
django-qartez Documentation

Release 0.8.1

Artur Barseghyan <artur.barseghyan@gmail.com>

Feb 20, 2020

Contents

1	Features	3
2	Prerequisites	5
3	Installation	7
3.1	1. Install	7
3.2	2. Add <i>qartez</i> to your <i>INSTALLED_APPS</i>	7
4	Usage and examples	9
4.1	foo/sitemap.py	9
4.2	urls.py	10
4.3	foo/models.py	11
4.4	foo/views.py	12
4.5	foo/urls.py	12
5	License	13
6	Support	15
7	Author	17
8	Project documentation	19
8.1	Release history and notes	20
8.1.1	0.8.1	20
8.1.2	0.8	20
8.1.3	0.7.1	20
8.1.4	0.7	20
8.1.5	0.6	20
8.1.6	0.5	21
8.1.7	0.1	21
9	Indices and tables	23

The missing XML sitemaps for Django.

At the moment the following XML sitemaps are implemented:

- *qartez.sitemaps.ImagesSitemap*: XML images sitemaps according to the [specs](#).
- *qartez.sitemaps.StaticSitemap*: Sitemap for service pages. Add named patterns or URLs to the sitemap to have it nicely displayed in a separate service XML sitemap.
- *qartez.sitemaps.RelAlternateHreflangSitemap*: Sitemaps: rel="alternate" hreflang="x" implementation. Read the [specs](#).

CHAPTER 2

Prerequisites

- Django: 1.11, 2.0, 2.1, 2.2 and 3.0.
- Python: 2.7, 3.5, 3.6, 3.7, 3.8

3.1 1. Install

Latest stable version on PyPI:

```
pip install django-qartez
```

Latest stable version from GitHub:

```
pip install https://github.com/barseghyanartur/django-qartez/archive/stable.tar.gz
```

3.2 2. Add *qartez* to your *INSTALLED_APPS*

```
INSTALLED_APPS = (  
    # ...  
    'django.contrib.sitemaps',  
    'qartez',  
    # ...  
)
```


CHAPTER 4

Usage and examples

We have an imaginary foo app.

See the [example code](#).

4.1 foo/sitemap.py

```
from django.contrib.sitemaps import Sitemap

from qartez.sitemaps import (
    ImagesSitemap, StaticSitemap, RelAlternateHreflangSitemap
)

from foo.models import FooItem

# ----- XML images sitemap part -----
# Dictionary to feed to the images sitemap.
foo_item_images_info_dict = {
    # Base queryset to iterate when procuding a site map
    'queryset': FooItem._default_manager.exclude(image=None),
    'image_location_field': 'image_url', # Image location (URL)
    'image_title_field': 'title', # Image title
    # An absolute URL of the page where image is shown
    'location_field': 'get_absolute_url'
}

# XML images sitemap.
foo_item_images_sitemap = {
    'foo_item_images': ImagesSitemap(foo_item_images_info_dict,
                                     priority=0.6),
}

# ----- Static sitemap part -----
```

(continues on next page)

(continued from previous page)

```

# Sitemap for service pages like welcome and feedback.
foo_static_sitemap = StaticSitemap(priority=0.1, changefreq='never')
foo_static_sitemap.add_named_pattern('foo.welcome')
foo_static_sitemap.add_named_pattern('foo.contact')

# ----- Normal sitemap part -----
# Normal Foo items sitemap.
class FooItemSitemap(Sitemap):
    changefreq = "weekly"
    priority = 1.0

    def location(self, obj):
        return obj.get_absolute_url()

    def lastmod(self, obj):
        return obj.date_published

    def items(self):
        return FooItem._default_manager.all()

# ----- Alternate hreflang sitemap part -----
# Alternate hreflang sitemap.
class ArticleSitemap(RelAlternateHreflangSitemap):
    # If you want to serve the links on HTTPS.
    protocol = 'https'

    def alternate_hreflangs(self, obj):
        return [('en-us', obj.alternative_object_url),]

    def items(self):
        return FooItem._default_manager.all()

```

4.2 urls.py

```

from foo.sitemap import foo_item_images_sitemap, foo_static_sitemap
from foo.sitemap import FooItemAlternateHreflangSitemap, FooItemSitemap

sitemaps = {
    'foo-items': FooItemSitemap,
    'foo-items-alternate-hreflang': FooItemAlternateHreflangSitemap,
    'foo-static': foo_static_sitemap
}

urlpatterns = [
    # Sitemaps
    (
        r'^sitemap\.xml$',
        'django.contrib.sitemaps.views.index',
        {'sitemaps': sitemaps},
    ),
    (
        r'^sitemap-foo-images\.xml$',
        'qartez.views.render_images_sitemap',
    ),

