
django-relatives Documentation

Release 0.2.0

Trey Hunner

July 16, 2013

CONTENTS

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

Contents:

USAGE

1.1 Installation

Install from PyPI:

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

1.2 Edit links in inlines

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

from models import Company, Employee

class EmployeeInline(admin.TabularInline):
    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)
```

Result:

1.2.1 Customizing edit link text

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

Change company

History

Name:

Employees		
	First name	Last name
8000	Elmer	Fudd
7000	Road	Runner

✖ Delete

```
from models import Company, Employee
```

```
class EmployeeInline(admin.TabularInline):
    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)
```

Result:

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

1.3 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:

```
{% include "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`.

Result:

Change employee History

Employee id:	<input type="text" value="7000"/>
First name:	<input type="text" value="Road"/>
Last name:	<input type="text" value="Runner"/>
Company:	<input type="text" value="ACME"/>

✖ Delete Save and add another Save and continue editing Save

1.4 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. Either:

1. Set the `change_form_template` in your custom model admin

2. Override your model's `change_form.html` template

1.4.1 Customize ModelAdmin

The easiest way to link to reverse relations is to override the `change_form_template` in your `ModelAdmin` subclass:

```
from django.contrib import admin

from models import Company, Employee

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

admin.site.register(Company, CompanyAdmin)

admin.site.register(Employee)
```

1.4.2 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.

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

```
{% include "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`.

1.4.3 Result

Both of the above methods will result in the following:

Change company History

Name:

✖ Delete

Relations

[Employees](#)

API REFERENCE

2.1 Utility Functions

`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)`

2.2 Template Tags

`relatives.templatetags.relatives.contents_or_fk_link(field)`

Return field contents or link to related object if foreign key field

`relatives.templatetags.relatives.related_objects(obj)`

Return list of objects related to the given model instance

Example:

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

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

INDICES AND TABLES

- *genindex*
- *search*

PYTHON MODULE INDEX

r

`relatives.templatetags.relatives, ??`
`relatives.utils, ??`