED_APPS = (
    ...
    'relatives',
)

```

Next create a `admin/includes/fieldset.html` template file:

```
{% extends "relatives/includes/fieldset.html" %}
```

Also make sure this template file is in a custom template directory or an app listed before your admin app in `INSTALLED_APPS`.

Example Screenshot

1.1.5 Linking to reverse relations

The `related_objects` template tag makes it easy to link to change lists filtered for reverse relations (objects that have a foreign key to a given object).

Django Relatives also provides a custom `change_form.html` template that may be used to add a “Relations” sidebar to change forms. This sidebar provides links to change list queries for all objects that contain a foreign key to the current object.

To use the custom fieldset template you must add `relatives` to `INSTALLED_APPS` in your settings file:

```
INSTALLED_APPS = (
    ...
    'relatives',
)
```

Now you can customize the change form template for your desired models/apps. The easiest way to link to reverse relations is to override the `change_form_template` in your `ModelAdmin` subclass.

Example

Code

```
from django.contrib import admin

from models import Company, Employee

class CompanyAdmin(admin.ModelAdmin):
    change_form_template = 'relatives/change_form.html'
```

(continues on next page)

(continued from previous page)

```
admin.site.register(Company, CompanyAdmin)

admin.site.register(Employee)
```

Screenshot

Linking to reverse relations with custom template

If you don't have access to change the `ModelAdmin` for your model or you are already customizing your model's admin change form, you will need to use a custom admin template instead.

Create a `admin/YOURAPP/YOURMODEL/change_form.html` template file that extends from `relatives/change_form.html`:

```
{% extends "relatives/change_form.html" %}
```

Also make sure this template file is in a custom template directory or an app listed before your admin app in `INSTALLED_APPS`.

1.2 API Reference

1.2.1 Utility Functions

class `relatives.utils.GenericObjects` (*_object*)

Search `GenericForeignKey` over all models and returns related objects if has relations with given object.

class `relatives.utils.RelatedObject` (*field, ct_pk*)

Generates fake django `RelatedObject`

`relatives.utils.get_admin_url` (*obj*)

Return admin URL for given object (raise `NoReverseMatch` on error)

`relatives.utils.object_edit_link` (*edit_text=None, blank_text=None*)

Return function that takes an object and returns admin link to object

Arguments:

- `edit_text` is displayed in link text
- `blank_text` is displayed in unlinked text (when no admin link)

`edit_text` defaults to the object's unicode representation and `blank_text` defaults to the object's unicode representation if `edit_text` is `None` and an empty string otherwise

`relatives.utils.object_link(obj)`
Return admin link to given object or blank text if no link
Equivalent to `object_edit_link()(obj)`

1.2.2 Template Tags

`relatives.templatetags.relatives.contents_or_fk_link(field)`
Return field contents or link to related object if foreign key field

Example Usage:

```
{% load relatives %}
{{ field|contents_or_fk_link }}
```

`relatives.templatetags.relatives.related_objects(obj)`
Return list of objects related to the given model instance

Example Usage:

```
{% load relatives %}
{% related_objects obj as related_objects %}
{% for related_obj in related_objects %}
    <a href="{{ related_obj.url }}">{{ related_obj.plural_name }}</a>
{% endfor %}
```

CHAPTER 2

Contributing

Visit the project on [Github](#) to view the source, submit issues and pull requests.

3.1 Turn read-only foreign key fields into hyperlinks

Change company

History

Name:

ACME

Employees

	Employee id	First name	Last name	Delete?
8000				
Edit	8000	Elmer	Fudd	<input type="checkbox"/>
7000				
Edit	7000	Road	Runner	<input type="checkbox"/>

+

Add another Employee

✖ Delete

Save and add another

Save and continue editing

Save

3.2 Link to child objects in the sidebar

Change company

Name:	<input type="text" value="ACME"/>
✖ Delete Save and add another Save and continue editing Save	

History

Relations
Employees

CHAPTER 4

Indices and tables

- `genindex`
- `search`

r

`relatives.templatetags.relatives`, [8](#)

`relatives.utils`, [7](#)

C

`contents_or_fk_link()` (in module `relatives.templatetags.relatives`), 8

G

`GenericObjects` (class in `relatives.utils`), 7

`get_admin_url()` (in module `relatives.utils`), 7

O

`object_edit_link()` (in module `relatives.utils`), 7

`object_link()` (in module `relatives.utils`), 7

R

`related_objects()` (in module `relatives.templatetags.relatives`), 8

`RelatedObject` (class in `relatives.utils`), 7

`relatives.templatetags.relatives` (module), 8

`relatives.utils` (module), 7