
simple-translation

Release 0.8.6

Øyvind Saltvik

June 03, 2016

1	Overview	1
2	Indices and tables	5
	Python Module Index	7

Overview

There are six steps for using simple-translation:

1. Set settings.LANGUAGES to the languages you want to have translations in.

```
# project/settings.py
LANGUAGES = (
    ('en', 'English'), ('de', 'German')
)
```

2. Make two models in your app, one having the non-translated fields and the other having the translated fields a language field and a ForeignKey to the non-translated model.

```
# appname/models.py
from django.db import models
from cms import settings

class Entry(models.Model):
    pub_date = models.DateTimeField()

class EntryTitle(models.Model):
    entry = models.ForeignKey(Entry)
    language = models.CharField(max_length=2, choices=settings.LANGUAGES)
    title = models.CharField(max_length=255)
    slug = models.SlugField()

    class Meta:
        unique_together = ('language', 'slug')

    def _get_absolute_url(self):
        language_namespace = \
            'simple_translation.middleware.MultilingualGenericsMiddleware' in settings.MIDDLEWARE_CLASSES and '%s:' % self.language or ''
        return ('%sentry_detail' % language_namespace, (), {
            'year': self.entry.pub_date.strftime('%Y'),
            'month': self.entry.pub_date.strftime('%m'),
            'day': self.entry.pub_date.strftime('%d'),
            'slug': self.slug
        })
    get_absolute_url = models permalink(_get_absolute_url)
```

3. For the models to be translatable, create a simple_translate.py file where you register the translated model in the translation_pool.

```
# appname/simple_translate.py
from models import Entry, EntryTitle

from simple_translation.translation_pool import translation_pool
translation_pool.register_translation(Entry, EntryTitle)
```

```
# or if with a language field other than 'language'
translation_pool.register_translation(Entry, EntryTitle, language_field='lang')
```

4. To be able to edit the translated models in the admin. Register the models using the custom TranslationAdmin ModelAdmin.

```
# appname/admin.py
from django.contrib import admin
from models import Entry
from simple_translation.admin import TranslationAdmin

class EntryAdmin(TranslationAdmin):
    pass

admin.site.register(Entry, EntryAdmin)
```

Note

Make sure 'languages' is listed in list_display.

5. Add 'simple_translation.middleware.MultilingualGenericsMiddleware' to settings.MIDDLEWARE_CLASSES

Set up some urls using generic views:

```
# appname/urls.py
from appname.models import Entry
from django.conf.urls.defaults import *

entry_info_dict = {
    'queryset': Entry.objects.all(),
    'date_field': 'pub_date',
    'allow_future': True,
    'slug_field': 'entrytitle__slug'
}

urlpatterns = patterns('',

    (r'^(?P<year>\d{4})/(?P<month>\d{2})/(?P<day>\d{2})/(?P<slug>[-\w]+)/$',
      'django.views.generic.date_based.object_detail', entry_info_dict, 'entry_detail')

)
```

Wrap the urls to namespace them:

```
# translated_urls.py
from django.conf import settings
from django.conf.urls.defaults import *

urlpatterns += patterns('', url(r'^$',
    include('appname.urls', app_name='appname')
))

for langcode in dict(settings.LANGUAGES).keys():
    urlpatterns += patterns('', url(r'^%s/' % langcode,
        include('appname.urls',
            namespace=langcode, app_name='appname'),
        kwargs={'language_code': langcode}
    ))
)
```

6. Add templates for generic views.

```
# templates/appname/entry_detail.html
{% load simple_translation_tags %}

<h1>{% with object|get_preferred_translation_from_request:request as title %}{{ title }}{% endwith %}</h1>
<p>Also available in {{ object|render_language_choices:request|safe }}</p>
```

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

S

`simple_translation`, [1](#)

S

`simple_translation` (module), [1](#)