```

(continues on next page)

(continued from previous page)

```

        {'sitemaps': foo_item_images_sitemap},
    ),
]

```

Note, that it's necessary to add the `'template_name': 'qarteZ/rel_alternate_hreflang_sitemap.xml'` only in case if you are going to use the `qarteZ.RelAlternateHreflangSitemap`.

```

(
    r'^sitemap-(?P<section>.+)\.xml$',
    'django.contrib.sitemaps.views.sitemap',
    {
        'sitemaps': sitemaps,
        'template_name': 'qarteZ/rel_alternate_hreflang_sitemap.xml'
    }
),

```

In order to just get a better idea what kind of models and views are given in the example, see the code parts below.

4.3 foo/models.py

```

class FooItem(models.Model):
    title = models.CharField(_("Title"), max_length=100)
    slug = models.SlugField(_("Slug"), unique=True)
    body = models.TextField(_("Body"))
    date_published = models.DateTimeField(
        _("Date published"),
        blank=True,
        null=True,
        auto_now_add=True
    )

    # Image to be used for XML images sitemap.
    image = models.ImageField(
        _("Headline image"),
        blank=True,
        null=True,
        upload_to='foo-images'
    )

    # URL to be used for alternative hreflang attribute.
    alternative_url = models.URLField(
        _("Alternative URL"),
        blank=True,
        null=True
    )

    class Meta:
        verbose_name = _("Foo item")
        verbose_name_plural = _("Foo items")

    def __str__(self):
        return self.title

```

(continues on next page)

(continued from previous page)

```
def get_absolute_url(self):
    kwargs = {'slug': self.slug}
    return reverse('foo.detail', kwargs=kwargs)

# Shortcut to full image URL for XML images sitemap.
def image_url(self):
    return self.image.url if self.image else ''
```

4.4 foo/views.py

```
# Service welcome page
def welcome(request, template_name='foo/welcome.html'):
    context = {}
    return render_to_response(
        template_name,
        context,
        context_instance=RequestContext(request)
    )

# Service contact page
def contact(request, template_name='foo/contact.html'):
    context = {}
    return render_to_response(template_name, context, \
                              context_instance=RequestContext(request))
```

4.5 foo/urls.py

```
urlpatterns = patterns('foo.views',
    # ...
    # Contact URL
    url(r'^contact/$', view='contact', name='foo.contact'),
    # ...
    # Welcome URL
    url(r'^welcome/$', view='welcome', name='foo.welcome'),
    # ...
)
```


CHAPTER 5

License

GPL-2.0-only OR LGPL-2.1-or-later

CHAPTER 6

Support

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

CHAPTER 7

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

Contents:

Table of Contents

- *django-qartez*
 - *Features*
 - *Prerequisites*
 - *Installation*
 - * *1. Install*
 - * *2. Add qartez to your INSTALLED_APPS*
 - *Usage and examples*
 - * *foo/sitemap.py*
 - * *urls.py*
 - * *foo/models.py*
 - * *foo/views.py*
 - * *foo/urls.py*
 - *License*
 - *Support*
 - *Author*
 - *Project documentation*
 - *Indices and tables*

8.1 Release history and notes

Sequence based identifiers are used for versioning (schema follows below):

`major.minor[.revision]`

- It's always safe to upgrade within the same minor version (for example, from 0.3 to 0.3.4).
- Minor version changes might be backwards incompatible. Read the release notes carefully before upgrading (for example, when upgrading from 0.3.4 to 0.4).
- All backwards incompatible changes are mentioned in this document.

8.1.1 0.8.1

2020-02-20

- Tested against Django 3.0.

8.1.2 0.8

2020-02-19

- Tested against Django 2.0, 2.1, 2.2.
- Tested against Python 3.7 and 3.8.
- Drop support for Django versions prior 1.11.

8.1.3 0.7.1

2017-11-27

- Fixes in docs.

8.1.4 0.7

2017-11-27

- Django 1.8, 1.9, 1.10 and 1.11 support. Drop support for older Django versions.
- Minor fixes.
- Improved tests, PyTest test runner, coverage, introduce factories.

8.1.5 0.6

2014-10-12

- Django 1.7 support.
- Softened *six* requirements.

8.1.6 0.5

2013-09-09

- Python 3 support.

8.1.7 0.1

2013-02-04

- Initial.

CHAPTER 9

Indices and tables

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